Exploring Task Definitions with LLMs: A Study in Citation Text Generation

Anonymous ACL submission

Abstract

Large language models (LLMs) can perform a wide range of tasks in a zero-shot fashion. Yet, defining the task and communicating it to the model remains a challenge. While prior work focuses on prompting strategies taking the task definition as a given, we explore the novel use of LLMs for arriving at an optimal task definition in the first place. We propose an experimental framework consisting of a prompt manipulation module, reference data and a measurement kit, and use it to study citation text generation – a popular natural language processing task without clear consensus on the task definition. Our results highlight the importance of both task definition and task instruction for prompting LLMs, and reveal non-trivial relationships between different evaluation metrics used for the citation text generation task. Our human study illustrates the impact of task definition on non-author human-generated output and reveals the discrepancies between automatic and manual NLG evaluation. Our work contributes to the study of citation text generation in NLP and paves the path towards the systematic study of task definitions in the age of LLMs. Our code is publicly available.[1]

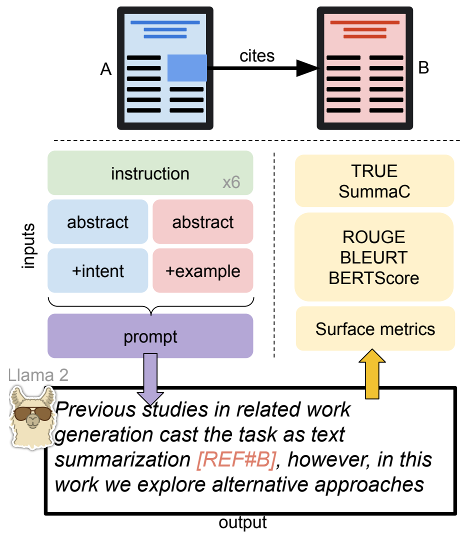

Figure 1: Citation text generation with LLMs. The task is to generate a paragraph of related work from the citing paper (A) about a cited paper (B). This task can be formalized in many different ways. We use the Llama 2-Chat LLM to explore the relationship between task definition and model outputs by manipulating the available inputs and the task instruction (left) and evaluating the output using a range of measurements (right) on a reference collection (top).

1 Introduction

Conventional empirical studies in natural language processing (NLP) mostly follow an established methodology: a task is defined, a model is constructed, and a performance metric is used to evaluate the model. Through a combination of large-scale pre-training and instruction-tuning followed by fine-tuning with human feedback, modern large language models (LLMs) learn to perform many tasks in a zero-shot fashion following a natural language prompt. This allows for unprecedented flexibility and speed with which new tasks can be specified, while removing the need for costly task-specific architecture design and model training (Touvron et al., 2023a,b; Taori et al., 2023; Ouyang et al., 2022; OpenAI, 2023; Chung et al., 2022). Yet, it remains unclear how to effectively leverage LLMs to formally define complex NLP tasks. Furthermore, accurately conveying these tasks to LLMs in natural language poses a novel and ongoing challenge.

We highlight the conceptual difference between the *task definition* and the *task instruction*. Task definition is a set of *input components* considered sufficient to solve the problem at hand, and the expected *output*. For example, sentiment analysis can be defined as predicting a label $l \in L : \{positive|netural|negative\}$ given an input sentence s_i. Task instruction is a free-form natural

[1] repo upon publication

language description of the task based on the task definition. Coupled with the instance-specific data inputs, it forms a *prompt*. An example task instruction would be *"Given a sentence, predict its sentiment from the following options:..."*. Both task definitions and task instructions are variable. The input for sentiment analysis can be enriched with context, and the output can use a different sentiment scale. The task definition can be verbalized into an instruction in many ways, as well.

Task definitions and instructions have been around throughout the history of NLP. While instructions are commonplace in annotation studies, their direct use during inference is a novel feature introduced by LLMs. Prompting study searches for optimal strategies to arrive at good task instructions for LLMs (Section 2.1). Strategies to explore task definitions, on the other hand, are less studied. While this has historically required modifications to the model architecture and fine-tuning of the model, due to their zero-shot capabilities and flexibility with respect to the input, LLMs provide a new and exciting opportunity for such exploration.

In this work, we use LLMs to systematically study the task of *citation text generation* – a widely studied scholarly text generation task (Li and Ouyang, 2022; Funkquist et al., 2022). This task is particularly well-suited for our work since it lacks consensus on the precise task definition, features a complex input space combined with multiple plausible outputs, and has not yet been tackled in a zero-shot setting with instruction-tuned LLMs. While Funkquist et al. (2022) unify multiple citation text generation datasets to enable systematic comparison of NLP models, they leave open the exact definition of the task and focus on the supervised learning scenario, while leaving zero-shot citation generation under-investigated.

To address this gap, we design a framework to systematically investigate the impact of task definition and task instruction on citation text generation (Figure 1). It consists of three parts: the (1) prompt manipulation module systematically varies the task instruction and the input components available to the model; (2) reference data serves as a source of examples and reference for evaluation; (3) measurement kit allows characterizing the model outputs in response to the prompts. Through extensive experiments, we study the *interactions* between the instruction, input components and measurable properties of the outputs for citation text generation. In summary, this work contributes the following:

- We outline a framework for studying task definitions for citation text generation using LLMs, featuring a novel use of *unstructured intents* as an input component to guide the generation process;
- We introduce a measurement kit to characterize the generated citation texts from multiple perspectives, along with a novel reference corpus of citation texts based on the ACL Anthology enriched with unstructured citation intents;
- We use our framework to study the impact of task definition on the model outputs, and examine the relationships between the metrics in the measurement kit;
- We refine our findings in a human evaluation study, where we compare human- and machine-generated citation texts in terms of both automatic measurements and human rankings.

Summary of findings. We find (Section 5) that LLM generations do not always obey the formal requirements stated in the task instruction and tend to over-generate text. Task definition and task instruction both impact the generations, and their effects add up. The results suggest that while the *relative* performance of different task definitions might be estimated using a small set of instructions, the best *absolute* performance requires experimenting with a wide array of instruction wordings. Through correlation analysis we observe that the NLG metrics used in our measurement kit are complementary, motivating the use of wide-spanning measurement sets for NLG tasks that feature several equally acceptable answers. Our human studies (Section 6) reveal that – contrary to the automatic measurements – humans still prefer human-generated citation texts, and that the effects of task definition on LLM generation quality can be replicated in a setting where humans generate citation texts manually. Our qualitative analysis provides additional hypotheses and insights to guide future work in LLM-based citation text generation.

2 Background

2.1 LLMs and Prompting

Instruction-tuned large language models (LLMs) demonstrate competitive performance across a wide range of NLP tasks (Touvron et al., 2023a,b; Taori et al., 2023; Ouyang et al., 2022; OpenAI, 2023; Chung et al., 2022). Unlike traditional models, LLMs can be prompted with free-form textual queries. Prompts can be manipulated through sim-

Study	Level	Abstract	Intent	Example	Model	Evaluation
(AbuRa'ed et al., 2020)	sent	Tgt	-	-	PG	ROUGE
(Xing et al., 2020)	sent	Tgt	-	-	PG	ROUGE, Human
(Ge et al., 2021)	sent	Tgt	C	-	Enc. + LSTM	ROUGE, Human
(Kasanishi et al., 2023)	para	Tgt	-	-	FiD	ROUGE, Human
(Chen et al., 2021)	para	Tgt	-	-	Hier. Enc.	ROUGE, Human
(Luu et al., 2021)	sent	Src/Tgt	-	-	GPT-2	ROUGE, BLEU, Human
(Lu et al., 2020)	para	Src/Tgt	-	-	PG	ROUGE, Human
(Arita et al., 2022)	sent	Src/Tgt	C	-	T5	ROUGE
(Jung et al., 2022)	sent	Src/Tgt	C	-	T5, BART	ROUGE, SciBERTScore Human
(Wu et al., 2021)	para	Src/Tgt	C	-	FiD	ROUGE, BLEU, BLEURT, Meteor
Ours	para	Src/Tgt	F	✓	Llama 2-Chat	ROUGE, BERTScore, BLEURT, TRUE, SummaC, Surface measurements, Human

Table 1: Our work in the context of prior work on citation text generation. We explore alternative task definitions for citation text generation in the context of state-of-the-art instruction-following LLMs, using a comprehensive measurement kit and two novel input components: free-form citation intent and example sentence. sent – sentence, para – paragraph, PG – pointer-generator network, FiD – fusion-in-decoder network, C – categorical intents, F - free-form intents, Src - source (citing) paper, Tgt - target (cited) paper.

ple textual adjustments, allowing the user to guide model behavior at inference time without the need to update the model.

