Unsupervised Contrast-Consistent Ranking with Language Models

Niklas Stoehr[1]* **Pengxiang Cheng**[2] **Jing Wang**[2]
Daniel Preoţiuc-Pietro[2] **Rajarshi Bhowmik**[2]
[1]ETH Zürich [2]Bloomberg
niklas.stoehr@inf.ethz.ch {pcheng134, jwang1621, dpreotiucpie, rbhowmik6}@bloomberg.net

Abstract

Language models contain ranking-based knowledge and are powerful solvers of in-context ranking tasks. For instance, they may have parametric knowledge about the ordering of countries by size or may be able to rank product reviews by sentiment. We compare pairwise, pointwise and listwise prompting techniques to elicit a language model's ranking knowledge. However, we find that even with careful calibration and constrained decoding, prompting-based techniques may not always be self-consistent in the rankings they produce. This motivates us to explore an alternative approach that is inspired by an unsupervised probing method called Contrast-Consistent Search (CCS). The idea is to train a probe guided by a logical constraint: a language model's representation of a statement and its negation must be mapped to contrastive true-false poles consistently across multiple statements. We hypothesize that similar constraints apply to ranking tasks where all items are related via consistent, pairwise or listwise comparisons. To this end, we extend the binary CCS method to Contrast-Consistent Ranking (CCR) by adapting existing ranking methods such as the Max-Margin Loss, Triplet Loss and an Ordinal Regression objective. Across different models and datasets, our results confirm that CCR probing performs better or, at least, on a par with prompting.

1 Introduction

"What is the correct ordering of the following countries by size: [USA, China, Russia, Canada, ...]?"

Language models have been shown to store plenty of facts and have powerful reasoning capacities (Petroni et al., 2019; Brown et al., 2020). Ranking tasks require both of these skills: multiple items have to be put in relation based on a comparison criterion. We are posing the question: what is the best approach to elicit a model's ranking knowledge and in-context ranking capacities without supervision? Knowing the answer to this question would allow us to uncover knowledge gaps, outdated information and existing biases before applying the language model. Once we trust a model, we could then put this best approach to action for solving in-context ranking tasks.

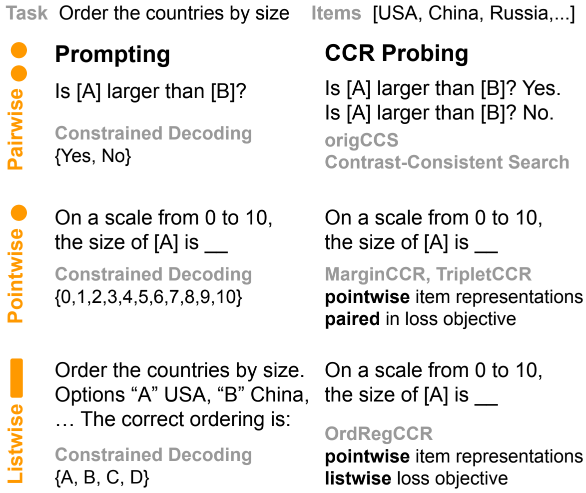

Figure 1: We study pairwise, pointwise, and listwise prompting and probing for unsupervised ranking.

A natural starting point for unsupervised ranking is prompting. In §2, we explore different task formulations: pairwise, pointwise and listwise prompting as outlined in Fig. 1. In the pairwise setting, any two items are compared and pairwise results are converted into a global ranking post-hoc. In pointwise prompting, the model assigns a score to each item individually. The listwise approach tasks the model to directly decode the entire ranking. For either approach, constrained decoding is essential to ensure the output can be converted into a ranking that includes all items. Yet, even with constrained decoding and calibration, we find that prompting often leads to inconsistent rankings.

*Work done during an internship at Bloomberg
github.com/niklasstoehr/contrast-consistent-ranking

Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 900–914

prompt type	template	prompting	CCR probing
ItemPair P	Is {item A} more in terms of rank crit than {item B}? X	constrain X to {Yes / No}	set X to {Yes / No}
ItemSingle S	{"optional": On a scale from 0 to 10,} The rank crit of {item} is X	constrain X to {0, 1,...,10}	set X to [MASK]
ItemList L	{"optional": context}. Order by rank crit. Options: "A" {item A}, "B" {item B}... .The correct ordering is: X	constrain X to {A, B, ...}	embed via ItemSingle then listwise loss

Table 1: We consider three different prompt types, ItemPair P, ItemSingle S and ItemList L, that all consist of a ranking criterion, a comparison token and one or multiple items to be ranked. ItemPair and ItemSingle can be used for prompting and CCR probing in a similar fashion. To realize listwise CCR probing, we first obtain individual vector representations of items via ItemSingle and then connect all items through a listwise loss objective.

For this reason, we turn to the model-internal representations of ranking tasks and their items in §3. We train a "probing model" with different unsupervised ranking objectives to find a latent ordering direction in the items' vector representations. Burns et al. (2023) recently proposed the Contrast-Consistent Search (CCS) method to find a direction in a language model's activation space that distinguishes truthful from false statements (Li et al., 2023a). This is achieved with a loss that imposes a logical constraint: the representation of a statement and its negation must be mapped to opposite (contrastive) poles.

Ranking tasks share similar properties: we can convert a ranking task into multiple pairwise comparisons and train a probe to find a "ranking direction" that allows ranking one item higher than the other consistently across all pairs. This has one significant advantage over the original CCS method for factual statements—instead of requiring a training set of multiple yes-no questions, we can source all pairwise permutations from a list of items which allows training the probe on a single ranking task.

We extend and adapt the binary CCS method to Contrast-Consistent Ranking (CCR) by exploring pairwise (§3.1), pointwise (§3.2) and listwise (§3.3) approaches as illustrated in Fig. 1. Pairing items in the prompt and obtaining the vector representations of all pairs is computationally expensive. Moreover, binary, contrastive poles may not be ideally suited for ranking tasks where the distances between items are not unit-length. Similar to the pointwise prompting approach, we instead embed each item individually, e.g., "The size of the US is [MASK], The size of China is [MASK], ...". We then pair the items represented by the activations of the [MASK] tokens in the loss function. In particular, we propose variants of the well-known Max-Margin and Triplet loss by including a *consistency* and *confidence* component. As a final adjustment, we mitigate the limitation that pairwise and pointwise objectives do not guarantee transitivity: item A may be ranked above B, B above C, but C above A, creating a circular contradiction. To address this, we introduce an unsupervised ordinal regression objective for listwise CCR probing.

Our experiments in §4 confirm that CCR probing outperforms prompting with a DeBERTa (He et al., 2021) and GPT-2 (Jiang et al., 2021) model across 6 datasets. Among the CCR probing methods, the Triplet Loss variant performs best on average. Even for a much larger MPT-7B (MosaicML, 2023) model, CCR probing performs at least on a par with prompting. Yet, CCR probing has the advantage of better control, reliability and interpretability as we discuss in §5.

2 Prompting for Rankings

Prompting is an accessible way to test a language model's ranking knowledge (Li et al., 2022a). We experiment with three different prompt types outlined in Tab. 1: pairwise, pointwise and listwise prompting (Qin et al., 2023). All prompt types contain at least one item to be ranked, a criterion to rank on, and what we refer to as comparison token. In every setting, we rely on some form of "constrained decoding" (for decoder-only) or "constrained mask-filling" (for encoder-only models). In essence, we restrict the vocabulary to a list of candidates and select the tokens with the highest-scoring logits.

