An Investigation of Prompt Variations for Zero-shot LLM-based Rankers

Shuoqi Sun[1*][0009−0000−9329−9731], Shengyao Zhuang[2][0000−0002−6711−0955], Shuai Wang[3][0000−0002−0726−5250], Guido Zuccon[3][0000−0003−0271−5563]

[1] RMIT University, Melbourne, Australia
shuoqi.sun@student.rmit.edu.au
[2] CSIRO, Australia
shengyao.zhuang@csiro.au
[3] The University of Queensland, St. Lucia, Australia
{shuai.wang, g.zuccon}@uq.edu.au

Abstract. We provide a systematic understanding of the impact of specific components and wordings used in prompts on the effectiveness of rankers based on zero-shot Large Language Models (LLMs). Several zero-shot ranking methods based on LLMs have recently been proposed. Among many aspects, methods differ across (1) the ranking algorithm they implement, e.g., pointwise vs. listwise, (2) the backbone LLMs used, e.g., GPT3.5 vs. FLAN-T5, (3) the components and wording used in prompts, e.g., the use or not of role-definition (role-playing) and the actual words used to express this. It is currently unclear whether performance differences are due to the underlying ranking algorithm, or because of spurious factors such as better choice of words used in prompts. This confusion risks to undermine future research. Through our large-scale experimentation and analysis, we find that ranking algorithms do contribute to differences between methods for zero-shot LLM ranking. However, so do the LLM backbones – but even more importantly, the choice of prompt components and wordings affect the ranking. In fact, in our experiments, we find that, at times, these latter elements have more impact on the ranker's effectiveness than the actual ranking algorithms, and that differences among ranking methods become more blurred when prompt variations are considered.

Keywords: LLM Rankers; Prompt Variations and Optimization

1 Introduction

Large Language Models (LLMs) are massively parametrised transformer models that have undergone training on a large extent of text [30]. Generative LLMs are capable of generating text in response to textual input (the prompt or context) [3]. This prompting facility has been used to instruct LLMs to perform specific tasks [3,27,28,34,5,33].

In this paper, we investigate the use of LLMs to create zero-shot rankers [13,22,23,31,18,34]. With zero-shot, we mean ranking methods that are not specifically trained for ranking tasks (i.e. not contrastive fine-tuning). These rankers operate by following the instructions provided in the prompt, which include the query and the k

---

[*] This work was conducted while Shuoqi Sun was a student at The University of Queensland.

documents that should be considered for ranking. Using the LLM, rankers then generate an answer to comply with the ranking instruction. Finally, either the generated answer contains the ranking provided by the method, or the logits of the answer are examined to infer a ranking.

Four families of zero-shot LLM rankers have been proposed: pointwise [31], pairwise [18], listwise [13,22,23], and setwise [34]. They differ because of the ranking algorithm (or mechanism) implemented in the instructions described in the prompt. For example, in pointwise the LLM is instructed to determine the relevance of a document, while in pairwise the LLM is instructed to determine which of two documents is more relevant. Within each family, one or more methods have been proposed. They typically differ in the backbone LLM, the wording of the prompts, and in how the ranking is formed once the answer is generated by the LLM.

Recent work has shown that, once these zero-shot LLM rankers are compared using the same backbone LLM and fixing how the ranking is derived (i.e. generation vs. logits), setwise methods are the most effective [34]. Depending on the dataset, pairwise and listwise methods are similarly effective, with pointwise methods providing lower effectiveness overall. While they did recognise that the use of different backbone LLMs in previous work biased the comparison between methods, they did not identify that the actual prompts used by the different rankers differed not just in terms of the words used to describe the ranking algorithm, i.e. the instruction associated with how scoring should be performed, but also in terms of "*accessory wording*" used. For example, compare the prompt of the PRP pointwise approach [18] and the one of the RankGPT listwise approach [22] – see Table 1: RankGPT's prompt includes also a "role-playing" component (in the system part of the prompt: "You are RankGPT [...]"), absent in PRP.

With this respect, then, we ask: *What is the effect of such differences in wording used in the prompts?* And, more broadly: Are differences in effectiveness due to the actual ranking algorithm, or they are due to the choice of words used in the prompts? Are differences due to LLM characteristics such as backbone and size? These are important questions because answering them will give us an understanding of what impacts the effectiveness of LLM rankers, and will influence how methods should be compared in future. We explore these directions of enquiry through a wide array of experiments, for a total of 1,248 prompts, in which we fix the backbone LLMs of the rankers, and we vary prompts in a controlled manner, categorising different prompt components and investigating their effects.

2   Related Work

Sensibility of LLMs to Prompt. Previous work has shown that LLMs are sensitive to the prompt formulation [9,25,10,8,2,15]. For example, Kim et al. [9] explored the effect of various prompts and prompting technology on deploying LLMs across language generation tasks, showing dependency between prompt and effectiveness. Similar findings were reported by Thomas et al. [25] when considering prompt variations in the context of relevance labelling. Kamruzzaman&Kim [8] explored the relation between prompting strategies and social biases in LLM outputs: reduced biases were observed in more complex prompts. The most comprehensive analysis of the effect of prompt variations has been recently provided by Mizrahi et al. across a range of NLP tasks [15]. These previous works, among others, have demonstrated that the structure and subtle nuances

Table 1: The PRP prompt (pointwise) [18] differs from that of RankGPT (listwise) [22] also because of accessory wordings (e.g., role-playing in RankGPT).