The search for efficient prompting strategies is a trending research topic. The initial enthusiasm about zero-shot capabilities of LLMs (Brown et al., 2020; Kojima et al., 2022; Sanh et al., 2022) has been countered by evidence that LLMs are sensitive to minor changes in prompt formulation (Lu et al., 2022; Mishra et al., 2022; Wang et al., 2023a; Zhu et al., 2023). Several techniques for arriving at an optimal task wording have been proposed, e.g. choosing lowest-perplexity prompts (Gonen et al., 2022; Yin et al., 2023; Gu et al., 2023; Lou et al., 2023). In-context learning (ICL) based on task demonstrations has shown promise (Ouyang et al., 2022; Wang et al., 2022b, 2023b; Chung et al., 2022), yet Min et al. (2022) suggest that the main source of performance improvements in ICL is not the task demonstration, but the information it provides about the label space, input distribution and output format. All in all, findings to date emphasize the importance and complexity of communicating the task at hand to an LLM. While prior work focuses on arriving at an optimal task *instruction*, we investigate the impact of alternative task *definitions* on LLM behavior for citation text generation.

2.2 Citation Text Generation

Citation text generation is a widely studied task aiming to increase the efficiency of scientific work. It has been cast as a sentence-level (AbuRa'ed et al., 2020; Ge et al., 2021; Li et al., 2022b, 2023) and paragraph-level task (Lu et al., 2020; Chen et al., 2021, 2022; Wu et al., 2021; Kasanishi et al., 2023), as extractive (Hoang and Kan, 2010; Hu and Wan, 2014; Chen and Zhuge, 2019; Wang et al., 2020) and abstractive summarization (AbuRa'ed et al., 2020; Li et al., 2022a; Lu et al., 2020; Chen et al., 2021; Luu et al., 2021; Kasanishi et al., 2023). Different input components such as categorical citation intents and citation network information have been explored (Wu et al., 2021; Arita et al., 2022; Gu and Hahnloser, 2022; Jung et al., 2022; Ge et al., 2021; Wang et al., 2021, 2022a; Chen et al., 2022; Gu and Hahnloser, 2023). Table 1 summarizes task definitions and modeling approaches from prior work: we are the first to systematically assess the impact of different task definitions for citation text generation using a modern instruction-tuned LLM.

The differences in task definitions prevent systematic comparison of citation text generation approaches. To address this, Funkquist et al. (2022) propose a benchmark that incorporates multiple prior datasets under a general task definition framework and casts the task as text-to-text generation. Our paper builds upon this work and differs from it in two major regards. First, Funkquist et al. (2022) unify a range of prior datasets adhering to different task definitions, yet they do not systematically compare different task definitions and leaves the question of *"what information is in fact required* *to produce accurate citation texts"* open for future investigation. Our work addresses this question. Second, while Funkquist et al. (2022) assume the supervised learning scenario, we – for the first time – explore citation text generation in a zero-shot setting using instruction-tuned LLMs, in the broader context of state-of-the-art LLM research.

In addition, we explore the impact of citation intents on citation text generation. Citation intent prediction and the use of intent in citation text generation have been previously investigated (Teufel et al., 2006; Abu-Jbara et al., 2013; Jurgens et al., 2018; Cohan et al., 2019; Lauscher et al., 2022). Citation intent is commonly modeled via categorical labels, e.g., "Background" or "Method" (Wu et al., 2021; Arita et al., 2022; Gu and Hahnloser, 2022; Jung et al., 2022). Directly integrating categorical intents into generation has potential limitations: information loss due to coarse labeling will lead to difficulties in generating a paragraph-level citation text based on a single intent label. Motivated by this, we for the first time experiment with alternative machine-generated unstructured intents derived for each citation text paragraph, discussed in Section 3.2 and exemplified in Figure 2.

2.3 NLG Evaluation

Natural language generation (NLG) is notoriously hard to evaluate automatically, and human evaluation is often associated with high cost and low reproducibility (Belz et al., 2023). Conventional automatic evaluation metrics based on token or token embedding similarity like ROUGE (Lin, 2004), BERTScore (Zhang et al., 2020), BLEURT (Sellam et al., 2020) are widely used in NLG. Yet, these metrics cannot detect factual errors in the model output. Furthermore, they are not well suited for evaluating whether the model output meets the formal criteria set by the task definition.

The former challenge can be partially addressed by natural language inference-based metrics. In particular, TRUE (Honovich et al., 2022) and SummaC (Laban et al., 2022) aim to detect compatibility between the generated output and the reference. The latter challenge – lack of formal evaluation of the outputs – can be mitigated by using simple surface-level metrics to check whether task instructions are followed. Yet this type of analysis is often omitted (Jang et al., 2022). While most prior work in citation text generation relies on a small number conventional evaluation metrics (Table 1), our measurement kit encompasses conventional, surface-level and NLI-based metrics and enables comprehensive analysis of the generated texts. We complement this by a human evaluation study where we manually rank citation texts, detailed in Section 6.

3 Method

The goal of our study is to explore the impact of task definition on citation text generation outputs in the context of state-of-the-art LLMs. We focus on paragraph-level citation text generation for the paragraphs that cite a single paper, as it represents the most dominant use case and provides an ideal, straightforward setup to explore the task definition space for citation text generation. The key components of our experimental framework are the prompt manipulation module, the reference data, and the measurement kit, detailed below.

3.1 Prompt Manipulation

The prompt manipulation module enables systematic variation of task definitions and the subsequent task instructions. For the task definition, we experiment with four types of input components, combined with six distinct dynamically-adjusted human-written task instructions. The four task definition input components are as follows:

- **Target (cited) paper abstract:** Contains the abstract of the cited paper, which is expected to contain core information about the cited work.
- **Source (citing) paper abstract:** Contains the abstract of the citing paper, which is expected to provide additional context to guide generation. Cited and citing paper abstracts are commonly used input components in citation text generation literature (see Table 1).
- **Citation intent:** A single natural-language sentence describing the intent of the citation paragraph automatically derived from the reference paragarph (Section 3.2).
- **Example sentence:** An example sentence that refers to the cited paper but does not belong to the currently considered citing paper (Section 3.2).

The instructions generally ask the model to write a single related work paragraph based on the input components from the citing and cited paper, while using `[REF#1]` to refer to the cited paper (Figure 2). The specific wording of the instructions varies. The full list of instructions is given in the Appendix D. The prompt is constructed by *adjust-*

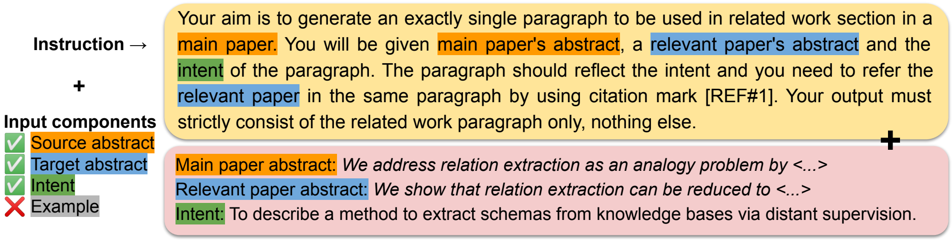

Figure 2: Prompt manipulation module constructs the prompt by combining the instruction (top) with selected input components (left) and the corresponding instance data (bottom), incl. machine-generated citation intent sentence. The result serves as input to the LLM.

ing the instruction depending on the chosen input component combination, and concatenating the instruction with the input data for a given instance. The result is passed to the model for inference.