Pairwise Prompting. ItemPair P: *Is {item A} more in terms of ranking criterion than {item B}? Yes / No*—Between any two items, the language model is tasked to make ranking decisions which are then converted into a ranking post-hoc as elab-

	prompt type	emb calls	loss / model	datapoints
CCR probing	ITEMPAIR P	$\mathcal{O}(N^2)$	ORIGCCS	$\mathcal{O}(N^2)$
	ITEMSINGLE S	$\mathcal{O}(N)$	ORIGCCS	$\mathcal{O}(N^2)$
	ITEMSINGLE S	$\mathcal{O}(N)$	MARGINCCR	$\mathcal{O}(N^2)$
	ITEMSINGLE S	$\mathcal{O}(N)$	TRIPLETCCR	$\mathcal{O}(N^3)$
	ITEMSINGLE S	$\mathcal{O}(N)$	ORDREGCCR	$\mathcal{O}(N)$
prompting	ITEMPAIR P	$\mathcal{O}(N^2)$	MLM / causal	$\mathcal{O}(N)$
	ITEMSINGLE S	$\mathcal{O}(N)$	MLM / causal	$\mathcal{O}(N)$
	ITEMLIST L	$\mathcal{O}(N)$	MLM / causal	$\mathcal{O}(1)$

Table 2: Complexity of each approach as a factor of the number of items N per ranking task. We distinguish between the number of required calls of an "embedding function" (i.e., a language model) and the number of resulting data points to be considered in a subsequent loss objective. The asymptotic complexity of permutations and combinations is both $\mathcal{O}(N^2)$.

orated in §4.3. Without calibration (Zhao et al., 2021), the model tends to always output the token most frequently observed during training, disregarding the task. Following (Burns et al., 2023), we compute the mean logit score of the "Yes" and "No" tokens in all pairwise prompts and then subtract the respective mean from each token's score.

Pointwise Prompting. ITEMSINGLE S: *On a scale from* 0 *to* 10, *the ranking criterion of {item} is* X—In pointwise prompting, the language model ranks one item at a time. If two items are assigned the same rank (i.e., the same candidate token from the list $X \in \{0, 1, 2, \ldots, 10\}$), we break the tie via sorting by the tokens' logit scores.

Listwise Prompting. ITEMLIST L *optional: context. Order by ranking criterion. Options: "A" {item A}, "B" {item B}... The correct ordering is:* X—For listwise prompting, we apply a step-wise approach: we let the model select the highest-scoring item from the list of candidates $X \in \{A, B, ...\}$, remove this token from the list and append it to the prompt. We repeat the process until the candidate list is exhausted. Importantly, the ordering of the candidate options in the prompt poses a "positional bias" (Han et al., 2023; Wang et al., 2023). Therefore, we randomly shuffle the ordering of the options and repeat the listwise prompting multiple times.

3 Unsupervised Probing for Rankings

Querying a language model's knowledge via prompting, we limit ourselves to prompt design and evaluating the tokens' logit scores. In contrast, probing accesses the information contained within a language model more directly by operating on latent vector representations. Conventionally, probing involves training a "diagnostic classifier" to map the vector representations of an utterance to a target label of interest (e.g., tense, gender bias, etc.) in a supervised fashion. The goal typically is to measure what information is contained within a language model (Alain and Bengio, 2016; Belinkov et al., 2017, *inter alia*). While the motivation of this work is closely related, we focus on an unsupervised probing variant and consider supervised probing only as a performance upper bound for validation purposes in §4.5 and §5.

Contrast-Consistent Search (CCS). Burns et al. (2023) propose Contrast-Consistent Search (CCS), an unsupervised probing method which seeks to train a probe to satisfy logical constrains on the model's activations. Instead of labels, CCS requires paired prompts in the form of yes-no questions:

$$\begin{aligned} x_i^+ &= \text{"Are elephants mammals? Yes"} \\ x_i^- &= \text{"Are elephants mammals? No"} \end{aligned} \tag{1}$$

Both statements x_i^+ and x_i^- are fed to a language model and the activations of the model's last hidden layer corresponding to the "Yes" and "No" token, $\boldsymbol{x}_i^+$ and $\boldsymbol{x}_i^-$ (bolded), are considered in subsequent steps. First, the vector representations $\boldsymbol{x}_i^+$ and $\boldsymbol{x}_i^-$ from different yes-no questions have to be Z-score normalized to ensure they are no longer forming two distinct clusters of all "Yes" and "No" tokens. Next, the paired vectors are projected to a score value s_i via the probe $f_\theta(\boldsymbol{x}_i) = \sigma(\boldsymbol{\theta}^T \boldsymbol{x}_i + b)$ which is trained using the ORIGCCS loss objective:

$$\begin{aligned} \text{ORIGCCS} = &\overbrace{\left(f_\theta(\boldsymbol{x}_i^+) - \left(1 - f_\theta(\boldsymbol{x}_i^-)\right)\right)^2}^{\textit{consistency}} \\ &+ \underbrace{\min\left(f_\theta(\boldsymbol{x}_i^+), f_\theta(\boldsymbol{x}_i^-)\right)^2}_{\textit{confidence}} \end{aligned} \tag{2}$$

ORIGCCS comprises two terms: the *consistency* term encourages $f_\theta(\boldsymbol{x}_i^+)$ and $f_\theta(\boldsymbol{x}_i^-)$ to sum up to 1. The *confidence* term pushes the scalars away from a deficient $f_\theta(\boldsymbol{x}_i^+) = f_\theta(\boldsymbol{x}_i^-) = 0.5$ solution, and instead encourages one to be close to 0 and the other to be close to 1. Thus, the ORIGCCS objective promotes mapping true and false statements to either 0 or 1 consistently, when the probe is trained on multiple yes-no questions.[1]

[1]CCS (and CCR) are direction-invariant, see App. A.

From Yes-No Questions to Rankings. ORIGCCS relies on logical constraints to identify a true-false mapping in the models' activations. We argue that ranking properties can similarly be expressed as logical constraints which are discernable by a probing model. In fact, the pairing of yes-no statements in Eq. (1) resembles the ITEMPAIR prompt type presented in Tab. 1.

One advantage of ranking tasks is that we can source many pairwise comparisons from a single ranking task which reduces the need for a training set of different yes-no questions. In the original CCS paper, it has been shown that a training set of as few as 8 pairwise comparisons can be enough for good test set performance. A ranking task of 8 items allows for 28 comparisons when considering all pairwise combinations and even 56 comparisons when considering all pairwise permutations.

We adapt binary CCS to Contrast-Consistent Ranking (CCR) by gradually modifying three components of the original method: in §3.1, we start by changing only the prompt. In §3.2, we explore pointwise CCR probing which requires changing the loss function in addition. Finally, in §3.3, we also alter the probing model to propose a listwise regression approach. Importantly, all CCR approaches are unsupervised and involve training a linear probing model whose number of parameters is held constant across settings to allow for a fair comparison.

3.1 Pairwise CCR Probing

Pairwise CCR probing for rankings is straightforward as we only need to change the binary prompt in Eq. (1) to the ITEMPAIR P prompt type in §3.1, but apply the original ORIGCCS objective (Eq. (2)), which we abbreviate as "ORIGCCS (P)".

3.2 Pointwise CCR Probing

We observe several methodological shortcomings of the pairwise CCR probing approach based on ORIGCCS that we address in the following. We start with the observation that it is computationally expensive to "embed" all pairwise item permutations as depicted in Tab. 2. Instead, we propose to "embed" each item individually and to pair their representations in the subsequent loss objective. To this end, we consider the ITEMSINGLE S prompt type for CCR probing which requires much fewer "calls" of a language model, precisely as many as there are items in a ranking task:

$$\begin{aligned} x_{i,1} &= \text{"The size of \{country 1\} is [MASK]"} \\ &\dots \\ x_{i,N} &= \text{"The size of \{country N\} is [MASK]"} \end{aligned} \tag{3}$$

In the original CCS approach, one data point i is given by a binary yes-no question. Adapted to ranking, we denote a ranking task with i and index its N items with n. Since we never compare items between different ranking tasks, we omit the i index for simplicity in the following. Now, the probing model f_θ assigns a ranking score $s_n = \sigma(\boldsymbol{\theta}^T \boldsymbol{x}_n + b)$ directly to each item x_n. Scores s_n can then be directly plugged into the ORIGCCS objective, instead of $f_\theta(\boldsymbol{x}_i)$, resulting in "ORIGCCS (S)".

