# Extracting Conceptual Spaces from LLMs Using Prototype Embeddings

**Anonymous ACL submission**

## Abstract

Conceptual spaces represent entities and concepts using cognitively meaningful dimensions, typically referring to perceptual features. Such representations are widely used in cognitive science and have the potential to serve as a cornerstone for explainable AI. Unfortunately, they have proven notoriously difficult to learn, although recent LLMs appear to capture the required perceptual features to a remarkable extent. Nonetheless, practical methods for extracting the corresponding conceptual spaces are currently still lacking. While various methods exist for extracting embeddings from LLMs, extracting conceptual spaces also requires us to encode the underlying features. In this paper, we propose a strategy in which features (e.g. *sweetness*) are encoded by embedding the description of a corresponding prototype (e.g. *a very sweet food*). To improve this strategy, we fine-tune the LLM to align the prototype embeddings with the corresponding conceptual space dimensions. Our empirical analysis finds this approach to be highly effective.

## 1 Introduction

Conceptual spaces (Gärdenfors, 2000) are geometric representations of meaning, in which concrete entities are represented as vectors. Different from word embeddings in NLP, the dimensions of a conceptual space (typically) correspond to perceptual features. For instance, in a colour space, entities would be represented using three dimensions, corresponding to their hue, saturation and intensity. Conceptual spaces are used in cognitive science as theoretical models to explain phenomena such as analogy (Osta-Vélez and Gärdenfors, 2024), non-monotonic reasoning (Osta-Vélez and Gärdenfors, 2022) and concept learning (Douven, 2023). Within AI, the use of conceptual spaces has been advocated as an interface between neural and symbolic representations (Aisbett and Gibbon, 2001). As such, they can play an important role in explainable AI, for instance to enable interpretable classifiers (Derrac and Schockaert, 2015; Banaee et al., 2018; Bidusa and Markovitch, 2025) and computational creativity (McGregor et al., 2015). In practice, however, these applications have been hampered by the difficulty in learning conceptual spaces. Within cognitive science, most work has relied on spaces that are learned from human similarity judgments, for instance to study perception of colour (Douven et al., 2017), music (Forth et al., 2010), taste (Paradis, 2015) or smell (Jraissati and Deroy, 2021). Clearly, however, such a solution is not scalable enough for explainable AI.

A natural alternative is to try to construct conceptual spaces using NLP models, such as word embeddings or Large Language Models (LLMs). In fact, even within cognitive science, researchers have looked at NLP models as a promising route to obtain conceptual spaces in a cheaper way (Moullec and Douven, 2025). Starting from a pre-trained embedding space, it is often indeed possible to identify directions within that space that capture meaningful ordinal properties (Gupta et al., 2015; Derrac and Schockaert, 2015; Garí Soler and Apidianaki, 2020; Grand et al., 2022; Erk and Apidianaki, 2024). However, modelling *perceptual* features with traditional models has proven more challenging. This is intuitively due to the fact that many perceptual features are only rarely stated in text. For instance, Paik et al. (2021) highlighted how language models struggle with predicting colours, due to a divergence between the typical colour of an object and the distribution of co-occurring colour terms (e.g. the phrase “green banana” being more common than “yellow banana” in text). However, recent LLMs have proven more capable at modelling perceptual features, where promising results have been reported for colour (Liu et al., 2022a; Patel and Pavlick, 2022; Marjieh et al., 2024), taste (Kumar et al., 2024; Marjieh et al., 2024), touch

(Zhong et al., 2024a), smell (Zhong et al., 2024b) and sound (Marjieh et al., 2024), among others.

One problem that is not addressed by these works is how to *extract* conceptual spaces from LLMs. For instance, Kumar et al. (2024) prompt LLMs to make pairwise judgments (e.g. which is sweeter, banana or cucumber?), which only allows us to *rank* the entities along some conceptual space dimensions, without capturing how much the entities differ. Using pairwise comparisons is also intractable when dealing with thousands of entities. Marjieh et al. (2024) use LLMs to make pairwise similarity judgments, which is again too inefficient for constructing conceptual spaces at scale.