3.2 Reference Data

The range of possible task definition depends on the available data. Thus, our study requires rich input data representation. For paragraph-level generation, the data must contain *full paragraphs*. We further focus on paragraphs that belong to *related work sections*, where the authors are most likely to discuss cited work rather than their own contributions, compared to other sections. This requires the papers to be *structured at least on the section level*. The cited papers' data should be *readily accessible* based on the citation. Both citing and cited papers should be complemented with *metadata*, including at least their abstracts, since this information is commonly used to generate citation texts.

Among public datasets, Kasanishi et al. (2023) and Lu et al. (2020) come closest to our requirements. Yet, Kasanishi et al. (2023) is limited to literature review and survey papers, and our preliminary investigation of Lu et al. (2020) has shown that some abstracts and citations were missing from the data. To address these limitations, we compiled a new reference dataset based on the parsed ACL Anthology by Rohatgi (2022). The dataset construction details and statistics are provided in Appendix C. We have used the above parsed corpus to extract citation text paragraphs, limiting our paragraph selection such that the cited papers also belong to our reference data, ensuring that full paper content and metadata are readily available for both citing and cited papers. Using the structured parses from the data and a set of rule-based heuristics we selected $5,971$ related work paragraphs – comparable in size to the test set of Lu et al. (2020). For the experiment (Section 4), the data was further filtered to paragraphs that contain a citation to a *single* paper, resulting in $2,729$ related work paragraphs.

We also use this related work paragraph collection to extract **example sentences** that exemplify how a certain paper can be cited independently from the current citing paper. During experiments, we use this pool to select example sentences most similar to the gold reference paragraph via the SBERT model (Reimers and Gurevych, 2019). Additionally, to steer generation, we enrich the reference paragraphs with free-form **intent sentences** defined as a single sentence describing the reason a particular paper is cited in a given paragraph. Intuitively, intents serve as a "hint" to reduce the possible space of generations and steer the LLM output towards the golden reference.[2] In this work, we used FlanT5-XXL (11B) model to generate the intents: an example generated intent sentence can be found in Figure 2. We discuss the advantages and limitations of this approach in Section 8, and provide details on intent generation along with examples in Appendix C.5.

3.3 Measurement Kit

We characterise the generated paragraphs with multiple groups of measurements: surface metrics, conventional NLG metrics, and NLI-based metrics. As we show later, these groups provide complementary insights about the model outputs in response to the varying task definition and instruction.

Surface metrics. All of our task instructions request the model to generate one paragraph of citation text. However, the model might not follow this requirement precisely. To evaluate, we measure the average *paragraph count* in generated citation

[2]This is in line with the expert recommendations for writing literature reviews: for instance, Ridley (2012) suggests to use informal writing to form the basis for the actual literature review, such as "*What are the methodological flaws of the previous methods?*"

texts. Similarly, our instructions request the model to use a *citation mark* to refer to the cited paper in the generated text, e.g. `[REF#1]`. We check whether the model has used this token at least once during generation. Lastly, we calculate n-gram overlap between the input and the model output to check whether the model copies from the prompt.

Conventional metrics. To compare the generated text to the reference, we compute several conventional NLG metrics: ROUGE-L (Lin, 2004), BERTScore (Zhang et al., 2020) and BLEURT (Sellam et al., 2020). ROUGE is the most commonly used metric in prior work on citation text generation – yet it operates on the surface level and lacks the capacity to evaluate semantic correspondence between the two sequences. This is addressed by the two more recent metrics – BERTScore and BLEURT – that use BERT-based (Devlin et al., 2019) representations to compare the generated text to the reference on semantic level, showing greater robustness to paraphrases and better alignment with human assessments.

NLI-based metrics. To measure factual consistency between the gold reference and the model output, we use two NLI models (TRUE and SummaC) trained on curated fact-checking datasets. Note that we use {gold reference, model output} instead of {abstracts, model output} as the input to the NLI models because we focus on exploring the task definition space for related work generation and identifying the key input components needed to reconstruct the gold reference. TRUE makes binary decisions regarding entailment for a given textual pair (Honovich et al., 2022). SummaC (Laban et al., 2022) generates NLI scores from the sentences of compared texts and calculates an overall score.

4 Experiments

For all experiments we use Llama 2-Chat (13B) (Touvron et al., 2023b) – a state-of-the-art, open instruction-tuned LLM. We use the prompt manipulation module to generate prompts consisting of instructions and data inputs, according to the chosen configuration. It is passed to the model for inference, for each data instance. We analyze the outputs using our measurement kit. Generating citation texts for all instances and all configurations discussed below takes ~30 hours on a single NVIDIA A100 GPU with 80GB memory. Further details are specified in Appendix A.

Conf.	NG-3	PC	CM	(ctd.)	NG-3	PC	CM
1+A	26.70	1.50	30.69	4+A	24.3	1.01	54.55
1+A+I	24.09	1.48	41.62	4+A+I	24.35	1.02	42.73
1+A+E	26.97	1.64	74.36	4+A+E	26.61	1.03	82.07
1+A+I+E	24.56	1.63	77.54	4+A+I+E	25.18	1.05	78.56
2+A	26.04	1.08	63.07	5+A	30.04	1.40	25.95
2+A+I	26.11	1.11	91.30	5+A+I	27.02	1.56	30.74
2+A+E	23.74	1.11	82.87	5+A+E	28.42	1.58	76.99
2+A+I+E	24.52	1.15	89.71	5+A+I+E	26.45	1.77	76.20
3+A	25.37	1.31	37.56	6+A	23.55	1.01	92.55
3+A+I	25.54	1.32	28.42	6+A+I	26.81	1.07	85.90
3+A+E	27.33	1.48	76.25	6+A+E	24.88	1.07	95.34
3+A+I+E	26.93	1.47	75.52	6+A+I+E	27.24	1.10	95.77

Table 2: Surface measurements. #Instruction + **A**bstract + **I**ntent + **E**xample. NG-3: averaged 3-gram overlap (%); PC: paragraph count, CM: citation mark usage (%).

5 Results

We use the following notation to discuss experimental configurations: `#(+A)(+I)(+E)`, where `#` is the instruction identifier, `+A` denotes source and target paper abstracts , `+I` denotes the intent sentence, `+E` denotes an example citation sentence that cites the given cited paper. Note that the instructions are adjusted to reflect the input components present in a given configuration. The example input in Figure 2 corresponds to the configuration `1+A+I`. Table 4 and Figure 3 present our measurements across different configurations; full results are given in Appendix B. The measurements allow us to explore a range of questions about the role of task definition in citation text generation in the context of modern LLMs.

RQ1: What are the characteristics of the generated citation texts? By construction our reference texts consist of a single paragraph with a single citation marker. Yet, the generated texts often violate this constraint (Table 2). Some configurations like `5+A+I+E` systematically over-generate text with an average of 1.77 paragraphs per output, others like `5+A` under-generate citation markers. We note that for five out of six instructions, explicitly introducing an example sentence with a citation marker makes the model generate it more consistently – yet, in other cases like `6+A` the instruction itself suffices for the model to reliably generate the citation mark. Similarly, in `4+A` and `6+A`, the model follows the paragraph count limitation almost perfectly.

RQ2: What is the impact of the task definition on generated texts? We find that additional input components in the task definition have positive influence on performance in terms of both conventional and NLI-based measurements (Figure

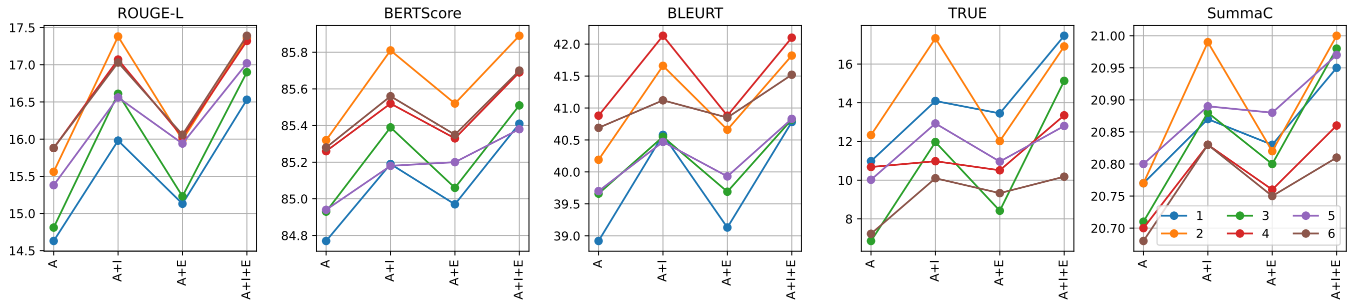

Figure 3: Conventional and NLI-based metric results. **A**bstract + **I**ntent + **E**xample, #Instruction color-coded.