However, the ORIGCCS loss enforces a hard binary decision, while an important property of rankings is that the distances between items do not necessarily have unit length. This "ordinal property" is typically reflected by some notion of "margin" in existing ranking objectives such as the Max-Margin and Triplet loss. To incorporate this, we propose the MARGINCCR loss which represents a modification of the well-known Max-Margin loss.

$$\begin{aligned} \min\Big(&\max\big(0, (f_\theta(\boldsymbol{x}_n^A) - f_\theta(\boldsymbol{x}_n^B)) + m\big), \\ &\max\big(0, (f_\theta(\boldsymbol{x}_n^B) - f_\theta(\boldsymbol{x}_n^A)) + m\big)\Big) \end{aligned} \tag{4}$$

MARGINCCR enforces that x_n^A ranks higher or lower than x_n^B by at least a margin m which can be seen as a *confidence* property. Since there are no labels however, the probe has to figure out whether scoring x_n^A higher or lower than x_n^B yields better *consistency* and reduces the loss across all item pair permutations.

In a similar style, we can adapt the popular Triplet Loss to TRIPLETCCR. To simplify notation, we denote the distance $|f_\theta(\boldsymbol{x}_n^A) - f_\theta(\boldsymbol{x}_n^B)|$ between two items x_n^A and x_n^B as $d(x_n^A, x_n^B)$ and compute TRIPLETCCR according to:

$$\begin{aligned} \min\Big(&\max\big(0, d(x_n^C, x_n^A) - d(x_n^C, x_n^B) + m\big), \\ &\max\big(0, d(x_n^C, x_n^B)) - d(x_n^C, x_n^A) + m\big)\Big) \end{aligned}$$

Intuitively, the objective forces the "positive item" to be closer to a third item x_n^C, referred to as "anchor", than a "negative item", plus a *confidence* margin m. Yet, this is enforced without knowing

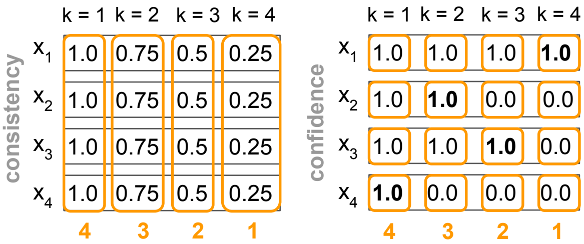

Figure 2: We translate the two aspects of *consistency* and *confidence* from the binary CCS objective to an ordinal multi-class setting resulting in ORDREGCCR.

which item is to be labeled as “positive” and “negative”. Instead, the probe is trained to make this decision by being *consistent* across all items in a given ranking task. We refer to both presented methods as “MARGINCCR (S)” and “TRIPLETCCR (S)” and provide further technical details on batching and vector normalization in App. A.

3.3 Listwise CCR Probing

Pairwise methods are not guaranteed to yield transitivity-consistent rankings: item A may win over B, B may win over C, yet C may win over A, creating a circular ordering (Cao et al., 2007). To tackle this shortcoming, we design a listwise probing method with a loss objective that considers all items at the same time. Various existing ordinal regression methods are based on binary classifiers (Li and Lin, 2006; Niu et al., 2016; Shi et al., 2021), making them a natural candidate for a CCS-style objective that does not require more parameters. These methods often rely on the extended binary representation (Li and Lin, 2006) of ordered classes, where, for instance, rank $k = 3$ out of $K = 4$ would be represented as $[1, 1, 1, 0]$, as illustrated on the right side of Fig. 2.

We first obtain a vector representation $\boldsymbol{x}_n$ of item x_n using the ITEMSINGLE prompt type. Next, we consider the COnsistent Rank Logits (CORAL) model (Cao et al., 2020), which offers guarantees for rank-monotonicity by training a probe $f_{\theta,k}$ to map $\boldsymbol{x}_n$ to one of K ranks. The probe consists of the weight vector $\boldsymbol{\theta}^T$ and K separate bias terms b_k to assign a rank score s_n^k according to $s_n^k = f_{\theta,k}(\boldsymbol{x}_n) = \sigma(\boldsymbol{\theta}^T \boldsymbol{x}_n + b_k)$. In essence, for each item n, the CORAL probe outputs a vector of K scores. Scores are monotonically decreasing because the bias terms b_k are clipped to be monotonically decreasing as k grows larger. Predicting a rank in the extended binary representation thus comes down to $\hat{k} = 1 + \sum_{k=1}^{K} \mathbb{1}[s^k > 0.5]$.

In a listwise approach, all N items are to be jointly considered and assigned a rank k.[2] The predicted scores can thus be represented as a square $N \times K$ matrix as displayed in Fig. 2. We propose an unsupervised ordinal regression objective that encourages a unique rank assignment, which we term ORDREGCCR:

$$\overbrace{\sum_{k}^{K-1} \left(\left(K - (k-1)\right) - \sum_{n}^{N} s_n^k \right)}^{\textit{consistency}} + \underbrace{\sum_{n}^{N} \sum_{k}^{K} \min \left(s_n^k, \left(1 - s_n^k\right) \right)}_{\textit{confidence}} \quad (5)$$

For a ranking of $K = 4$ items, the *consistency* term encourages each column to sum up to 4, 3,..., 1 respectively, as visualized in Fig. 2. Yet, to avoid a degenerate solution, the *confidence* term enforces each score towards either 0 or 1.

When applying this “ORDREGCCR (S)” approach, there are two difficulties to overcome: firstly, we require the number of parameters of the probing model to be the same across different approaches to ensure a fair comparison. Secondly, we prefer training a probing model whose parameters are independent from the number of items of a given ranking task. To mitigate both issues, we parametrize the K bias terms via a polynomial function as elaborated in App. A. This function, in turn, is parametrized by only two parameters, α and β, which are optimized during training.

4 Experimental Design

4.1 Language Models

We evaluate the prompting and CCR probing methods on an encoder-only and a decoder-only model. For the encoder-only model, we choose `deberta-v1-base` (He et al., 2021) which has 100 million parameters and is the best-performing encoder-only model for answering yes-no questions in the original CCS paper. For the decoder-only model, we consider `gpt2` (small) (Jiang et al., 2021) which has 124 million parameters. We compare these models against prompting results achieved with a much bigger, 7 billion parameter `mpt-7b` (MosaicML, 2023) model.

[2] We note that the number of ranks K equals the number of items N, but keep both letters for notational simplicity.

	dataset	tasks	avg. items	ranking example
fact-based	SYNTHFACTS	2	6.00	criterion: order the numbers by cardinality items: {1, 10, 100, 1000...}
fact-based	SCALARADJ	38	4.47	criterion: order the adjectives by semantic intensity items: {small, smaller, tiny, microscopic...}
fact-based	WIKILISTS	69	9.23	criterion: order the countries by size items: {Russia, Canada, China, United States...}
context-based	SYNTHCONTEXT	2	6.00	context: "Tom owns \$100, Jenny has \$1000,..." items: {Tom, Jenny, Emily, Sam...} criterion: order entities by wealth
context-based	REVIEWS	805	5.00	context: {A: I endorse this product..., B: The product is bad...} items: {review A, review B...} criterion: order the product reviews by stance
context-based	ENTSALIENCE	362	7.50	context: "The UN secretary met with climate activists..." items: {UN secretary, climate activists, US government...} criterion: order the entities by salience in the given text

Table 3: Overview of datasets, their number of ranking tasks and the average number of items per task. The first three datasets require knowledge of facts (fact-based), the latter three require in-context reasoning (context-based).