Method	Prompt
PRP [18]	Passage: {text} Query: {query} Does the passage answer the query?
RankGPT [22]	You are RankGPT, an intelligent assistant that can rank passages based on their relevancy to the query. I will provide you with num passages, each indicated by number identifier []. Rank the passages based on their relevance to query: {query}. {PASSAGES} Search Query: {query}. Rank the num passages above based on their relevance to the search query. The passages should be listed in descending order using identifiers. The most relevant passages should be listed first. The output format should be [] > [], e.g., [1] > [2]. Only response the ranking results, do not say any word or explain.

of prompt phrasing can dramatically impact the performance of the LLMs, across a wide range of tasks and contexts; however such an analysis does not exist for LLM rankers – our work is novel as it contributes a better understanding of this phenomenon in the specific context of LLM rankers.

Prompt Optimisation and Self-Optimisers. Strategic prompt design is not only beneficial but necessary to harness the full capabilities of LLMs. Recent studies have taken this further by investigating LLMs as self-optimisers [29,6,19]. These models utilize their generative capabilities to iteratively refine prompts, thereby enhancing their performance on downstream tasks. In our work we do not explore the adaptation of self-optimisers to prompts for LLM rankers: but our study motivates pursuing this as a future direction.

Zero-Shot LLM Rankers. We next review the four families of zero-shot generative LLM-based rankers: pointwise [31], pairwise [18], listwise [13,22,23], and setwise [34].

In pointwise, two approaches can be followed: *generation* [11,16] and *likelihood* [32,35]. In *generation*, LLMs are prompted with a query-document pair and asked to generate a binary answer ("yes"/"no") to the question of whether the document is relevant to the query: the likelihood of generating "yes" (extracted from the associated token logits) decides the ranking. In *likelihood*, a query likelihood model is involved in ranking. The LLM is prompted with the document and asked to produce a relevant query. Then, the ranking of documents is based on the likelihood of generating the provided query [20], which is obtained from the associated token logits. In our experiments with pointwise we implemented the generation approach, which is more commonly used in previous work.

Pairwise methods compare the relevance of two documents to a query, by prompting the LLM to answer which document is more relevant to the query. The ranking of documents is based on the relative preferences [18].

For listwise ranking a list of documents and a query is passed to the LLM via the prompt. LLMs are driven to generate the document labels of relevant documents in a certain order; the ranking then depends on this [13,22,17].

In setwise, a set of documents and the query are provided to the LLM, which is prompted to select the most relevant document, in an iterative way. Then sorting algorithms are used to obtain the ranking based on the preferences to documents' stopping occurs after k documents have been selected (if more than k were given as input) [34].

In common across all these methods is their use of LLM backbones in a zero-shot fashion, i.e. without the need for further training the LLM.

3   Methodology

Different strategies for using LLMs as rankers have been proposed; they not only differ in the ranking algorithm and backbone used, but also in the specific wording provided to the LLMs to perform the task. Next, we aim to collect the original prompts used in the literature, which will serve as the foundational prompt setting for our experiments. We then analyse the individual characteristics of each prompt, building a taxonomy of the components used in the prompts along with a list of instantiations used for each component across the different prompts (Section 3.1). With this information, we aim then to design experiments where we can systematically analyse the impact of components and specific wordings associated with components across different ranking algorithms and backbones. We do this by assembling prompt variations (Section 3.2). Experiment configurations are then outlined in Section 3.3.

3.1   Prompts Collection and Taxonomy of Ranking Prompt Components

We started by collecting the original prompts used in current zero-shot LLM-based rankers [31,18,13,22,23,34]. After the original prompts were collected, we analysed the prompts to identify high level components that are present in at least one original prompt, along with the associated variants. We also augmented the list of components and variants with other wordings we devised to explore specific categories further, e.g., tone words (see below). The analysis revealed five components:

- Evidence (EV): these are the query and the associated passages to rank.
- Task Instruction (TI): the instructions associated to the specific ranking strategy: these outline to the LLM the algorithmic steps to follow to produce a ranking. Example wordings include "which passage is more relevant" (pairwise) and "is this passage relevant to the query" (pointwise).
- Output Type (OT): the instructions that specify the format of the output the LLM needs to generate. For example, for pointwise ranking the LLM could be instructed to generated a Yes/No or a True/False answer.
- Tone Words (TW): words that express a positive, negative, or neutral connotation and that help express the attitude of the prompt author towards the ranking instruction, e.g., "please" or "you better get this right or you will be punished".
- Role Playing (RP): a description of the tool implemented by the LLM, used to make the LLM "impersonate" that role.

Note that TI and OT largely determine the ranker family that is implemented, while TW and RP can be generally applied to any ranker family and EV are always present. Table 2 lists the options we consider in our experiments for each of these components

Table 2: Type and wording alternatives of prompt components. None (Wording Alternative 0) refers to a prompt that does not use that component. A dash (−) means all prompts require that component and so the Wording Alternative 0 cannot be used (i.e. that component cannot be left empty). A checkmark means the prompt can be created without that component.

Component	Ranker	None (0)	Wording Alternatives				
			1	2	3	4	5
Task Instruction (TI)	pointwise	-	Does the passage answer the query?	Is this passage relevant to the query?	For the following query and document, judge whether they are relevant.	Judge the relevance between the query and the document.	-
	pairwise	-	Given a query, which of the following two passages is more relevant to the query?	-			
	listwise	-	Rank the {num} passages based on their relevance to the search query.	Sort the Passages by their relevance to the Query.	I will provide you with {num} passages, each indicated by number identifier []. Rank the passages based on their relevance to query.	-	
	setwise	-	Which one is the most relevant to the query.	-			
Output Type (OT)	pointwise		Judge whether they are "Highly Relevant", "Somewhat Relevant", or "Not Relevant".	From a scale of 0 to 4, judge the relevance.	Answer 'Yes' or 'No'.	Answer True/False.	-
	pairwise	-	Output Passage A or Passage B.	-			
	listwise	-	Sorted Passages = [	The passages should be listed in descending order using identifiers. The most relevant passages should be listed first. The output format should be [] >[], e.g., [1] >[2].	-		
	setwise	-	Output the passage label of the most relevant passage.	Generate the passage label.	Generate the passage label that is the most relevant to the query, then explain why you think this passage is the most relevant.	-	
Tone Words (TW)	All	✓	You better get this right or you will be punished.	Only output the ranking results, do not say any word or explanation.	Please	Only	Must
Role Playing (RP)	All	✓	You are RankGPT, an intelligent assistant that can rank passages based on their relevancy to the query.	-			

(but EV, since it is always the same). Some options refer to variations found in an original ranking prompt formulation; we augmented these with wordings we devised to explore additional alternatives for some of the components.