In a wider context, the problem of learning embeddings of text fragments using LLMs is well-studied (Reimers and Gurevych, 2019; Gao et al., 2021; Liu et al., 2021a; Wang et al., 2024; BehnamGhader et al., 2024; Lee et al., 2024). We therefore consider the following research question: is it possible to extract conceptual spaces directly from LLM-generated embeddings? Entity embeddings can straightforwardly be obtained using standard techniques. However, we also need to model the perceptual features. For instance, given an embedding *emb*("banana") of the word banana, how do we determine its level of sweetness? As already mentioned, previous work has shown that many features of interest can be modelled as directions in pre-trained embeddings. It is thus natural to assume that there exists a vector $\mathbf{v}_{sweet}$ such that *emb*("banana") $\cdot\, \mathbf{v}_{sweet}$ reflects the degree of sweetness of a banana. One possibility is to estimate this vector $\mathbf{v}_{sweet}$ from labelled examples, but such data is not readily available for most domains. Another possibility is to estimate the vector from seed words, i.e. examples of entities at both extremes of the ranking, but such directions can be unreliable, being highly sensitive to choice of seeds (Antoniak and Mimno, 2021; Erk and Apidianaki, 2024).

In this paper, we consider a simple alternative, which is to estimate the vector $\mathbf{v}_f$ encoding some feature $f$ as the description of a generic prototype. For instance, $\mathbf{v}_{sweet}$ could be modelled as *emb*("a very sweet food"). Unfortunately, with pre-trained LLM embedding models, the performance of this approach is sub-optimal, as the embedding of such a generic prototype description lies in a different subspace than the entities themselves (see Figure 1). We therefore propose a fine-tuning strategy, which encourages the embeddings of such descriptions to be aligned with the embeddings of the corresponding entities. We find that a small training set, synthetically generated using GPT-4o, is sufficient to achieve state-of-the-art results.

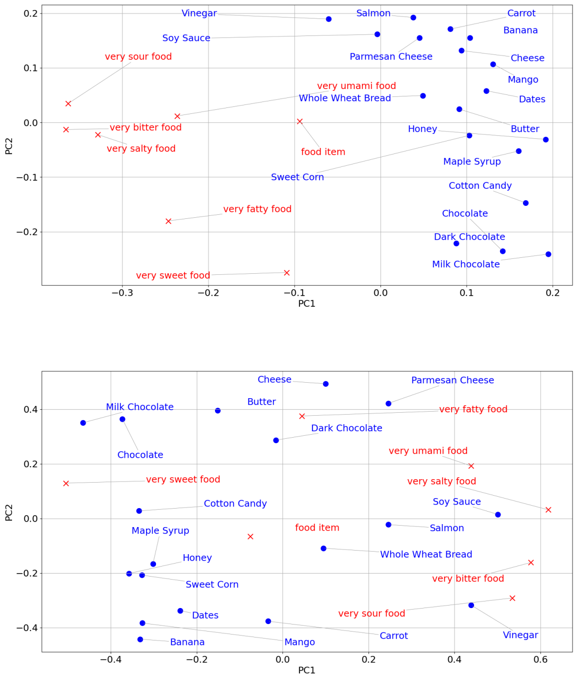


Figure 1: Embeddings of entities and prototypes in pre-trained LLM embedding models (top) and after fine-tuning (bottom), showing the first two principal components.