4.2 Ranking Task Datasets

We consider two types of ranking tasks with three datasets each. We denote the first type "fact-based" as solving it depends mostly on world knowledge. In contrast, the required information for the second type is provided "in-context". All datasets, displayed in Tab. 3, are publicly available and we discard all ranking tasks with fewer than four items and those including ties between items.

Fact-based Ranking Tasks. SYNTHFACTS: we manually conceive two synthetic ranking tasks with six items each. One task asks to rank adjectives based on sentiment, the other to rank numbers based on cardinality (App. Tab. 4). SCALARADJ: we consider rankings of scalar adjectives based on de Melo and Bansal (2013) and curated by Garí Soler and Apidianaki (2020), which are ordered by their semantic intensity, e.g., "small, smaller, tiny,...". WIKILISTS: we manually curate a dataset of 69 ranking tasks based on constant or rarely changing facts about the world and cap each task at 10 items at maximum (see App. Tab. 5).

In-Context Ranking Tasks. SYNTHCONTEXT: analogously to SYNTHFACTS, we design two synthetic in-context ranking tasks (App. Tab. 4). The first concerns ranking colors by popularity where the popularity is unambiguously stated in a prepended context. The second task is to order entities by their wealth as described in a prepended context. REVIEWS: We consider reviews and their ratings pertaining to the same product / company from the TrustPilot dataset (Hovy et al., 2015), particularly the US geo-coded version. ENTSALIENCE: As another in-context ranking task, we consider the Salient Entity Linking task (SEL) (Trani et al., 2016). Given a news passage, we ask the model to rank the mentioned entities by salience.

4.3 Evaluation Metrics

We are considering pairwise, pointwise and listwise approaches as displayed in Tab. 1. This means, we need to convert pairwise results to a listwise ranking and vice versa and consider evaluation metrics for pairwise as well as listwise results. Following the original CCS method, our evaluation is direction-invariant as further discussed in App. A. In essence, the ranking $A > B > C$ is considered the same as $C > B > A$.

Pairwise Metric and Conversion to Ranking. We rely on accuracy to evaluate pairwise comparisons. To account for direction-invariance, we reverse the predicted order if the reverse order yields better results. This means that accuracy will always be $\geq 50\%$. For aggregating pairwise results into a listwise ranking, we follow Qin et al. (2023): if an item wins a pairwise comparison it gains a point and points are summed to obtain a ranking. If the sum of wins is tied between items, we break the tie by considering the sum of the items' logit scores for all comparisons.

Ranking Metric and Conversion to Pairs. To evaluate rankings, we consider Kendall's tau correlation which is independent of the number of items per ranking task and the directionality of the or-

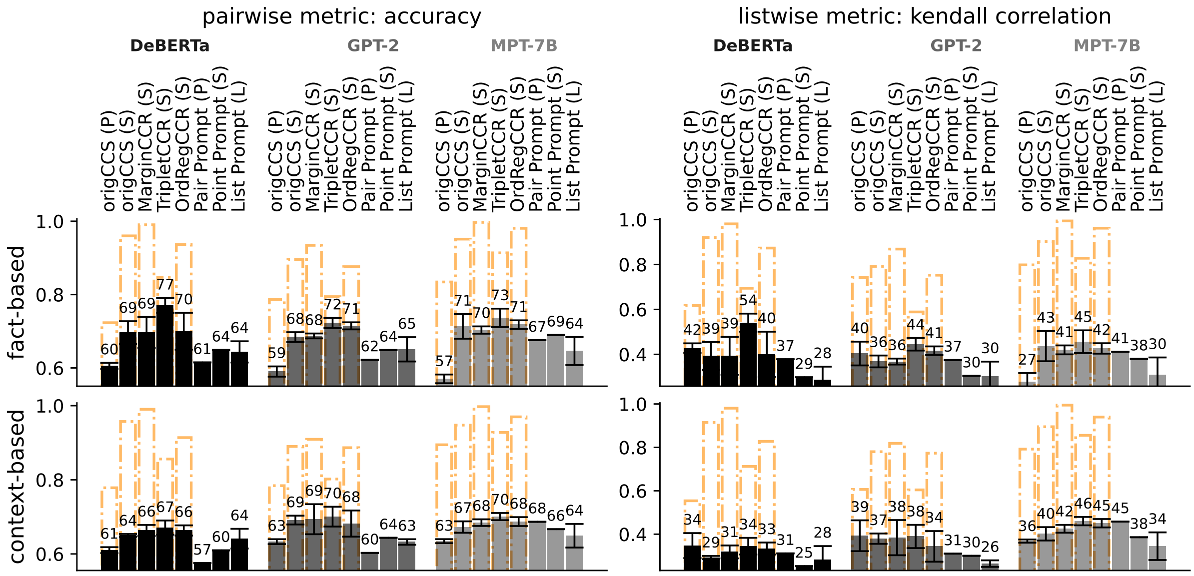

Figure 3: Pairwise and listwise results of the prompting and CCR probing methods for the DeBERTa, GPT-2 and MPT-7B model, meaned over all fact-based and context-based learning datasets. Results show mean and standard deviation over 5 runs. We find that CCR probing often outperforms prompting for the same-size model. Among the CCR probing methods, TRIPLETCCR is the best-performing. Orange bars represent ceilings of a supervised probe trained and tested on the same ranking task. As model size increases (MPT-7B), prompting performance improves.

dering. These desiderata are not given by other ranking and retrieval metrics such as the Normalized Discounted Cumulative Gain (NDCG) (Wang et al., 2013). A Kendall's tau of 0 represents the baseline of "no correlation" while 1 indicates an entirely correct ordering. We derive pairwise comparisons from a ranking by simply permuting and labeling any two items.

4.4 Supervised Ceilings

Both the prompting as well as CCR probing approaches can be applied in an unsupervised way, thus not requiring a train-test split. We also consider a supervised probe to obtain a performance upper bound that offers an indication on the difficulty of a task and the suitability of a certain prompt design. For instance, if a prompt is entirely random, even a supervised probe would not be able to discriminate between different items. For the supervised probe, we rely on the unaltered, original loss functions, e.g., Binary Cross-Entropy instead of ORIGCCS, Max-Margin loss instead of MARGINCCR, etc. (see Fig. 6 for an overview). Importantly, in §4.5, we do not consider a train-test split and thus train and test the supervised probe on the same ranking task. In §5, we consider a more traditional setting, where we train the probing model on ranking tasks that are distinct from the ones that we test it on.

4.5 Results

We present the results averaged over all datasets containing either fact-based or context-based ranking tasks in Fig. 3. All individual results are provided in Fig. 5 in the appendix. Most importantly, we find that CCR probing outperforms prompting for the smaller-size models, DeBERTa and GPT-2. For the much larger MPT-7B model, CCR probing and prompting yield narrower gaps in performance, potentially because of the stronger reasoning capabilities that boost the prompting performance of the larger models (Amini and Ciaramita, 2023). Among the CCR probing methods, TRIPLETCCR is the best performing approach across all models and datasets. The orange dashed lines represent the supervised ceilings for each of the CCR probing approaches as motivated in §4.4. Between the fact-based and context-based datasets, performance drops overall, but more for the encoder-only DeBERTa model. When considering the listwise metric, our results confirm that listwise prompting is inferior to pairwise and surprisingly also to pointwise prompting (Qin et al., 2023; Liusie et al., 2023). However, pairwise methods, here indicated with a P symbol, are also computationally more expensive, making CCR probing even more favorable. For pairwise methods, we observe a bigger discrepancy between the pairwise accuracy and listwise kendall correlation metric. This stems from the fact that pairwise methods are more fault-tolerant—some of the pairwise comparisons may be erroneous, but, in aggregate, the resulting ranking can still be correct. Similarly, we observe that listwise approaches (L) are generally more volatile, possibly due to more difficult calibration or positional biases (Han et al., 2023; Wang et al., 2023).

5 Discussion

To scrutinize our results, we explore settings with a train-test split, and discuss the interpretability considerations of CCR probing.