In addition to these components, our analysis of existing ranking prompts identified different approaches in the ordering of some of the components within the prompts; in particular:

- Evidence Ordering (EO): the relative ordering of the query and passage(s) provided to the LLM – whether the query is given first, followed by the passage(s), which we label as QF, or vice versa, passage(s) followed by the query (labelled PF).

- Position of Evidence (PE): instruction to specify the position of the evidence in the prompt – at the beginning (B) or at the end of the prompt (E).

Table 3: Prompt templates that combine the five components with the four ordering options available. Q and P denote query text passage(s) text, respectively.

EO/PE	B	E
QF	RP+ TI (Q) + P + TW+ OT	RP+ TW+ OT + TI (Q) + P
PF	RP+ P + TI (Q) + TW+ OT	RP+ TW+ OT + P + TI (Q)

3.2 Building Prompt Variations

To explore the role of the identified components, their interactions, and the effect of specific wordings used to instantiate each of the components, we setup a large scale experiment where prompts with unique combinations of these aspects are built.

To build prompts, we first consider the ordering options we identified, and build a prompt template for each combination: this gives rise to four prompt templates, shown in Table 3.

Then, for each template, we consider all possible instantiations. The number of variations of wordings for the task instruction (TI) and output type (OT) components differ across families of rankers – thus giving rise to different number of prompt instantiations across each family. Overall, our experiments consider 1,248 prompt variations: 768 unique prompts for pointwise; 48 for pairwise; 288 for listwise; and 144 for setwise.

Finally note that we did minor modifications to the original prompts from previous works to improve their consistency: for example, some prompts enclosed the query in quotes, while others appended the query after a colon; other differences included the presence of multiple line breaks. We settled on adapting a unique format for these aspects. From preliminary experiments, we observed that these differences in prompt formatting resulted in non statistically significant differences in effectiveness: e.g., [18]'s original prompt on COVID and Llama 3-8B backbone obtained an nDCG@10 of 0.8014, while our adjusted prompt lead to an nDCG@10 of 0.7966.

3.3 Experiments Settings

To experiment with the LLM rankers and the prompt variations, we setup a two stage ranking pipeline in line with previous work. For the first stage, we use the BM25 implementation from Pyserini [12] to retrieve the top 100 documents for a query (k1=0.9, b=0.4). For the second stage, we use the LLM ranker to re-rank these 100 documents.

As LLMs backbones, we selected instruction-tuned checkpoints from the Flan-T5 family [4], Mistral 7B [7], and Llama3 8B [1]. These are popular and highly-performant open LLMs; we excluded close-source LLMs like GPT4 in our analysis because of the high costs associated with running our experiments on commercial APIs.

For Flan-T5 we considered checkpoints of different sizes: Large (783M), XL (2.85B) and XXL (11.3B). This allowed us to explore the role of backbone size in instruction following ability and ranking. We did not consider extending the experiments to larger sizes of the other backbones, e.g., Llama3 70B, because of the high computational costs associated with doing this systematically.

We used three datasets from the ir_datasets [14] python library: TREC Deep Learning (DL) 2019 and 2020 (43 and 48 queries respectively), and BeIR TREC COVID (50

queries) [26,24]. We performed our analysis using nDCG@10, the primary metric across these datasets, and tested statistical significance with a paired, two-tails t-test.

Flan-T5 models are constrained by a maximum input length of 512 tokens. By analysing the query and passage lengths within the evaluation datasets, we found that methodologies like setwise and listwise, which incorporate multiple passages within a single prompt, would exceed this limit, with listwise requiring the longest input length. To ensure equitable comparisons across different experimental setups, we uniformly truncated the inputs for all methods and LLM backbones to adhere to the 512-token constraint. Our determination of truncation lengths was based on a comprehensive analysis of the datasets employed. We calculated the length of the longest non-query-and-passage prompt instruction (pure instruction length) alongside the distribution of query and document lengths. Consequently, we standardized the query length to 20 words and the document length to 80 words. This decision was driven by the need to preserve the integrity of instructional content within the prompts, ensuring that the loss of such information would not compromise experimental consistency across different LLM backbones. In our datasets, while queries do not exceed 20 words, documents can reach up to approximately 120 words, with the majority being about 80 words. Although this truncation might result in some loss of document information, our focus was not on evaluating the absolute ranking performance but on ensuring equitable conditions for each ranker across different backbones. This strategy ensures that, even with the most extensive prompt variations and the shortest model input capabilities, no instructional wording is omitted.

4   Results Analysis

We base our empirical analysis along six main lines of enquiry that help us investigate the impact prompts have on LLM rankers, including what makes ranking prompts effective, how rankers respond to different prompts, how ranking methods truly compare at the net of variations in prompt wording, and the impact of LLM backbones.

4.1   Are there better prompts?

For each family of rankers, we compare the effectiveness of the original prompts with all other prompt variations. Results are displayed in Figure 1: for each family of rankers, the star symbols identify the original prompts, while the boxplot shows the distribution of effectiveness across all prompt variations. We found prompts that can achieve higher effectiveness than the original prompts across all cases (with most differences being statistically significant), with the exception of listwise and pairwise on COVID when the Llama 3 backbone is used. For the pairwise approach, only prompts for the FlanT5-XXL model on the COVID dataset show significant differences, likely due to the lower effectiveness variation inherent in the pairwise method. We also identify cases where the original prompt was the worst among those considered for the specific ranking family: for example for the listwise prompt evaluated on DL19 and DL 20 with the FlanT5-Large backbone. We present the actual nDCG@10 scores of original prompts and the best prompts in Table 4.