3). We observe that providing the model with only abstracts (+A) systematically yields the lowest degree of correspondence between the generated text and the reference across all task instructions and all automatic evaluation metrics. We also observe that providing models with intent (+I) increases the correspondence between generated and reference citation texts for all six instructions, while example sentence (+E) has this effect for the four out of six instructions. Providing intent and example jointly shows a combined effect and yields the best correspondence in 16 out of 18 (six main configuration x three metrics) measurements in conventional metrics and in 10 out of 12 comparisons for NLI based metrics. The positive impact of *intent* and *example* replicates in our experiments on non-author human-generated text (Section 6). We note that the ranking of configurations remains mostly consistent across the task instructions and measurements. This suggests that the *relative* performance of different input configurations might be estimated based on a small number of instruction variations.

RQ3: What is the effect of the instructions? We observe that the instruction – i.e. how the task is described to the model – affects the correspondence between generated and reference citation texts (Figure 3). Our results suggest that the effects of the instructions and input components are orthogonal and thus add up: the difference between highest- and lowest-performing configuration are up to 2.8 (6+A+I+E vs 1+A) points ROUGE-L, 1.1 (2+A+I+E vs 1+A) points BERTScore, 3.2 (4+A+I+E vs 1+A) points BLEURT and 10.6 points for TRUE[3]. In addition, the effect of the instruction can be observed in surface measurements: for example, there is a substantial difference between 1+A and 6+A in terms of the average paragraph count and the average citation mark ratio. Hence, both instruction and input configuration are important factors in comparing citation text generation models, and should be investigated jointly. In terms of absolute performance, the best input configuration might be undermined by suboptimal instruction wording. In contrast to RQ2, this suggests that in search for the highest *absolute* performance, a wide range of instructions should be explored.

[3]The magnitude is within the common range reported in related work, e.g. (Funkquist et al., 2022; Kasanishi et al., 2023; Wu et al., 2021) for ROUGE, BERTScore and BLEURT, and (Gao et al., 2023) for TRUE.

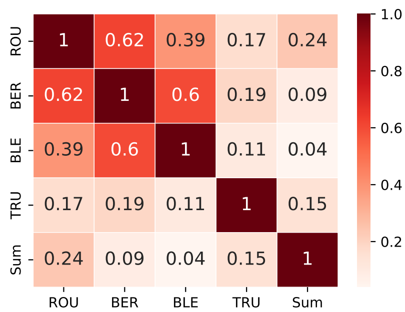

Figure 4: Pearson correlation between instance-level measurements over all configurations.

RQ4: What are the relationships between the measurements? From Figure 4, we observe that conventional metrics show high correlations among themselves, but the correlations to the NLI-based metrics are low. TRUE and SummaC are less correlated with each other compared to conventional metrics. We hypothesise that since TRUE evaluates the entailment relation between two sequences in binary manner, i.e. "entailment" or "contradiction", it might be sensitive to the changes in outputs. SummaC, on the other hand, processes paragraphs at the sentence level and produces an overall score by convolution – decreasing its sensitivity, but also leading to smaller differences between prompt configurations. These observations highlight the importance of multiple complimentary measurements for the citation text generation as opposed to the standard single-metric ROUGE-based evaluation.

Configuration	ROUGE-L	BERTScore	BLEURT	TRUE	SummaC	Best-Worst Scaling
[LLM] 6+A	15.52	85.27	41.93	13.33	20.53	-0.33
[LLM] 6+A+I+E	**17.49**	**85.87**	**43.96**	**20.00**	**20.77**	-0.23
[H] 6+A	14.16	85.69	37.37	10.00	21.33	-0.03
[H] 6+A+I+E	**16.25**	**85.88**	**38.56**	**13.33**	**22.00**	**<u>0.52</u>**

Table 3: Human study results on a subsample of instances. H – human-generated, LLM – machine-generated.

6 Human evaluation

To get further insights into citation text generation with LLMs and the impact of task definition on this process, we have conducted a human generation study and a human evaluation study (Appendix E).

Human vs machine-generated citation texts. For generation, we sampled 30 instances from the single-paragraph reference data used in our main experiment. Three human annotators with background in NLP composed related work paragraphs for these instances given two prompts: `6+A` (abstracts only) and `6+A+I+E` (abstracts, intents and example). We then compared human-generated texts to the ones generated by the LLM using our measurement kit (Table 3). We observe that conventional NLG evaluation metrics and TRUE favor LLM outputs, while SummaC shows preference for human-generated texts.

Human ratings. Same annotators carried out human evaluation comparing LLM-generated and human-generated paragraphs[4]. We used *Best-Worst Scaling* (Louviere et al., 2015), which is more dependable than pairwise comparisons while requiring less annotation effort (Kiritchenko and Mohammad, 2017). Given the gold reference and four outputs (two LLM-generated, two human-generated), the annotators selected the best and worst outputs in terms of their correspondence with the gold reference. The score was calculated as the difference between the percentage of times the configuration was selected as the best or worst, from -1 (always the worst) to 1 (always the best). Table 3 presents the results and allows two observations. First, contrary to the conventional metric results, humans preferred human-written citation texts to the LLM generations. Second, the positive effect of providing intent and example from the main experiment holds in the case when the citation texts are generated by human annotators. This implies that both components are important input for the citation text generation task in *real-world scenarios* and should be integrated into future research.

[4]The instances were distributed such that no annotator would rate their own generated instance to avoid bias.

Qualitative observations. Our evaluation yielded few informal insights which we deem useful for follow-up research. Despite the conditions being hidden, we were often able to distinguish LLM-generated texts from human-generated ones: LLM generations were typically less brief and less specific. We observed that the wording of the instruction affects the style of the generated paragraph: for some instructions, the model tended to generate a text *comparing* two papers, instead of *discussing* one paper in context of the other. As this is not reflected in the metric performance scores, we hypothesize that pragmatic mismatch might not be captured by the automatic evaluation metrics. We found that the success of generations depended on the content of the gold reference: while high-level discussion of related work can be generated from the abstracts, going into specifics of a paper requires the information not available in the input. The content of the abstracts affected the generations as well: uninformative abstracts were hard to generate from, both for humans (who wrote short and uninformative citation texts in response) and for LLMs (that were forced to hallucinate text). Since the setting of our human study is insufficient to investigate these observations empirically, we leave this exploration for future research.

7 Conclusion

To solve a task, one needs to define the task. As NLP tasks become increasingly complex, creative and applied, the space of possible inputs and acceptable outputs grows as well, motivating the need for approaches to systematically compare task definitions. We have proposed a framework for comparing task definitions for a popular scholarly NLP task – citation text generation. We used our framework to study the impact of task definition and task instruction on the task performance, both by LLMs and by human annotators. Our insights contribute to a better understanding of the role of task definitions and instructions in LLM-based language processing, and our framework facilitates the study of citation text generation in the age of LLMs.

8 Limitations

We now turn to the limitations of our study to be addressed by future work.

Comparison to state of the art. We do not compare the performance of our citation text generation system to prior models, since the goal of our work is to study the *effect* of task definition and instructions, and *not* to produce a top-performing model instance. Besides, given the capabilities of modern LLMs, side-by-side comparison to prior work would likely put earlier models at unfair disadvantage and conflate a wide range of potential sources of improvement.

Modeling human preference. Task definition encompasses input components and the output which are both variable. In this work, we focused on systematically varying the input space, while resorting to a wide range of metrics and human evaluation to characterize the output space. The results of our human evaluation suggest that there is still a gap between automatic measurements and human preference. We claim that more accurate models of human preference are urgently needed for the citation text generation task. Our qualitative insights can serve as a basis for constructing such models in the future.