Ranking Direction across Tasks. Instead of training our probes on a single ranking task, we train them on a training set of multiple rankings and evaluate on a held-out set. To this end, we use 4-fold cross-validation which allows comparing CCR probing against supervised probing in a fair setup. This setup is more similar to the experiments in the original CCS paper (Burns et al., 2023) and thus rests on a similar hypothesis: is there are a more universal "ranking direction" in the activations of a language model that holds across ranking tasks? Fig. 6 in the appendix presents the results of this k-fold validation experiment. Firstly, our probes identify ranking properties that exist across different ranking tasks. This particularly holds for ranking tasks that resemble each other more closely as in SCALARADJ or REVIEWS. Secondly, CCR probing does not fall far behind supervised probing. Since this is especially evident for datasets with fewer ranking tasks, we hypothesize that CCR probing is less likely to overfit and instead exploits general ranking properties.

Interpretability. Besides performance, another argument for CCR probing is control and post-hoc interpretability offered by the parametric probe. In Fig. 4 for instance, we plot the ranking scores $s_n = \sigma(\boldsymbol{\theta}^T \boldsymbol{x}_n + b)$ for each item predicted by the linear probing model trained with TRIPLETCCR. This allows us to inspect the distances between items projected onto the latent ranking scale. The predictions and parameters are deterministic opposed to prompt-based generations from stochastic decoding methods. On a more abstract level, we relate multiple language model queries through a surrogate model that projects the language model's outputs to a shared ranking scale.

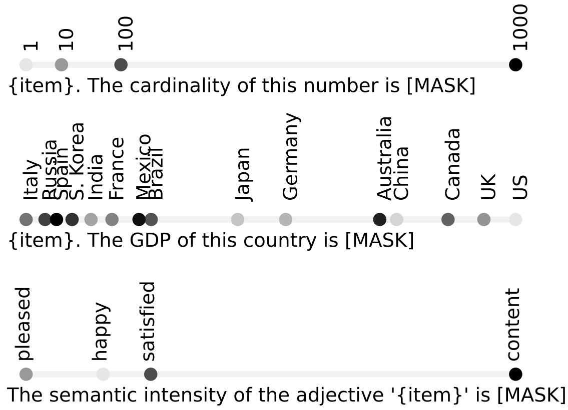

Figure 4: CCR probing offers interpretability benefits such as the post-hoc analysis of the probe's parameters. The gray scale hue of the individual dots represents the ground truth ranking of the respective items.

6 Related Work

This paper builds upon Contrast-Consistent Search (CCS) (Burns et al., 2023), which has inspired multiple other follow-up works: some explore calibrated versions of CCS (Tao et al., 2023), others adapt CCS to order-invariant, multi-class settings (Zancaneli et al., 2023), compare different CCS objective functions (Fry et al., 2023) and elicit inference-time interventions to increase truthfulness (Li et al., 2023a).

Farquhar et al. (2023) raise concerns that CCS may not "discover knowledge", but instead simply latches onto to most salient features. We argue that our CCR approach is less affected by this concern as we are mainly focused on achieving good predictive performance in unsupervised ranking tasks by making consistent measurements across multiple prompts. To this end, we test our method against regression-based variants in §3.3 and evaluate on an unseen held-out set in §5.

Pairwise and listwise prompting have been explored in different tasks (Ma et al., 2023; Lee and Lee, 2023; Liusie et al., 2023), but is most frequently focused on document retrieval (Ferraretto et al., 2023). Pairwise (RankNet) (Burges et al., 2005) and listwise (ListNet) (Cao et al., 2007) ranking approaches have also been compared outside of language model prompting. We additionally explore pointwise prompting (Fu et al., 2023) and find that, counter-intuitively, pointwise often outperforms listwise prompting. To move beyond prompting, we propose an expansion of the CCS method to rankings. CCS and CCR are conceptually different to "contrast consistency" which refers to contrastive data perturbations (Gardner et al., 2020; Zhang et al., 2023). They are also different to "contrastive decoding" (Li et al., 2023b) which contrasts log-probabilities between an expert and an amateur model. Instead, our CCR probing approach is strongly influenced by unsupervised ranking (Frydenlund et al., 2022) and probing of semantic, ordinal axes (Garí Soler and Apidianaki, 2020; Li et al., 2022b; Stoehr et al., 2023a,b).

7 Conclusion

We analyze the ranking capabilities of language models by comparing pairwise, pointwise and listwise prompting techniques and find that listwise prompting is less computationally expensive, but more susceptible to mistakes. We then propose an unsupervised probing method termed Contrast-Consistent Ranking (CCR). CCR learns an affine mapping between a language model's activations and a model-inherent ranking direction. Especially for smaller language models, CCR outperforms prompting, is easier to control, less susceptible to prompt design and more interpretable. We see a lot of potential in in-context probing for making consistent measurements with language models.

Acknowledgments

This work was completed while the first author, Niklas Stoehr, did a research internship at Bloomberg. We would like to thank Ozan Irsoy, Atharva Tendle, Faner Lin, Ziyun Zhang, Ashim Gupta, Suchin Gururangan, and the entire Bloomberg AI group for valuable feedback on the manuscript. We would like to express special thanks to Kevin Du and Luca Beurer-Kellner from ETH Zürich for early-stage discussions.

Limitations

We methodologically compare pairwise, pointwise and listwise prompting and CCR probing approaches as illustrated in Fig. 1. One may argue that our proposed versions of pointwise and listwise CCR probing violate this categorization because pointwise CCR uses a pairwise loss objective. Similarly, the loss objective of listwise CCR may be listwise, but the prompt type is ITEMSINGLE. To draw the distinction, we consider prediction time at which the probe trained with MARGINCCR or TRIPLETCCR outputs a single, thus pointwise, ranking score per item (see Fig. 4). Similarly, the probe trained with ORDREGCCR predicts a full vector of scores for all (listwise) ranks. Yet, we do encourage future work to explore further pointwise and listwise CCR probing approaches.

The direction-invariance of both CCS and CCR poses another potential limitation that may be lifted by future work as further outlined in App. A. In particular, for pointwise and listwise prompting, omitting the direction of a desired ranking can hurt performance. The language model may be confused whether to rank the highest or lowest item first, leading the items' corresponding logit scores to cannibalize each other. This weakness of prompting may be interpreted as a strength of CCR probing however, as it is less prompt-sensitive. An important direction for future work is testing prompting and CCR probing in ranking tasks with even larger or instruction-tuned language models.

Since we do not consider a train-validation-test set split in this work, we refrain from hyperparameter-tuning (e.g., margins, learning rate, sub-batching, probe initialization). However, based on initial prototyping, we see performance boosts for CCR when tuning these hyperparameters. We envision further boost in CCR probing performance through more expressive probing models, e.g., non-linear kernels or neural networks. Yet, the admissible number of probe parameters and the requirement to use the same probe for different ranking tasks irrespective of their number of items are limiting factors.

Impact Statement

Throughout this work, we evaluate language models in "transformative" rather than "generative" tasks—we avoid any free-form generation and strongly constrain a model's output to an explicit list of answer candidates. Moreover, the focus of this work lies on mitigating model hallucinations in the context of ranking. We pursue this goal in two ways: on the one hand, testing a model's parametric ranking-based knowledge may indicate knowledge gaps, outdated information or biases. On the other hand, constraining a model's output in in-context reasoning tasks leads to more consistent and thus more truthful ranking results. All datasets considered in this work are publicly available, but are in English only. We thoroughly checked all licensing terms and adhered to the intended use of the data, We also manually verified that the data do not contain personally identifiable information.

References

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv*, 1610.01644.

Afra Amini and Massimiliano Ciaramita. 2023. Probing in context: Toward building robust classifiers via probing large language models. *arXiv*, 2305.14171.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *ACL*, pages 861–872.

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.