Fig. 1: Effectiveness (nDCG@10) of zero-shot LLM-based rankers across ranking methods, prompt variations, LLM backbones and datasets. Effectiveness achieved by original prompts for each method is marked with a star, and annotated with value.
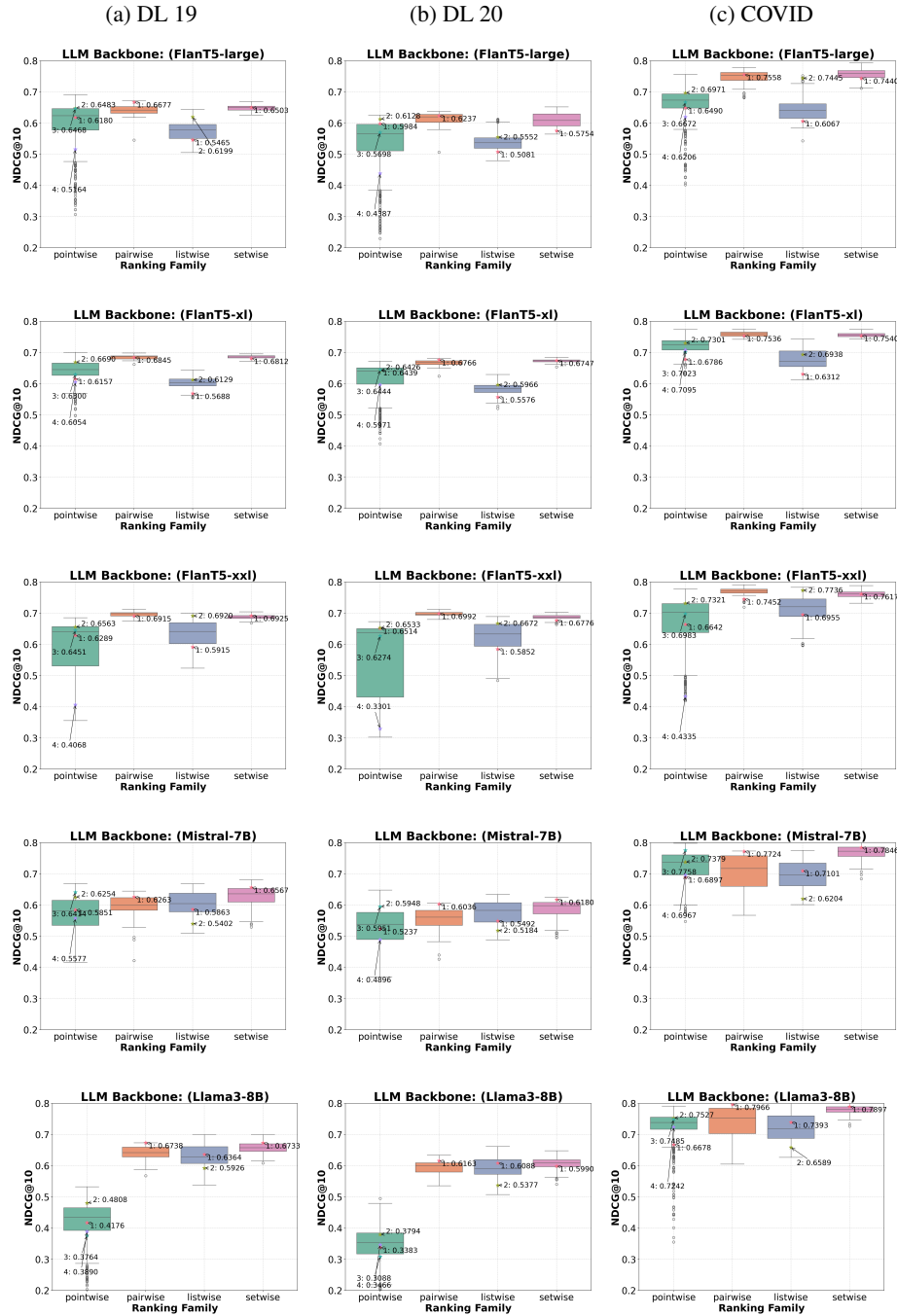

(a) DL 19   (b) DL 20   (c) COVID

Table 4: nDCG@10 across rankers, LLMs and datasets. *Po*: pointwise, *Pa*: pairwise, *Li*: listwise, *Se*: setwise. *Original*: the original (adapted) prompt. *Best*: the best prompt in our experiments. For each dataset/backbone – italics: the best performing method that used the original prompt; bold: the best performing method overall. Statistical significance differences (*: $p < 0.05$, †: $p < 0.01$) between the original prompt with the best prompt measured with two-tails paired t-test.