## 2 Related Work

The problem of learning entity embeddings using language models has received considerable attention, especially for bidirectional models of the BERT family (Devlin et al., 2019). For instance, a number of authors have proposed to represent entities by averaging the contextualised embeddings of their mentions in a corpus, using pre-trained (Ethayarajh, 2019; Bommasani et al., 2020; Vulić et al., 2020; Liu et al., 2021b) or fine-tuned (Li et al., 2023b) language models. However, approaches that directly extract embeddings based on the name of an entity have also been studied (Vulić et al., 2021; Liu et al., 2021a; Gajbhiye et al., 2022). Most relevant to our work, several authors have focused on predicting semantic and commonsense properties of concepts from their embeddings (Gajbhiye et al., 2022; Li et al., 2023b; Rosenfeld and Erk, 2023). For instance, Chatterjee et al. (2023) evaluated a BERT encoder that was fine-tuned to predict commonsense properties on the task of predicting taste dimensions such as sweetness, showing that their encoder was able to match the performance of GPT-3. Kumar et al. (2024) showed that a fine-tuned Llama 3 model is able to outperform BERT encoders. In this paper, we build on these results, aiming to extract embeddings from models such as Llama3, rather than using them for making pairwise judgments.

Compared to encoder-only models such as BERT, it is somewhat less straightforward to use decoder-only LLMs for embedding text. However, in recent years, several successful strategies have been proposed for fine-tuning LLMs to become general-purpose text embedding models (Wang et al., 2024; BehnamGhader et al., 2024; Lee et al., 2024), with the Massive Text Embedding Benchmark (Muennighoff et al., 2023) serving as a key driver. However, the focus of this benchmark is on sentence and paragraph level tasks, and little is currently known about the quality of LLM embedding models when it comes to representing entities. Our analysis in this paper partially addresses this gap, by comparing the quality of the conceptual space representations that are obtained by several recent models. LLMs can also be used to predict embeddings without fine-tuning. Jiang et al. (2024) suggested an Explicit One word Limitation (EOL) prompt, of the following form, for this purpose: *"This sentence: [text] means in one word:"*. We will also rely on prompts with this one-word limitation.

## 3 Methodology

**Problem Formulation** Let *emb* be an LLM-based embedding model, where we write $\textit{emb}(x) \in \mathbb{R}^n$ for the encoding of a phrase $x$. Let us furthermore assume that a set of entities $\mathcal{E}$ is given which all belong to some natural category. For instance, the entities in $\mathcal{E}$ could represent different types of food (e.g. banana, roast chicken, cake). For an entity $e$, we write $\gamma(e)$ for the verbalization of that entity, i.e. $\gamma(e)$ is a phrase that describes $e$. The entity $e$ can then be represented by its embedding $\textit{emb}(\gamma(e))$. We are interested in modelling semantic features of the entities based on these embeddings, where our focus is on perceptual features such as the sweetness of a food item or the intensity of an odour. Let $f$ be some real-valued feature, such that every entity $e \in \mathcal{E}$ has a corresponding feature value $f(e) \in \mathbb{R}$. We want to find an encoding $\tau_f : \mathbb{R}^n \to \mathbb{R}$ of the feature $f$ such that $\tau_f(\textit{emb}(\gamma(e))) \in \mathbb{R}$ corresponds to the feature value $f(e)$. We want to find the encoding $\tau_f$ without any supervision, other than a verbalization of the feature $f$, hence we cannot expect $\tau_f(\textit{emb}(\gamma(e))) = f(e)$, as there is typically no unique way to measure the degree to which a perceptual feature is satisfied. Instead, we want the *rankings* induced by the functions $f(.)$ and $\tau_f(\textit{emb}(\gamma(.)))$ to be as similar as possible.

**Embedding Entities** We experiment with two types of models: standard LLMs such as Llama-3 and pre-trained LLM-based embeddings models such as E5. The latter models can directly be used to obtain an embedding of $\gamma(e)$. To obtain embeddings with standard LLMs, we use a variant of the EOL trick from Jiang et al. (2024). Specifically, we use the following prompt:

*The description of the term '$\gamma(e)$' in one word is*

The embedding $\textit{emb}(\gamma(e))$ is then defined as the *normalized* encoding of the LLM for the last token. To verbalize the entity $e$, we observed that adding the name of the considered category leads to more informative embeddings for most models. For instance, we verbalize the entity *banana* as "food item banana" rather than "banana". This intuitively helps with resolving some ambiguities (e.g. orange as a fruit rather than a colour) and with specializing the embeddings to the domain of interest (e.g. strawberry as an odour rather than a food).