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. 2005. Learning to rank using gradient descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 89–96.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations*.

Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. 2020. Rank consistent ordinal regression for neural networks with application to age estimation. *Pattern Recognition Letters*, 140.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: From pairwise approach to listwise approach. In *International Conference on Machine Learning*, pages 129–136. ACM.

Gerard de Melo and Mohit Bansal. 2013. Good, great, excellent: Global inference of semantic intensities. *Transactions of the Association for Computational Linguistics*, 1:279–290.

Sebastian Farquhar, Vikrant Varma, Zachary Kenton, Johannes Gasteiger, Vladimir Mikulik, and Rohin Shah. 2023. Challenges with unsupervised LLM knowledge discovery. *arXiv*, 2312.10029.

Fernando Ferraretto, Thiago Laitz, Roberto Lotufo, and Rodrigo Nogueira. 2023. Exaranker: Explanation-augmented neural ranker. *arXiv*, 2301.10521.

Hugo Fry, Seamus Fallows, Ian Fan, Jamie Wright, and Nandi Schoots. 2023. Comparing optimization targets for contrast-consistent search. *arXiv*, 2311.00488.

Arvid Frydenlund, Gagandeep Singh, and Frank Rudzicz. 2022. Language Modelling via Learning to Rank. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10636–10644.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. GPTScore: Evaluate as you desire. *arXiv*, 2302.04166.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323.

Aina Garí Soler and Marianna Apidianaki. 2020. BERT knows Punta Cana is not just beautiful, it's gorgeous: Ranking scalar adjectives with contextualised representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7371–7385.

Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. 2023. Prototypical calibration for few-shot learning of language models. *International Conference on Learning Representations*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTa: Decoding-enhanced BERT with disentangled attention. *International Conference on Learning Representations*.

Dirk Hovy, Anders Johannsen, and Anders Søgaard. 2015. User review sites as a resource for large-scale sociolinguistic studies. In *Proceedings of the 24th International Conference on World Wide Web*, pages 452–461.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? On the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, page 337.

Bruce W. Lee and Jason Lee. 2023. Prompt-based learning for text readability assessment. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1819–1824, Dubrovnik, Croatia. Association for Computational Linguistics.

Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. 2022a. Probing via prompting. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1144–1157.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023a. Inference-time intervention: Eliciting truthful answers from a language model. *arXiv*, 2306.03341.

Ling Li and Hsuan-tien Lin. 2006. Ordinal regression by extended binary classification. In *Advances in Neural Information Processing Systems*.

Lucy Li, Divya Tadimeti, and David Bamman. 2022b. Discovering differences in the representation of people using contextualized semantic axes. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023b. Contrastive decoding: Open-ended text generation as optimization. In *ACL*, pages 12286–12312.

Adian Liusie, Potsawee Manakul, and Mark J. F. Gales. 2023. Zero-shot NLG evaluation through pairware comparisons with LLMs. *arXiv*, 2307.07889.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. *arXiv*, 2305.02156.

NLP Team MosaicML. 2023. MPT-7B Language Model.

Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. 2016. Ordinal regression with multiple output CNN for age estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4920–4928.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 2463–2473.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv*, 2306.17563.

Xintong Shi, Wenzhi Cao, and Sebastian Raschka. 2021. Deep neural networks for rank-consistent ordinal regression based on conditional probabilities. *Pattern Analysis and Applications*, 26.

Niklas Stoehr, Ryan Cotterell, and Aaron Schein. 2023a. Sentiment as an ordinal latent variable. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*.

Niklas Stoehr, Benjamin J. Radford, Ryan Cotterell, and Aaron Schein. 2023b. The Ordered Matrix Dirichlet for state-space models. In *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*.

Lucas Tao, Holly McCann, and Felipe Calero Forero. 2023. Calibrated contrast-consistent search. *Stanford CS224N*.

Salvatore Trani, Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, and Raffaele Perego. 2016. SEL: A unified algorithm for entity linking and saliency detection. In *Proceedings of the 2016 ACM Symposium on Document Engineering*, pages 85–94. ACM.

Peiyi Wang, Lei Li, Liang Chen, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023. Large language models are not fair evaluators. *arXiv*, 2305.17926.

Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Tie-Yan Liu, and Wei Chen. 2013. A Theoretical Analysis of NDCG Type Ranking Measures. *COLT*.

Diego Zancaneli, Santiago Hernández, and Tomás Pfeffer. 2023. Adapting the contrast-consistent search method to multiclass classification. *Stanford CS224N*.

Zhihan Zhang, Wenhao Yu, Zheng Ning, Mingxuan Ju, and Meng Jiang. 2023. Exploring contrast consistency of open-domain question answering systems on minimally edited questions. *Transactions of the Association for Computational Linguistics*, 11.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *International Conference on Machine Learning*.

A Appendix

Direction-invariance of CCS and CCR. We limit the scope of this work to direction-invariant rankings: i.e., the ranking $A > B > C$ is considered to be the same as $C > B > A$. This assumption aligns well with the original Contrast-Consistent Search (CCS) method (Burns et al., 2023). In CCS, the probe is trained to map statements and their negation to either a 0 or 1 pole consistently across multiple paired statements. However, it is not defined a priori, which of the two poles corresponds to all truthful and all false statements. We argue that this is even less a shortcoming for CCR than it is for CCS. While the meaning of the poles, “true” versus “false” for CCS, “high rank” versus "low rank" for CCR, needs to by interpreted post-hoc, the ordering of items obtained with CCR can be directly read off. With ORIGCCS, the probe predicts the label of a new statement according to

$$s_i = \frac{1}{2}\Big(f_\theta(\boldsymbol{x}_i^+) - \big(1 - f_\theta(\boldsymbol{x}_i^-)\big)\Big) \quad (6)$$

In the case of MARGINCCR, TRIPLETCCR and ORDREGCCR, the probe directly predicts a ranking score s_n, because items are represented by individual vectors via the ITEMSINGLE prompt type.

Bias Terms for ORDREGCCR. The CORAL model (Cao et al., 2020) used in combination with the ORDREGCCR objective (§3.3) comprises K bias terms b_k. Since we would like to limit the number of parameters, we parametrize these bias terms via a polynomial function with learnable parameters α and β. We first cut a $[0, 1]$ interval into $K - 1$ unit-length pieces with the cut-off points $\{\delta_k\}_1^{K-1}$. We then transform these points through a polynomial function $g_{\alpha,\beta}$ as follows

$$\delta_k^{'} = g_{\alpha,\beta}\big(\delta_k^{(a-1)}(1 - \delta_k)^{(b-1)}\big) \quad (7)$$

The function g is parametrized by only two parameters α and β similar to the Beta function. As an uninformative prior, we set $\alpha = 1.0$ and $\beta = 1.0$ and optimize the parameters during inference. The transformed cut-off points $\delta_k^{'}$ are further shifted to ensure they are monotonically decreasing and centered around 0. To this end, we first compute the reverse (right-to-left) cumulative sum according to $\delta_k^{''} = \sum_{k=0}^{K-2} \delta_{K-k}^{'}$. Finally, we compute the mean $\bar{\delta}^{''} = \frac{\sum_1^{K-1} \delta_k^{''}}{K-1}$ which we subtract from every transformed $\delta_k^{''}$ to finally obtain b_k.

Technical Details. In all CCR probing setups, we dynamically set the batch size to the number of items of a ranking task. For the pairwise approaches, we perform sub-batching with two items at a time. For the approaches based on ITEMSINGLE, we Z-score normalize all vector representations in a batch. We set the margin $m = 0.2$ and include an additional positive margin term in TRIPLETCCR to avoid the anchor and positive item to collapse to the same value. We train all supervised and unsupervised probes using the Adam optimizer (Kingma and Ba, 2015) with its default settings for 200 epochs. Experiments were run on a MacBook Pro M1 Max (64 Gb) and a NVIDIA TITAN RTX GPU. We publish code and data at github.com/niklasstoehr/contrast-consistent-ranking.