	Mistral-7B						Llama 3-8B					
	DL19		DL20		COVID		DL19		DL20		COVID	
Ranker	Original	Best	Original	Best	Original	Best	Original	Best	Original	Best	Original	Best
Po [18]	0.5851†		0.5237†		0.6897†		0.4176†		0.3383†		0.6678†	
Po [13]	0.6254†	0.6699	0.5948†	**0.6486**	0.7379†	0.7988	0.4808	0.5328	0.3794†	0.4950	0.7527	0.7916
Po [31].1	*0.6414*		0.5951*		0.7758		0.3764†		0.3088†		0.7485†	
Po [31].2	0.5577†		0.4896†		0.6967†		0.3890†		0.3466†		0.7242†	
Pa [18]	0.6263	0.6450	*0.6036*	0.6059	0.7724	0.7738	*0.6738*	0.6738	*0.6163*	0.6340	*0.7966*	0.8012
Li [13]	0.5863†	0.6686	0.5492†	0.6348	0.7101†	0.7758	0.6364†	**0.7004**	0.6088†	**0.6619**	0.7393†	**0.8123**
Li [22]	0.5402†		0.5184†		0.6204†		0.5926†		0.5377†		0.6589†	
Se [34]	0.6408*	**0.6811**	0.5988	0.6256	*0.7979*	**0.8053**	0.6654*	0.7000	0.6147*	0.6486	0.7933	0.8035

	FlanT5-Large						FlanT5-XL						FlanT5-XXL					
	DL19		DL20		COVID		DL19		DL20		COVID		DL19		DL20		COVID	
Ranker	Original	Best	Original	Best	Original	Best	Original	Best	Original	Best	Original	Best	Original	Best	Original	Best	Original	Best
Po [18]	0.6180†		0.5984*		0.6490†		0.6157†		0.6439*		0.6786†		0.6289†		0.6514		0.6642†	
Po [13]	0.6483†	**0.6918**	0.6128	0.6266	0.6971†	0.7570	0.6690*	**0.7010**	0.6426*	0.6727	0.7301†	0.7742	0.6563	0.6860	0.6533*	0.6733	0.7321†	0.7784
Po [31].1	0.6468†		0.5698†		0.6672†		0.6300†		0.6444*		0.7023†		0.6451*		0.6274†		0.6983†	
Po [31].2	0.5164†		0.4387†		0.6206†		0.6054†		0.5971†		0.7095†		0.4068†		0.3301†		0.4335†	
Pa [18]	*0.6677*	0.6724	*0.6237*	0.6382	0.7558	0.7788	0.6845	0.6986	*0.6766*	0.6823	0.7536	**0.7750**	0.6915	**0.7135**	*0.6992*	**0.7126**	0.7452†	**0.7917**
Li [13]	0.5465†	0.6443	0.5081†	0.6125	0.6067†	0.7521	0.5688†	0.6441	0.5576†	0.6300	0.6312†	0.7438	0.5915†	0.7011	0.5852†	0.6897	0.6955†	0.7834
Li [22]	0.6199		0.5552†		0.7445		0.6129*		0.5966†		0.6938*		*0.6920*		0.6672		*0.7736*	
Se [34]	0.6535	0.6693	0.6003†	**0.6525**	*0.7618†*	**0.7932**	*0.6859*	0.6959	0.6724	0.6855	*0.7543*	0.7745	0.6864*	0.7047	0.6679†	0.7036	0.7344†	0.7890

4.2 What are the characteristic of the best prompts?

Table 5 reports the optimal prompt components for various ranking methods and datasets. We identify notable patterns as below.

Task Instruction and Output Type: We analyse task instructions (TI) and output types (OT) together due to their inherent interrelation; OT largely depends on the ranking algorithm that the TI implements. We analyse these separately for each ranking family because instructions and output types vary across ranking families. We do not analyse pairwise here as for this method there are no alternative choices for these wordings.

Pointwise ranking does not show a consistent optimal choice; however, Task Instruction #3 is most prevalent, while Output Type #2 (judging relevance on a numerical scale) is seldom optimal.

Listwise ranking shows that the best task instruction varies by dataset and LLM backbone. Output Type #2 appears in 80% of the most effective configurations.

Setwise ranking studied with a single task instruction, finds Output Type #3 (label and explain the most relevant passage) as the most common among top-performing prompts, appearing in 53% of cases.

Tone Words: The inclusion of tone words does not present a clear trend, with the percentage of prompts achieving a higher effectiveness with no tone words or each of the five tone words distributed as follows: 18% (no tone words), 17%, 18%, 20%, 15%, and 12%. This indicates a relatively uniform influence of tone words choice on prompt

effectiveness. Furthermore, no consistent patterns were observed regarding the influence of LLM backbones or datasets on the effectiveness related to tone words. Nevertheless, including a tone word in the prompt led to increased effectiveness in 82% of the cases, underscoring the potential benefit of tone words in enhancing prompt performance.

Role Playing: Role playing often leads to the best effectiveness for pointwise and pairwise prompts (80% and 66% of the cases, respectively), it has mixed effects for setwise (is present in the best prompt about half of the times), while it is not associated with the best effectiveness for listwise (13%). Overall, 55% of the prompts with highest effectiveness include role playing wording. Role playing was originally only used by one ranking method, RankGPT [22], highlighting the difficulty in comparing different ranking methods if their prompt variations are not fairly explored.

Evidence Ordering: For pointwise ranking, presenting passage text before query text is preferred in 86% of top-performing prompts. The preference is less clear in other ranking types: 40% for pairwise; 40% for listwise; 46% for setwise where the best prompts have the passage text before the query text. Considering model backbones, Flan-T5 tends to perform best when presented with passage text before query text (66% of cases); while results are mixed for the other backbone models.

Position of Evidence: Among the best prompts, there tend to be an overall preference for prompts that provide the evidence at the beginning (before any other instruction): this is the case in 63% of the best prompts, with pointwise and listwise prompts exhibiting more often this pattern (73% and 67% respectively). Across all datasets, most best prompts for the FlanT5-XXL backbone have evidence at the beginning. This is also the case for Llama3-8B for DL19 and COVID and for Mistral-7B for COVID.

Summary. The analysis of the prompt templates associated with the prompts leading to the best effectiveness across all ranking families did not highlight any specific prompt wording combination that is more conducive of best effectiveness than others across all datasets and backbones – though we found that tone words and role playing are frequent among most of the best prompts. This analysis is further extended in Section 4.4 where the stability of rankers across prompts variations is considered.