Limitations of the setup. To keep our study tractable, we had to impose limitations on our setup. Considering only related work paragraphs that contain a single citation is a technical limitation, which can be revisited once open LLMs that can efficiently handle long inputs become available. We expect additional effects due the varying model's capability to discuss multiple cited papers in one paragraph at once. While we put effort into validating our findings using a range of instructions instead of a single prompt, adding more instructions would allow to further verify our findings and to get better estimates of the absolute performance. We thus recommend expanding the instruction pool for the follow-up work interested in producing a best-performing system. In our experiments we considered three groups of input components: abstracts, intents, and example sentence. This set can be easily extended based on our reference data, which contains both rich metadata and pointers to the dataset with the parsed full papers for both citing and cited works, with one and multiple citations per paragraph.

Language and domain Our experiments are limited to English and to the papers from the ACL Anthology. This is a common feature of scholarly NLP, due to English being the standard language of communication in many research fields and due to availability and open licensing of the ACL Anthology. Applying our approach in a cross-lingual and multi-lingual setting and in novel domains is an engaging future work direction which can be pursued once the research infrastructure is available.

Machine-generated intents We experiment with free-form, unstructured citation intents to guide the generation. Since manually creating a citation intent for each dataset instance is not feasible, we have generated them from the gold reference paragraphs using a separate model (Flan-T5 vs Llama 2 in the main experiment). The drawback of this approach is that these sentences might arguably leak some keywords and subsequences from the gold reference paragraphs, inflating the performance measurements. We point out that intent sentences normally do not contain enough information to generate a whole paragraph (Appendix C.6), which is verified through our human generation study. Furthermore, encountering some sequences from the given unstructured intent in the resulting generated citation text would be acceptable in a real-world application scenario. As alternative, future work can explore citation text generation with manually curated intent sentences on a smaller subset of our data. We note that we do not compare unstructured vs categorical intents in this work, as claiming superiority of one or the other approach lies beyond our scope. We leave this investigation to the future.

Ethics Statement

We believe that a systematic study of task definitions is an important basic research direction for NLP without ethical implications. While the misuse of citation text generation could lead to reduced engagement with the scientific literature, we believe that such systems – used as an aid, not as replacement for paper reading – could facilitate exploration of vast scientific literature, and that the benefits of such systems outweigh the risks. Our data is constructed based on publicly available, openly licensed sources, and our experiments are conducted with an open large language model, facilitating long-term reproducibility of our experiments, and allowing the community to build upon the artifacts of our study.

References

Amjad Abu-Jbara, Jefferson Ezra, and Dragomir Radev. 2013. Purpose and polarity of citation: Towards NLP-based bibliometrics. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 596–606, Atlanta, Georgia. Association for Computational Linguistics.

Ahmed AbuRa'ed, Horacio Saggion, Alexander Shvets, and Àlex Bravo. 2020. Automatic related work section generation: Experiments in scientific document abstracting. *Scientometrics*, 125(3):3159–3185.

Akito Arita, Hiroaki Sugiyama, Kohji Dohsaka, Rikuto Tanaka, and Hirotoshi Taira. 2022. Citation sentence generation leveraging the content of cited papers. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 170–174, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Anya Belz, Craig Thomson, Ehud Reiter, and Simon Mille. 2023. Non-repeatable experiments and non-reproducible results: The reproducibility crisis in human evaluation in NLP. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3676–3687, Toronto, Canada. Association for Computational Linguistics.

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.

Jingqiang Chen and Hai Zhuge. 2019. Automatic generation of related work through summarizing citations. *Concurrency and Computation: Practice and Experience*, 31(3):e4261.

Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Rui Yan, Xin Gao, and Xiangliang Zhang. 2022. Target-aware abstractive related work generation with contrastive learning. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 373–383, New York, NY, USA. Association for Computing Machinery.

Xiuying Chen, Hind Alamro, Mingzhe Li, Shen Gao, Xiangliang Zhang, Dongyan Zhao, and Rui Yan. 2021. Capturing relations between scientific papers: An abstractive model for related work section generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6068–6077, Online. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3586–3596, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Martin Funkquist, Ilia Kuznetsov, Yufang Hou, and Iryna Gurevych. 2022. CiteBench: A benchmark for scientific citation text generation. *arXiv preprint arXiv:2212.09577*.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023. Enabling large language models to generate text with citations. *arXiv preprint arXiv:2305.14627*.

Yubin Ge, Ly Dinh, Xiaofeng Liu, Jinsong Su, Ziyao Lu, Ante Wang, and Jana Diesner. 2021. BACO: A background knowledge- and content-based framework for citing sentence generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1466–1478, Online. Association for Computational Linguistics.

Hila Gonen, Srini Iyer, Terra Blevins, Noah A. Smith, and Luke Zettlemoyer. 2022. Demystifying prompts in language models via perplexity estimation.

Jiasheng Gu, Hongyu Zhao, Hanzi Xu, Liangyu Nie, Hongyuan Mei, and Wenpeng Yin. 2023. Robustness of learning from task instructions. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13935–13948, Toronto, Canada. Association for Computational Linguistics.

Nianlong Gu and Richard HR Hahnloser. 2022. Controllable citation text generation. *arXiv preprint arXiv:2211.07066*.

Nianlong Gu and Richard H.R. Hahnloser. 2023. SciLit: A platform for joint scientific literature discovery, summarization and citation generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 235–246, Toronto, Canada. Association for Computational Linguistics.

Shruthi Hariprasad, Sarika Esackimuthu, Saritha Madhavan, Rajalakshmi Sivanaiah, and Angel S. 2022. SSN_MLRG1@DravidianLangTech-ACL2022: Troll meme classification in Tamil using transformer models. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 132–137, Dublin, Ireland. Association for Computational Linguistics.

Cong Duy Vu Hoang and Min-Yen Kan. 2010. Towards automated related work summarization. In *Coling 2010: Posters*, pages 427–435, Beijing, China. Coling 2010 Organizing Committee.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Yue Hu and Xiaojun Wan. 2014. Automatic generation of related work sections in scientific papers: An optimization approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1624–1633, Doha, Qatar. Association for Computational Linguistics.

Joel Jang, Seonghyeon Ye, and Minjoon Seo. 2022. Can large language models truly follow your instructions? In *NeurIPS ML Safety Workshop*.

Shing-Yun Jung, Ting-Han Lin, Chia-Hung Liao, Shyan-Ming Yuan, and Chuen-Tsai Sun. 2022. Intent-controllable citation text generation. *Mathematics*, 10(10).

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics*, 6:391–406.

Tetsu Kasanishi, Masaru Isonuma, Junichiro Mori, and Ichiro Sakata. 2023. SciReviewGen: A large-scale dataset for automatic literature review generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6695–6715, Toronto, Canada. Association for Computational Linguistics.

Svetlana Kiritchenko and Saif Mohammad. 2017. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 465–470, Vancouver, Canada. Association for Computational Linguistics.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2022. SummaC: Re-visiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177.

Anne Lauscher, Brandon Ko, Bailey Kuehl, Sophie Johnson, Arman Cohan, David Jurgens, and Kyle Lo. 2022. MultiCite: Modeling realistic citations requires moving beyond the single-sentence single-label setting. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1875–1889, Seattle, United States. Association for Computational Linguistics.

Pengcheng Li, Wei Lu, and Qikai Cheng. 2022a. Generating a related work section for scientific papers: An optimized approach with adopting problem and method information. *Scientometrics*, 127(8):4397–4417.

Xiangci Li, Yi-Hui Lee, and Jessica Ouyang. 2023. Cited text spans for citation text generation. *arXiv preprint arXiv:2309.06365*.

Xiangci Li, Biswadip Mandal, and Jessica Ouyang. 2022b. CORWA: A citation-oriented related work annotation dataset. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5426–5440, Seattle, United States. Association for Computational Linguistics.