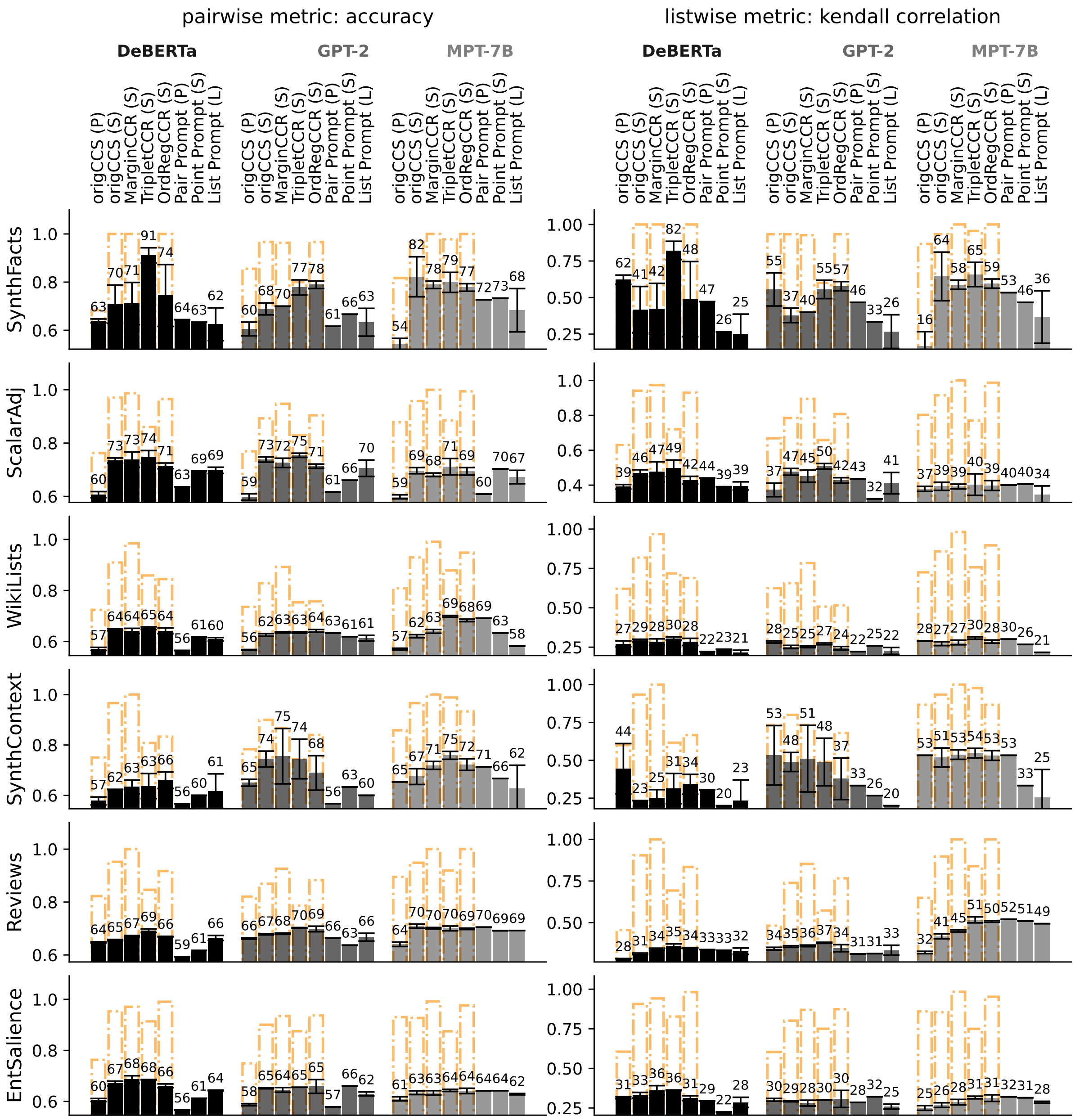

Figure 5: Mean ranking results and standard deviation for all methods and datasets over 5 runs.

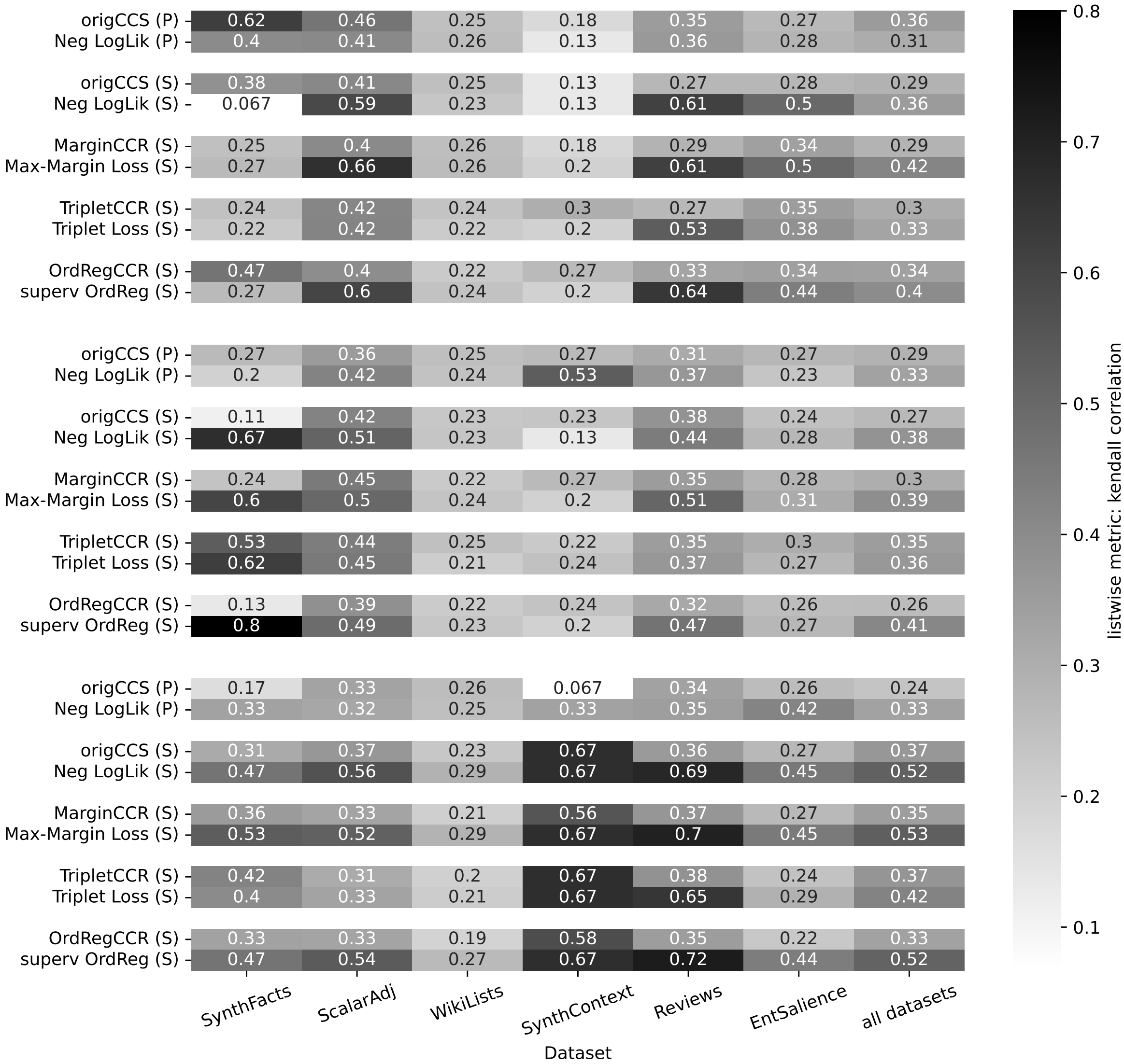

Figure 6: K-fold cross-validation results comparing unsupervised CCR probing and supervised probing.