4.3   Which ranking method is most effective?

Before our analysis of prompt variations, the only comparison of ranking methods across the four families within a consistent setting of datasets and backbone was provided by Zhuang et al. [34]. According to their results, the best performing rankers were setwise and pairwise (depending on dataset and backbone), followed by listwise and then pointwise, which were distinguishably worse.

Our analysis of prompt variations reveals a somewhat different picture. Consider the top results for each ranking family across datasets and LLM backbone (Figure 1). Similarly to previous findings, we also observe that overall pairwise and setwise methods deliver the best results. However, we find that pointwise can be as competitive as these previous methods if instructed with specific prompts. This is the case throughout all datasets and LLM backbones, with the exception of Llama 3. For Llama 3 on DL datasets, we observe that pointwise significantly underperforms other methods. We also observe that there are instances in which pointwise ranking can far exceed others: this is the case on DL19 when using FLanT5-Large and XL backbones (best pointwise nDCG@10

respectively 0.6918 and 0.7010, in most cases statistically significantly outperforming the others), and on DL20 when using Mistral-7B (best pointwise 0.6486).

Table 5: Prompt templates containing the components that lead to the most effective results across combinations of ranking method, LLM backbones, and dataset. The notation used corresponds to component definitions in Section 3.1 and options in Table 2 (e.g., TI_3 for *Task Instruction* with the *Third Option*).

(a) Dataset: DL19

Model	Pointwise	Pairwise	Listwise	Setwise
FlanT5-large	TI_3, OT_1, TW_0, PF, B, RP_1	TI_1, OT_1, TW_4, QF, B, RP_None	TI_3, OT_2, TW_2, PF, E, RP_None	TI_1, OT_3, TW_0, QF, B, RP_None
FlanT5-xl	TI_2, OT_3, TW_4, PF, E, RP_1	TI_1, OT_1, TW_2, QF, B, RP_1	TI_3, OT_2, TW_3, QF, B, RP_None	TI_1, OT_3, TW_3, QF, E, RP_1
FlanT5-xxl	TI_4, OT_3, TW_1, PF, B, RP_None	TI_1, OT_1, TW_0, QF, E, RP_1	TI_1, OT_2, TW_2, QF, B, RP_None	TI_1, OT_1, TW_4, QF, B, RP_1
Mistral-7B	TI_1, OT_4, TW_3, PF, E, RP_1	TI_1, OT_1, TW_0, QF, E, RP_1	TI_1, OT_1, TW_2, QF, B, RP_None	TI_1, OT_3, TW_0, QF, E, RP_1
Llama3-8B	TI_2, OT_3, TW_2, PF, B, RP_1	TI_1, OT_1, TW_0, QF, B, RP_None	TI_1, OT_2, TW_0, QF, B, RP_None	TI_1, OT_1, TW_3, PF, E, RP_None

(b) Dataset: DL20

Model	Pointwise	Pairwise	Listwise	Setwise
FlanT5-large	TI_3, OT_3, TW_3, PF, B, RP_1	TI_1, OT_1, TW_2, PF, E, RP_None	TI_3, OT_2, TW_1, PF, E, RP_None	TI_1, OT_1, TW_2, PF, E, RP_1
FlanT5-xl	TI_3, OT_3, TW_4, QF, B, RP_1	TI_1, OT_1, TW_5, QF, B, RP_1	TI_2, OT_2, TW_3, PF, E, RP_1	TI_1, OT_2, TW_5, PF, B, RP_None
FlanT5-xxl	TI_3, OT_1, TW_3, PF, B, RP_None	TI_1, OT_1, TW_2, PF, E, RP_1	TI_2, OT_2, TW_3, PF, B, RP_None	TI_1, OT_3, TW_3, PF, B, RP_1
Mistral-7B	TI_1, OT_4, TW_2, PF, E, RP_1	TI_1, OT_1, TW_1, QF, E, RP_None	TI_3, OT_1, TW_0, QF, B, RP_1	TI_1, OT_3, TW_2, QF, E, RP_1
Llama3-8B	TI_1, OT_3, TW_1, PF, B, RP_1	TI_1, OT_1, TW_3, PF, E, RP_1	TI_3, OT_2, TW_0, QF, B, RP_None	TI_1, OT_2, TW_4, PF, E, RP_None

(c) Dataset: COVID

Model	Pointwise	Pairwise	Listwise	Setwise
FlanT5-large	TI_3, OT_1, TW_1, PF, B, RP_1	TI_1, OT_1, TW_5, PF, B, RP_1	TI_3, OT_2, TW_4, PF, E, RP_None	TI_1, OT_1, TW_1, PF, B, RP_None
FlanT5-xl	TI_3, OT_1, TW_1, PF, B, RP_1	TI_1, OT_1, TW_5, PF, B, RP_1	TI_2, OT_2, TW_1, QF, E, RP_None	TI_1, OT_3, TW_5, QF, E, RP_None
FlanT5-xxl	TI_3, OT_1, TW_1, PF, B, RP_None	TI_1, OT_1, TW_0, PF, B, RP_1	TI_1, OT_2, TW_3, PF, B, RP_None	TI_1, OT_3, TW_5, PF, B, RP_1
Mistral-7B	TI_2, OT_2, TW_4, PF, B, RP_1	TI_1, OT_1, TW_3, QF, B, RP_1	TI_1, OT_1, TW_4, QF, B, RP_None	TI_1, OT_1, TW_2, QF, B, RP_None
Llama3-8B	TI_2, OT_3, TW_5, QF, E, RP_1	TI_1, OT_1, TW_4, QF, B, RP_None	TI_1, OT_2, TW_0, QF, B, RP_None	TI_1, OT_3, TW_1, QF, B, RP_1

4.4   Are ranking methods stable?

We consider the four ranking families independently, and study the variance of their effectiveness across all prompt variations for that family. Results are displayed in Figure 1. We observe that pointwise methods display the largest variability in effectiveness due to prompt variations, with some prompt variations delivering poor effectiveness. Setwise

and pairwise instead do better, displaying lower variability: setwise achieves this across all backbones and datasets investigated, while pairwise displays larger variability in specific conditions, e.g., when using Mistral, and for COVID when using Llama3.