Xiangci Li and Jessica Ouyang. 2022. Automatic related work generation: A meta study. *arXiv preprint arXiv:2201.01880*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Renze Lou, Kai Zhang, and Wenpeng Yin. 2023. Is prompt all you need? No. a comprehensive and broader view of instruction learning. *arXiv preprint arXiv:2303.10475*.

Jordan J. Louviere, Terry N. Flynn, and A. A. J. Marley. 2015. *Best-Worst Scaling: Theory, Methods and Applications*. Cambridge University Press.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Yao Lu, Yue Dong, and Laurent Charlin. 2020. Multi-XScience: A large-scale dataset for extreme multi-document summarization of scientific articles. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8068–8074, Online. Association for Computational Linguistics.

Kelvin Luu, Xinyi Wu, Rik Koncel-Kedziorski, Kyle Lo, Isabel Cachola, and Noah A. Smith. 2021. Explaining relationships between scientific documents. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2130–2144, Online. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to GPTk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612, Dublin, Ireland. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Diana Ridley. 2012. *The Literature Review: A Step-by-Step Guide for Students*. SAGE Study Skills Series. SAGE Publications.

Shaurya Rohatgi. 2022. ACL Anthology Corpus with full text. Github.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Rajalakshmi Sivanaiah, Angel Suseelan, S Milton Rajendram, and Mirnalinee T.t. 2020. TECHSSN at SemEval-2020 task 12: Offensive language detection using BERT embeddings. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2190–2196, Barcelona (online). International Committee for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford Alpaca: An instruction-following LLaMA model. `https://github.com/tatsu-lab/stanford_alpaca`.

Simone Teufel, Advaith Siddharthan, and Dan Tidhar. 2006. Automatic classification of citation function. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 103–110, Sydney, Australia. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jindong Wang, Xixu HU, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Wei Ye, Haojun Huang, Xiubo Geng, Binxing Jiao, Yue Zhang, and Xing Xie. 2023a. On the robustness of chatGPT: An adversarial and out-of-distribution perspective. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*.

Pancheng Wang, Shasha Li, Haifang Zhou, Jintao Tang, and Ting Wang. 2020. ToC-RWG: Explore the combination of topic model and citation information for automatic related work generation. *IEEE Access*, 8:13043–13055.

Qingqin Wang, Yun Xiong, Yao Zhang, Jiawei Zhang, and Yangyong Zhu. 2021. AutoCite: Multi-modal representation fusion for contextual citation generation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 788–796, New York, NY, USA. Association for Computing Machinery.

Yifan Wang, Yiping Song, Shuai Li, Chaoran Cheng, Wei Ju, Ming Zhang, and Sheng Wang. 2022a. DisenCite: Graph-based disentangled representation learning for context-specific citation generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11449–11458.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Arjun Naik, Atharva andAshok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022b. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jia-Yan Wu, Alexander Te-Wei Shieh, Shih-Ju Hsu, and Yun-Nung Chen. 2021. Towards generating citation sentences for multiple references with intent control. *arXiv preprint arXiv:2112.01332*.

Xinyu Xing, Xiaosheng Fan, and Xiaojun Wan. 2020. Automatic generation of citation texts in scholarly papers: A pilot study. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6181–6190, Online. Association for Computational Linguistics.

Fan Yin, Jesse Vig, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2023. Did you read the instructions? rethinking the effectiveness of task definitions in instruction learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3063–3079, Toronto, Canada. Association for Computational Linguistics.

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*.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, et al. 2023. PromptBench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528*.

A Implementation details

We have obtained Llama 2-Chat weights[5] and converted the checkpoints to the Huggingface format. We utilized the Huggingface framework (Wolf et al., 2020) for inference. We used a single NVIDIA A100 GPU with 80GB memory, batch size 8 and maximum sequence length of 1024, with

[5]`https://ai.meta.com/resources/models-and-libraries/llama-downloads/`

greedy decoding. Within this setting, we were able to ensure exact reproduction of the experimental results across different runs. Generating paragraphs for one configuration (e.g., `3+A+I`, see below) takes 75 minutes with greedy decoding, totalling 30 hours on a single GPU for generating citation texts for all configurations in this paper. For NLI-based measurements, we use TRUE model based on T5-XXL[6] and the best reported model for SummaC[7].

B Full results table

For the sake of detail and reproducibility, Table 4 lists all measurements obtained in the main experiment.

C Dataset

C.1 Title List

List of related work titles used in dataset creation is as follows.

{*"related work", "related works", "previous work", "background", "introduction and related works", "introduction and related work", "background and related work", "background and related works", "previous related work", "previous related works", "backgrounds", "previous and related work", "previous and related works"*}

C.2 Cleaning and Post-processing

We performed several additional cleanup operations on the data. We removed instances with corrupted components e.g., abstract, metadata, citation mark, PDF parsing. We encountered papers that were published in different venues with the same title and abstract. To avoid ambiguity, such duplicates were removed. A small number of non-English papers were removed. We determine the length threshold as 40 tokens separated by whitespace for extracted paragraphs and 10 for citation sentences. Since the related work paragraph dataset and the example citation sentence dataset are connected, cleaning process was run in parallel for these datasets. For example, if there were no instances left for a cited paper after the cleanup, citation sentences for that paper were also removed from the example sentence pool.

Some cited paper's citation sentences are not included in the example sentence dataset. The main reason of this situation is cleaning procedure that we follow. For instance, corresponding sentences may not be segmented well or their length may be below the token threshold. To extract sentences from the paragraphs, the *scispacy*[8] module is employed. While determining the most similar example citation sentence, *all-MiniLM-L6-v2*[9] version of SBERT is utilized.

C.3 Column Descriptions

Column names along with their descriptions for the related work paragraph and the citation sentence datasets are given in Tables 7 and 8, respectively.

C.4 Dataset Staticstics

Tables 5 and 6 show core statistics for the resulting self-contained collection of related work paragraphs along with the respective papers that they cite and example citation sentences.

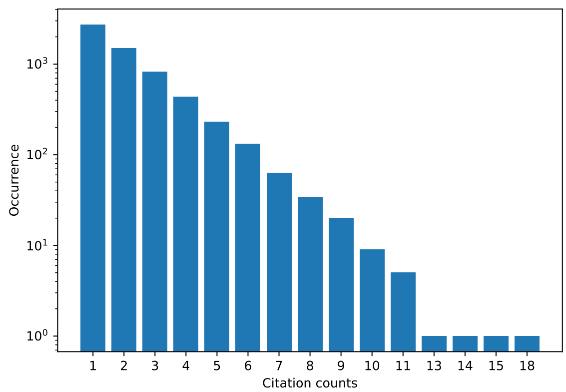

Figure 5: Citation count distribution in logarithmic scale

We also present distribution of the citation counts in the paragraphs in Figure 5. The number of paragraphs with larger number of citations decreases exponentially. Around 2,700 paragraphs include only one citation and the most crowded paragraph includes 18 citations. In the main paper experiment, we focus on the subset of paragraphs that include only one citation.