SYNTHFACTS	
sentiment of the adjective	horrible, bad, okay, good, great, awesome
cardinality of the number	1, 10, 100, 500, 1000, 10000
SYNTHCONTEXT	
popularity of the color	context: Most students selected blue as their favourite color, followed by red, then yellow. Brown ranked lowest, green second lowest and purple third lowest; items: brown, green, purple, yellow, red, blue
wealth of people	context: An owns 100 dollar, Tom owns 50 dollars more and Sam 75 dollars more. Jenny is the richest owning 1000 dollar. Emily and Muhammad are at the lower end owning only 5 dollar and 10 dollars respectively. items: Emily, Muhammad, An, Tom, Sam, Jenny

Table 4: Details of our synthetic ranking task datasets SYNTHFACTS and SYNTHCONTEXT.

Buildings by volume	https://en.wikipedia.org/wiki/List_of_largest_buildings
Buildings by floor area	https://en.wikipedia.org/wiki/List_of_largest_buildings
Buildings by height	https://en.wikipedia.org/wiki/List_of_tallest_buildings
Airports by passenger traffic	https://en.wikipedia.org/wiki/List_of_busiest_airports_by_passenger_traffic
Museums by visitors	https://en.wikipedia.org/wiki/List_of_most-visited_museums
Tallest church buildings	https://en.wikipedia.org/wiki/List_of_tallest_church_buildings
Football stadiums by capacity	https://en.wikipedia.org/wiki/List_of_association_football_stadiums_by_capacity
Tallest statues	https://en.wikipedia.org/wiki/List_of_tallest_statues
Architectural Styles	https://en.wikipedia.org/wiki/Timeline_of_architectural_styles
Periods in art history	https://en.wikipedia.org/wiki/Periods_in_Western_art_history
Plays by Shakespeare by time	https://www.britannica.com/topic/list-of-plays-by-Shakespeare-2069685
Operas by Puccini by premiere date	https://en.wikipedia.org/wiki/List_of_compositions_by_Giacomo_Puccini
Most expensive paintings sold	https://en.wikipedia.org/wiki/List_of_most_expensive_paintings
Planets in the solar system by size	https://en.wikipedia.org/wiki/List_of_Solar_System_objects_by_size
Planets in the solar system by distance from the Sun	https://en.wikipedia.org/wiki/Solar_System
Moons of Jupiter by radius	https://en.wikipedia.org/wiki/List_of_Solar_System_objects_by_size
Heaviest terrestrial animals	https://en.wikipedia.org/wiki/Largest_and_heaviest_animals
Chemical elements by atomic number	https://en.wikipedia.org/wiki/List_of_chemical_elements
Chemicals by boiling point	https://en.wikipedia.org/wiki/Melting_point
Chemicals by melting point (highest to lowest)	https://en.wikipedia.org/wiki/Melting_point
Materials by hardness on Mohs scale	https://en.wikipedia.org/wiki/Mohs_scale
Countries by population	https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_population
US Counties by population	https://en.wikipedia.org/wiki/List_of_the_most_populous_counties_in_the_United_States
Capital cities by elevation	https://en.wikipedia.org/wiki/List_of_capital_cities_by_elevation
Metropolitan areas by size	https://en.wikipedia.org/wiki/List_of_largest_cities
Religions by followers	https://en.wikipedia.org/wiki/List_of_religious_populations
Ethnic groups by size in the US	https://en.wikipedia.org/wiki/Race_and_ethnicity_in_the_United_States
Countries by unemployment rate according to OECD	https://en.wikipedia.org/wiki/List_of_countries_by_unemployment_rate
Oil producing countries	https://en.wikipedia.org/wiki/List_of_countries_by_oil_production
GDP per capita	https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(nominal)_per_capita
Wine producing countries	https://en.wikipedia.org/wiki/List_of_wine-producing_regions
Largest power stations	https://en.wikipedia.org/wiki/List_of_largest_power_stations
Tourists for city	https://en.wikipedia.org/wiki/List_of_cities_by_international_visitors
Total energy from solar sources by country	https://en.wikipedia.org/wiki/Solar_power_by_country
Solar capacity as share of total energy consumption by country	https://en.wikipedia.org/wiki/Solar_power_by_country
Countries by size	https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_area
US Counties by area	https://en.wikipedia.org/wiki/List_of_the_largest_counties_in_the_United_States_by_area
US States by area	https://en.wikipedia.org/wiki/List_of_U.S._states_and_territories_by_area
Lakes by surface	https://en.wikipedia.org/wiki/List_of_lakes_by_area
Lakes by depth	https://en.wikipedia.org/wiki/List_of_lakes_by_depth
Rivers by length	https://en.wikipedia.org/wiki/List_of_rivers_by_length
Mountains by height	https://en.wikipedia.org/wiki/List_of_highest_mountains_on_Earth
Islands by surface area	https://en.wikipedia.org/wiki/List_of_islands_by_area
Volcanoes by height	https://en.wikipedia.org/wiki/List_of_volcanoes_by_elevation
Waterfalls by height	https://en.wikipedia.org/wiki/List_of_waterfalls_by_height
Caves by depth	https://en.wikipedia.org/wiki/List_of_deepest_caves
Oceans by area	https://en.wikipedia.org/wiki/Ocean
Oceans by coastline	https://en.wikipedia.org/wiki/Ocean
Oceans by average depth	https://en.wikipedia.org/wiki/Ocean
Deserts by area	https://en.wikipedia.org/wiki/List_of_deserts_by_area
Oceanic trenches	https://en.wikipedia.org/wiki/Oceanic_trench#Deepest_oceanic_trenches
Countries by area	https://en.wikipedia.org/wiki/List_of_countries_and_dependencies_by_area
Canyons by depth	https://www.worldatlas.com/canyons/10-deepest-canyons-in-the-world.html
Oldest reigning monarchs	https://en.wikipedia.org/wiki/List_of_longest-reigning_monarchs
Presidents of the US	https://en.wikipedia.org/wiki/List_of_presidents_of_the_United_States
Sultans of the Ottoman Empire	https://en.wikipedia.org/wiki/List_of_sultans_of_the_Ottoman_Empire
Emperors of Rome	https://en.wikipedia.org/wiki/List_of_Roman_emperors
Kings of Rome	https://en.wikipedia.org/wiki/King_of_Rome
List of time periods in history	https://en.wikipedia.org/wiki/List_of_time_periods
Platonic solids by number of faces	https://en.wikipedia.org/wiki/Platonic_solid
Best selling artists by albums	https://en.wikipedia.org/wiki/List_of_best-selling_music_artists
Songs with most weeks at number one on the Billboard Hot 100	https://en.wikipedia.org/wiki/List_of_Billboard_Hot_100_chart_achievements_and_milestones
Football teams by UEFA Champions League trophies	https://en.wikipedia.org/wiki/List_of_European_Cup_and_UEFA_Champions_League_finals
Most Ballon d'Or Trophies	https://en.wikipedia.org/wiki/Ballon_d%27Or
Countries with the most FIFA World Cup trophies	https://en.wikipedia.org/wiki/FIFA_World_Cup
Men's tennis players with the most grand slams won in the open era	https://en.wikipedia.org/wiki/List_of_Grand_Slam_men%27s_singles_champions
Olympic summer games host cities by year	https://en.wikipedia.org/wiki/List_of_Olympic_Games_host_cities
List of Dutch football champions by number of titles	https://en.wikipedia.org/wiki/List_of_Dutch_football_champions
List of Romanian football cup winners by number of titles	https://en.wikipedia.org/wiki/Cupa_Rom%C3%A2niei

Table 5: Ranking tasks (mostly extracted from Wikipedia) and curated for our WIKILISTS dataset.