Two key insights arise: 1) ranking algorithms are susceptible to prompt variations and may exhibit wide-ranging performance variations, and 2) different ranking algorithms have varying sensitivity to prompts, indicating the potential for some to mitigate the impact of these variations. This suggests that more comprehensive prompting optimisation, e.g., via self-prompting, may further improve effectiveness across all rankers.

4.5   Does the LLM size matter?

We answer this question by focusing on the FlanT5 models only, which come in three different sizes (first three rows of plots in Figure 1). We observe that in general larger models deliver higher effectiveness and reduced variations across the board. However, the pointwise family represents an exception. Improvements are observed when passing from FlanT5-large to FlanT5-XL. However when using FlanT5-XXL we observe both decreased effectiveness and large variance in effectiveness across the prompt variations.

4.6   Does the LLM backbone matter?

We compare three backbones across the results from Figure 1): FlanT5-XXL (11.3B parameters), Mistral-7B and Llama3-8B: the last two are of comparable size, while the first has approximately 40-60% more parameters. We observe that generally Llama3 and FlanT5 outperform Mistral-based rankers. Llama3 and FlanT5 have overall similar effectiveness, though on COVID Llama3-based rankers consistently outperform those with FlanT5. We also observe that Mistral-based rankers exhibit larger variance in effectiveness due to prompt variations than rankers based on the other backbones.

5   Further Discussion and Limitations

Latency is an important factor affecting the deployment of rankers in production. We did not consider query latency as one of the dimensions of analysis and comparison because our focus was on varying the prompts to measure ranking effectiveness. Zhuang et al. [34] provided a first insight into comparing latency among the zero-shot LLM rankers we considered. Latency of the methods and prompts we considered in our analysis would be similar to that performed there; however we note that some of the prompts we consider are longer than others (e.g., if the role playing component is added to prompts) – longer prompts result in increased query latency.

Overall, our experiments considered 1,248 prompt variations. While this is a large number, many more prompt variations could have been designed and investigated (and the space of prompt variations is virtually infinite). However, sensibly increasing the number of prompt variations in our experiments would have been infeasible as it would have exceeded our compute budget. We ran these prompts across three datasets, and using five different LLMs with up to 11B parameters. The execution of these experiments required a large amount of compute power; we ran experiments across three clusters: one with Nvidia A100 GPUs, another with Nvidia H100 GPUs, and a smaller one with Nvidia A6000 GPUs. Overall, the experiments took in excess of 12,400 GPU-hours to

execute. Varying prompt for the pairwise ranking results in large compute requirements compared to other methods because of the extensive number of pairwise computations (and thus LLM inferences) required by this method to answer a query. Although we only considered zero-shot LLM ranking approaches and thus did not conduct any LLM training, we recognize that our experiments may have still consumed substantial energy, thereby contributing to CO_2 emissions [21] and water consumption [36].

In our analysis we considered instruction-tuned checkpoints from three LLM backbones. A wider range of LLMs could have been considered, importantly including OpenAI's GPT models such as GPT3.5 and GPT4. GPT3.5 for example was found more effective than Llama 2 and Vicuna based LLM rankers in a limited set of experiments in previous work [34]. We were however restricted to using non-commercial models because of the high costs involved in performing our large number of experiments with commercial APIs. For example, using the cost estimates from [34], executing our experiments for pairwise, listwise and setwise across all considered prompts using GPT3.5 alone would have costed approximately USD $5,000.

6   Conclusions

Recent works have shown the promise of using generative LLMs to implement effective zero-shot re-rankers. Although distinctions among various methods primarily arise from the underlying ranking algorithms, further differences emerge based on the characteristics of the prompts employed to implement these algorithms. These characteristics are not associated with the actual ranking algorithm: e.g., they may be wordings related to a role-playing strategy, or tone words. In this paper, we analysed these prompts and mapped prompt wordings into a set of components to form prompt templates that allowed us to better understand the content of these prompts. We then performed a systematic analysis of prompt variations across different types of zero-shot LLM rankers. Our analysis revealed that ranking effectiveness varies considerably across different implementations of prompt components. Optimal prompt wording showed variability depending on the ranking method, dataset, and LLM backbone employed, suggesting that automatic prompt optimization, tailored to specific ranking methods and datasets, may be more effective than manual prompt engineering for optimizing ranking performance. We make code, runs and analysis available at `https://github.com/ielab/zeroshot-rankers-prompt-variations`.

References

1. AI@Meta: Llama 3 model card (2024), `https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md`
2. Anagnostidis, S., Bulian, J.: How susceptible are llms to influence in prompts? In: First Conference on Language Modeling (2024)

3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)

4. Chung, H.W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al.: Scaling instruction-finetuned language models. Journal of Machine Learning Research **25**(70), 1–53 (2024)

5. Fan, W., Zhao, Z., Li, J., Liu, Y., Mei, X., Wang, Y., Tang, J., Li, Q.: Recommender systems in the era of large language models (llms). arXiv preprint arXiv:2307.02046 (2023)

6. Guo, Q., Wang, R., Guo, J., Li, B., Song, K., Tan, X., Liu, G., Bian, J., Yang, Y.: Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. arXiv preprint arXiv:2309.08532 (2023)

7. Jiang, A.Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D.S., Casas, D.d.l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al.: Mistral 7b. arXiv preprint arXiv:2310.06825 (2023)