C.5 Intent Generation

While piloting the study, for intent generation we experimented with a range of LLMs such as LLaMA (7B) (Touvron et al., 2023a), Alpaca (7B) (Taori et al., 2023) and BLOOMZ (7.1B) (Muennighoff et al., 2023). The performance of FLAN-T5

[6] https://huggingface.co/google/t5_xxl_true_nli_mixture
[7] https://github.com/tingofurro/summac
[8] https://allenai.github.io/scispacy/
[9] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Configuration	Surface					Conventional			NLI	
	NG-1	NG-2	NG-3	PC	CM	ROUGE-L	BERTScore	BLEURT	TRUE	SummaC
1+A	61.48	37.38	26.70	1.50	30.69	14.63	84.77	38.92	10.98	20.77
1+A+I	59.81	34.88	24.09	1.48	41.62	15.98	85.19	40.58	14.09	20.87
1+A+E	63.07	37.94	26.97	1.64	74.36	15.13	84.97	39.13	13.45	20.83
1+A+I+E	61.29	35.57	24.56	1.63	77.54	16.53	85.41	40.78	17.46	20.95
2+A	64.78	37.02	26.04	1.08	63.07	15.56	85.32	40.19	12.33	20.77
2+A+I	64.94	37.21	26.11	1.11	91.30	17.38	85.81	41.66	17.33	20.99
2+A+E	64.52	34.94	23.74	1.11	82.87	15.99	85.52	40.66	12.02	20.82
2+A+I+E	64.79	35.89	24.52	1.15	89.71	17.36	85.89	41.82	16.91	21.00
3+A	61.52	36.09	25.37	1.31	37.56	14.81	84.93	39.66	6.86	20.71
3+A+I	61.24	36.29	25.54	1.32	28.42	16.61	85.39	40.55	11.97	20.88
3+A+E	64.01	38.30	27.33	1.48	76.25	15.23	85.06	39.69	8.42	20.80
3+A+I+E	63.28	37.91	26.93	1.47	75.52	16.90	85.51	40.82	15.13	20.98
4+A	62.03	35.18	24.30	1.01	54.55	15.88	85.26	40.88	10.68	20.70
4+A+I	61.58	35.36	24.35	1.02	42.73	17.07	85.52	42.13	10.98	20.83
4+A+E	64.61	37.85	26.61	1.03	82.07	16.03	85.33	40.88	10.51	20.76
4+A+I+E	63.31	36.48	25.18	1.05	78.56	17.32	85.69	42.10	13.35	20.86
5+A	63.41	40.21	30.04	1.40	25.95	15.38	84.94	39.70	10.02	20.80
5+A+I	61.17	37.29	27.02	1.56	30.74	16.56	85.18	40.47	12.93	20.89
5+A+E	63.85	38.96	28.42	1.58	76.99	15.94	85.20	39.93	10.96	20.88
5+A+I+E	61.73	36.92	26.45	1.77	76.20	17.02	85.38	40.83	12.80	20.97
6+A	62.98	34.84	23.55	1.01	92.55	15.88	85.28	40.69	7.23	20.68
6+A+I	64.60	38.19	26.81	1.07	85.90	17.03	85.56	41.12	10.10	20.83
6+A+E	64.79	36.37	24.88	1.07	95.34	16.06	85.35	40.85	9.33	20.75
6+A+I+E	65.99	38.83	27.24	1.10	95.77	17.39	85.70	41.52	10.18	20.81

Table 4: Main results. **#**Instruction + **A**bstract + **I**ntent + **E**xample. NG, PC, CM represent averaged n-gram overlap ratio, paragraph count and citation mark usage ratio. All values apart from PC given in percent (0-100) for readability.

Paragraphs	5,971
Total citation	12,950
Unique citing papers	4,605
Unique cited papers	6,620
Avg. occur. of a cited paper	1.96
Sentence count per paragraph	4.22
Word count per paragraph	98.67

Table 5: Related work paragraph dataset statistics

Sentences	73,139
Unique citing papers	16,338
Unique cited papers	6,594
Sentence per cited paper	11.05
Word count per sentence	35.30

Table 6: Example sentence dataset statistics

was deemed most acceptable and consistent among the models..

We conducted preliminary experiments for intent generation on a subsample of our dataset, exploring both zero-shot and few-shot configurations. In the zero-shot setting, we instructed the models to generate intent of the given target paragraph without showing any examples. In few-shot setting, we provided two-three paragraphs and their corresponding intents. To generate example paragraph-intent pairs, we conducted 100 zero-shot generations and manually selected six examples that successfully reflect the intent of the paragraph. We observed that in the few-shot setting the models tended to copy the examples into the output. Therefore, we decided to use the zero-shot setting as our final configuration. We use the following Flan-T5 prompt:

What is intention of the following paragraph?
{Target paragraph}

We investigated several decoding strategies to optimize generations such as greedy search, beam search, multinomial sampling, multinomial sampling with beam search and contrastive search wıth different hyperparameters. In the final setting, we opted for greedy decoding due to its output quality and reproducibility of the outputs.

C.6 Example intents

Below we provide a random sample of 20 machine-generated intents used in our study:

Column name	Description
acl_id	Unique ACL ID of the citing paper. Since a paper can have different related work paragraphs that satisfy conditions, there can be instances with the same acl_id. Although it is a unique identifier for distinguishing papers in ACL Anthology, this is not a unique identifier for this dataset. This rule is also valid for other citing paper meta features.
abstract	Abstract of the citing paper.
corpus_paper_id	Semantic Scholar ID of the citing paper.
pdf_hash	sha1 hash of the PDF.
numcitedby	The citing paper's citation count based on Semantic Scholar.
url	URL of the citing paper.
publisher	Publisher of the citing paper.
address	Address of the conference or venue.
year	The citing paper's publication year.
month	The citing paper's publication month.
booktitle	The name of the proceedings if it is a conference paper.
author	Authors of the citing paper.
title	Title of the citing paper.
pages	Page information of citing paper.
doi	DOI identifier of the citing paper.
number	Article number of the citing paper if it is a journal paper.
volume	Volume number of the citing paper if it is a journal paper.
journal	Journal name of the citing paper if it is a journal paper.
editor	Name of the editors if it is a journal paper.
isbn	ISBN number of the citing paper.
paragraph_xml	Citation paragraph with XML tags. It also includes other information about the citations relative to citing paper.
paragraph	Citation paragraph without XML tags. Like normal text in an article.
cited_paper_marks	This includes XML tags of target cited papers relative to citing papers. Identifiers are not absolute but relative. These tags also exist in paragraph_xml column. Since there can be multiple cited papers in the paragraph each mark is separated by " %%% " (space + 3 consecutive % + another space) .
cited_paper_titles	Titles of the cited papers separated by " %%% ".
cited_papers_acl_ids	acl_ids of the cited papers separated by " %%% ".
cited_papers_abstracts	Abstracts of the cited papers separated by " %%% ".

Table 7: Column names and descriptions for the related work paragraph dataset.

Column name	Description
example_id	Unique id of the example sentence instances. Its construction formula is acl_id of cited paper + "%" + extraction order number
sentence	Example sentence citing target cited paper.
paragraph_xml	XML version of the paragraph which example sentence belongs to. (From the related work section of the citing paper)
paragraph	Textual version of the paragraph which example sentence belongs to. (From the related work section of the citing paper)
citation_mark	This includes XML tags of target cited paper's citation marks.

Table 8: Column names and descriptions for the example citation sentence dataset. The dataset also includes metadata of the citing and the cited papers as given in Table 7.

- To describe the state of the art in WSD systems.
- To describe the Universal Dependency project.
- To provide a comparison of the pruning distances for dependency-based relation extraction models.
- To describe the work
- To describe the problem and the solution.
- To describe the crowdsourcing approach used to bootstrap YARN.
- Toxicity is a common problem in natural language generation, and a common source of model misbehavior.
- To describe the relation between Nominal SRL and SemEval.
- To provide a brief overview of the state-of-the-art in unsupervised structured prediction.
- To compare the performance of our approach with Yarowsky et al. (2001) and other related work.
- To introduce naive, linguistically motivated regularization methods such as sentence length, punctuation and word frequency.
- To provide a comparison of UDon2 and Udapi.
- To present a new technique for combining NMT models that is capable of addressing i and ii.
- To describe the work
- To describe a study.
- To provide a brief overview of the state of the art in multilingual representation learning.
- To describe the problem of query expansion
- To provide a brief review of the related works.
- To describe the state of the art in multilingual model evaluation.
- To describe an email thread summarization approach.

D Task instruction templates

Llama 2-Chat model takes prompts in two segments: *system prompt* and *user message*. System prompt is a fixed instruction for each session to guide the model how to react to user messages. User message contains additional information related to the instance at hand. In most cases we use system prompt to provide the task instruction, and use the user message to provide instance-specific data – Template 2 is an exception in that there input components are embedded into the user message, and system prompt remains empty. The following subsections exemplify the system inputs used in our work for the case where all input components are included into the instruction.