8. Kamruzzaman, M., Kim, G.L.: Prompting techniques for reducing social bias in llms through system 1 and system 2 cognitive processes. arXiv preprint arXiv:2404.17218 (2024)

9. Kim, J., Park, S., Jeong, K., Lee, S., Han, S.H., Lee, J., Kang, P.: Which is better? exploring prompting strategy for llm-based metrics. arXiv preprint arXiv:2311.03754 (2023)

10. Koopman, B., Zuccon, G.: Dr chatgpt tell me what i want to hear: How different prompts impact health answer correctness. In: Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. pp. 15012–15022 (2023)

11. Liang, P., Bommasani, R., Lee, T., Tsipras, D., Soylu, D., Yasunaga, M., Zhang, Y., Narayanan, D., Wu, Y., Kumar, A., et al.: Holistic evaluation of language models. arXiv preprint arXiv:2211.09110 (2022)

12. Lin, J., Ma, X., Lin, S.C., Yang, J.H., Pradeep, R., Nogueira, R.: Pyserini: A python toolkit for reproducible information retrieval research with sparse and dense representations. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2356–2362 (2021)

13. Ma, X., Zhang, X., Pradeep, R., Lin, J.: Zero-shot listwise document reranking with a large language model. arXiv preprint arXiv:2305.02156 (2023)

14. MacAvaney, S., Yates, A., Feldman, S., Downey, D., Cohan, A., Goharian, N.: Simplified data wrangling with ir_datasets. In: SIGIR (2021)

15. Mizrahi, M., Kaplan, G., Malkin, D., Dror, R., Shahaf, D., Stanovsky, G.: State of what art? a call for multi-prompt llm evaluation. Transactions of the Association for Computational Linguistics **12**, 933–949 (2024)

16. Nogueira, R., Jiang, Z., Lin, J.: Document ranking with a pretrained sequence-to-sequence model. arXiv preprint arXiv:2003.06713 (2020)

17. Pradeep, R., Sharifymoghaddam, S., Lin, J.: Rankvicuna: Zero-shot listwise document reranking with open-source large language models. arXiv preprint arXiv:2309.15088 (2023)

18. Qin, Z., Jagerman, R., Hui, K., Zhuang, H., Wu, J., Shen, J., Liu, T., Liu, J., Metzler, D., Wang, X., et al.: Large language models are effective text rankers with pairwise ranking prompting. arXiv preprint arXiv:2306.17563 (2023)

19. Sabbatella, A., Ponti, A., Giordani, I., Candelieri, A., Archetti, F.: Prompt optimization in large language models. Mathematics **12**(6), 929 (2024)

20. Sachan, D.S., Lewis, M., Joshi, M., Aghajanyan, A., Yih, W.t., Pineau, J., Zettlemoyer, L.: Improving passage retrieval with zero-shot question generation. arXiv preprint arXiv:2204.07496 (2022)

21. Scells, H., Zhuang, S., Zuccon, G.: Reduce, reuse, recycle: Green information retrieval research. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 2825–2837 (2022)

22. Sun, W., Yan, L., Ma, X., Ren, P., Yin, D., Ren, Z.: Is chatgpt good at search? investigating large language models as re-ranking agent. arXiv preprint arXiv:2304.09542 (2023)
23. Tang, R., Zhang, X., Ma, X., Lin, J., Ture, F.: Found in the middle: Permutation self-consistency improves listwise ranking in large language models. arXiv preprint arXiv:2310.07712 (2023)
24. Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., Gurevych, I.: Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. arXiv preprint arXiv:2104.08663 (2021)
25. Thomas, P., Spielman, S., Craswell, N., Mitra, B.: Large language models can accurately predict searcher preferences. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1930–1940 (2024)
26. Voorhees, E., Alam, T., Bedrick, S., Demner-Fushman, D., Hersh, W.R., Lo, K., Roberts, K., Soboroff, I., Wang, L.L.: Trec-covid: constructing a pandemic information retrieval test collection. In: ACM SIGIR Forum. vol. 54, pp. 1–12. ACM New York, NY, USA (2021)
27. Wang, S., Scells, H., Koopman, B., Zuccon, G.: Can chatgpt write a good boolean query for systematic review literature search? In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1426–1436 (2023)
28. White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., Schmidt, D.C.: A prompt pattern catalog to enhance prompt engineering with chatgpt. arXiv preprint arXiv:2302.11382 (2023)
29. Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q.V., Zhou, D., Chen, X.: Large language models as optimizers. arXiv preprint arXiv:2309.03409 (2023)
30. Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., et al.: A survey of large language models. arXiv preprint arXiv:2303.18223 (2023)
31. Zhuang, H., Qin, Z., Hui, K., Wu, J., Yan, L., Wang, X., Bendersky, M.: Beyond yes and no: Improving zero-shot pointwise llm rankers via scoring fine-grained relevance labels. In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) (2024)
32. Zhuang, S., Li, H., Zuccon, G.: Deep query likelihood model for information retrieval. In: Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II 43. pp. 463–470. Springer (2021)
33. Zhuang, S., Ma, X., Koopman, B., Lin, J., Zuccon, G.: PromptReps: Prompting large language models to generate dense and sparse representations for zero-shot document retrieval. In: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. pp. 4375–4391. Association for Computational Linguistics (2024)
34. Zhuang, S., Zhuang, H., Koopman, B., Zuccon, G.: A setwise approach for effective and highly efficient zero-shot ranking with large language models. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 38–47 (2024)
35. Zhuang, S., Zuccon, G.: Tilde: Term independent likelihood model for passage re-ranking. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 1483–1492 (2021)
36. Zuccon, G., Scells, H., Zhuang, S.: Beyond co2 emissions: The overlooked impact of water consumption of information retrieval models. In: Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval. pp. 283–289 (2023)