D.1 Template 1

System prompt: *Your aim is to generate an exactly single paragraph to be used in related work section in a main paper. You will be given the main paper's abstract and a relevant paper's abstract. The paragraph should reflect the intent and you need to refer the relevant paper in the same paragraph by using citation mark [REF#1]. You can inspire from the given example.*

Custom instance prompt: *Main paper abstract: {Citing paper abstract}*
Relevant paper abstract: {Cited paper abstract}
Intent: {Intent of the paragraph}
Example: {Example citation sentence}

D.2 Template 2

System prompt: -

Custom instance prompt: *Assume that you are the author of a paper whose abstract is as follows:*
{Citing paper abstract}
In your paper's related work paragraph, you want to cite a paper whose abstract is as follows:
{Cited paper abstract}
Intent of the related work paragraph should be as follows:
{Intent of the paragraph}
You can inspire from the given example:
{Example citation sentence}
How would you write an exactly one related work paragraph for this purpose? While citing use the citation mark [REF#1]. Your output must strictly consist of the related work paragraph only, nothing else.

D.3 Template 3

System prompt: *Follow given instructions:*
1-) You will be given main paper's abstract, a relevant paper's abstract, an intent and an example sentence.
2-) Write a related work paragraph that is belonging to main paper and citing relevant paper.
3-) The goal of your paragraph should be the given intent.
4-) You can utilize example sentence as how the relevant paper is cited before.

5-) Start your paragraph without any other explanations.
6-) Use [REF#1] as citation mark.
7-) Your output should consist of exactly single paragraph.

Custom instance prompt: *Main paper abstract: {Citing paper abstract}*
Relevant paper abstract: {Cited paper abstract}
Intent: {Intent of the paragraph}
Example: {Example citation sentence}

D.4 Template 4

System prompt: *You are writing a research paper and want to discuss another, related paper, with a certain intent – the purpose of the discussion. Generate exactly one paragraph of text that discusses the related paper in context of the main paper and follows the intent. You will be given the main paper abstract, the related paper's abstract, and the intent sentence. You can also utilize the given example sentence. Refer to the related paper by using a citation mark [REF#1]. You should generate exactly one paragraph of text, nothing else.*

Custom instance prompt: *Main paper abstract: {Citing paper abstract}*
Relevant paper abstract: {Cited paper abstract}
Intent: {Intent of the paragraph}
Example: {Example citation sentence}

D.5 Template 5

System prompt: *Imagine that you are a scientist writing a research paper. Your goal is to write a related work paragraph that discusses the related paper in context of your main paper. The related paper should be mentioned in the paragraph by using a citation mark [REF#1]. You will be given the main paper abstract, the related paper abstract, and the intent – the reason why you are citing the paper. An example sentence is also given to show how the related paper has been cited before. Your output should consist of exactly one paragraph of text and include the citation mark.*

Custom instance prompt: *Main paper abstract: {Citing paper abstract}*
Relevant paper abstract: {Cited paper abstract}
Intent: {Intent of the paragraph}
Example: {Example citation sentence}

D.6 Template 6

System prompt: *You are given two research papers: main paper and related paper. Generate one paragraph of text that discusses the related paper in the context of the main paper, given the intent – the reason why the main paper discusses the related paper. A citation sentence is also given to be taken as example. Use a citation mark [REF#1] to refer to the related paper. Your output should consist of exactly one paragraph of text and include the citation mark.*

Custom instance prompt: *Main paper abstract: {Citing paper abstract}*
Relevant paper abstract: {Cited paper abstract}
Intent: {Intent of the paragraph}
Example: {Example citation sentence}

E Human Evaluation

Table 9 shows an example of the human generation task for the configuration `A+I+E`. Table 10 shows an example of the human evaluation input: annotators first manually generated citation text paragraphs based on the prompt, and later manually ranked human and LLM generations in different settings using best-worst scaling. The citation texts were written in bulk first for a less informative prompt (abstract-only), then for a more informative prompt (abstract, intent and example). During ranking, the annotators would not rank their own generated outputs, and the configuration and the source of the text (human vs machine) were not known to the annotators.

Main paper abstract: The ACL shared task of DravidianLangTech-2022 for Troll Meme classification is a binary classification task that involves identifying Tamil memes as troll or not-troll. Classification of memes is a challenging task since memes express humour and sarcasm in an implicit way. Team SSN_MLRG1 tested and compared results obtained by using three models namely BERT, ALBERT and XLNet. The XL-Net model outperformed the other two models in terms of various performance metrics. The proposed XLNet model obtained the 3rd rank in the shared task with a weighted F1-score of 0.558.

Relevant paper abstract: This paper describes the work of identifying the presence of offensive language in social media posts and categorizing a post as targeted to a particular person or not. The work developed by team TECHSSN for solving the Multilingual Offensive Language Identification in Social Media (Task 12) in SemEval-2020 involves the use of deep learning models with BERT embeddings. The dataset is preprocessed and given to a Bidirectional Encoder Representations from Transformers (BERT) model with pretrained weight vectors. The model is retrained and the weights are learned for the offensive language dataset. We have developed a system with the English language dataset. The results are better when compared to the model we developed in SemEval-2019 Task6.

Intent: To describe the work done to identify the presence of offensive language in social media posts using BERT.

Example sentence: We have used CNN, RNN, LSTM, BERT and COLBERT models for offensive language detection in earlier SemEval workshop tasks `[OTH]`, `[REF#1]`,`[OTH]` in which BERT models provides better results than other machine learning and deep learning models.",

Table 9: Human generation task. Abstracts belong to papers (Hariprasad et al., 2022; Sivanaiah et al., 2020), respectively. The humans are tasked with writing one paragraph of citation text based on the inputs. The inputs vary depending on the condition – the example here is for the condition `A+I+E`, i.e. full information available. [OTH] used to denote citations to other papers than the cited paper in question.

Gold: In `[REF#1]`, we worked to identify the presence of offensive language in social media posts using BERT. Deep network model with BERT embeddings was found to achieve better F1 score when compared to 1D-CNN model trained with GloVe pretrained embeddings, 2D-CNN and BiLSTM models with Word2Vec embeddings.

a: The related paper by team TECHSSN for solving the Multilingual Offensive Language Identification in Social Media (Task 12) in SemEval-2020 `[REF#1]` is relevant to the main paper's discussion of Troll Meme classification. Both papers involve the use of deep learning models with BERT embeddings for identifying offensive language in social media posts. The related paper describes the development of a system for identifying the presence of offensive language in social media posts using BERT, and the results show that the BERT model outperforms other machine learning and deep learning models. This is similar to the main paper's finding that the XLNet model outperformed BERT and ALBERT in terms of various performance metrics for Troll Meme classification. The related paper also discusses the challenges involved in identifying offensive language in social media posts, such as the difference between profanity and hate speech, and the need for a system that can accurately detect and categorize offensive language.

b: Our work is related to the line of work in offensive language detection in social media. `[REF#1]` proposes a method for detecting offensive language and determining whether a post targets a particular person, based on fine-tuning a pre-trained BERT model. Unlike this work, our paper focuses on the task of troll meme classification, and experiments with other models: BERT, ALBERT and XLNet.

c: The related paper by team TECHSSN (2020) provides a relevant framework for identifying offensive language in social media posts, which is closely related to the task of troll meme detection in the main paper `[REF#1]`. Both papers employ deep learning models with BERT embeddings to classify text as offensive or not. However, while the main paper focuses on the binary classification of memes as troll or not-troll, the related paper targets the more general task of identifying offensive language in social media posts. The related paper also highlights the challenges of differentiating between profanity and hate speech, and discusses the differences between these two concepts. Overall, the related paper provides valuable insights into the use of BERT embeddings for text classification tasks, and demonstrates the effectiveness of this approach in the context of offensive language detection.

d: Prior work has shown that BERT can be successfully used to detect offensive language in social media posts, for example, `[REF#1]` fine-tune a pre-trained BERT model on an offensive language dataset, leading to improved performance. Following this insight, we experiment with three state of the art pre-trained models – BERT, ALBERT and XLNet – for the related task of troll meme classification.

Table 10: Human evaluation task. Masked settings are follows a: LLM-generated `6+A+I+E`, b: Human-generated `6+A`, c: LLM-generated `6+A`, d: Human-generated `6+A+I+E`. Humans are requested to select the best and the worst generation in terms of correspondence with the gold reference.