**Modelling Features** A common approach for modelling semantic features based on embeddings is to fit a logistic regression model (or a linear SVM) based on some training data. However, for most perceptual features, such training data is not readily available. Another common approach relies on a few examples of seed words $h_1, ..., h_p$ which are known to have a high value for the considered feature, and examples of seed words $l_1, ..., l_q$ which are known to have a low value. We can then estimate a vector $\mathbf{v}_f$ that models the considered feature $f$ based on this vectors, e.g.:

$$\mathbf{v}_f = \frac{1}{p}\sum_{i=1}^{p} \textit{emb}(\gamma(h_i)) - \frac{1}{q}\sum_{i=1}^{q} \textit{emb}(\gamma(l_i))$$

and $\tau_f(\mathbf{e}) = \mathbf{e} \cdot \mathbf{v}_f$. In principle, $\mathbf{v}_f$ can then be estimated from just two seeds words (i.e. $p = q = 1$). However, several authors have pointed out that this approach can be unreliable (Antoniak and Mimno, 2021; Erk and Apidianaki, 2024). For instance, if we have *banana* as the only example of a sweet food, then the resulting vector $\mathbf{v}_{\text{sweetness}}$ might capture the property of being yellow (in addition to, or instead of sweetness).

We pursue a different strategy, estimating the vector $\mathbf{v}_f$ by embedding a description $\gamma(f)$ of the feature $f$. Gajbhiye et al. (2022) trained a BERT bi-encoder based on this idea. Specifically, they fine-tuned two different BERT models, one for encoding entities and one for encoding properties, using a large dataset of commonsense properties. With LLMs, this bi-encoder strategy is not practical, as it doubles the memory requirement compared to fine-tuning a single model. We therefore embed entities and features using the same model. However, we still need to ensure that the embeddings of entities and features are aligned, i.e. $\mathit{emb}(\gamma(e)) \cdot \mathit{emb}(\gamma(f))$ should reflect the extent to which $e$ has the feature $f$. To this end, we verbalize $f$ as a generic description of a prototypical entity with a high value for the feature $f$. For instance, we can choose:

$$\gamma(\text{sweetness}) = \text{“a very sweet food”}$$

However, as illustrated in Figure 1, the embeddings of such generic descriptions are not in the same subspace as those of the entities. We therefore add a fine-tuning step, as we explain next.

**Fine-tuning Strategy** We fine-tune the embedding model *emb* to encourage the encoding of a generic property to be similar to the encoding of entities that have that property. For instance we want *emb*(“a tall mountain”) to be similar to *emb*(“Mount Everest”). To this end, we collected a small dataset using GPT-4o, consisting of information about 123 target properties. For each target property (e.g. *long river*), the dataset lists 7 examples of entities which have this property (e.g. *Nile*, *Amazon*, *Yangtze*), as well as 4 negative properties, which the entities do not satisfy. Of these negative properties, 3 are closely related to the target property (e.g. *short river*) and one is non-sensical for the considered entity type (e.g. *small city* when the entities are rivers).[1] We encourage the target property embedding to be close to the centroid of the seven examples and further from the negative properties. Specifically, we fine-tune the LLM by minimizing the following *classification loss*:

$$-\log \frac{\exp\left(\frac{\mathit{emb}(\gamma(f_0))\cdot \mathbf{c}}{T}\right)}{\sum_{k=0}^{4} \exp\left(\frac{\mathit{emb}(\gamma(f_k))\cdot \mathbf{c}}{T}\right)}$$

[1]Appendix B provides more details about the dataset.

where $\mathbf{c}$ is the centroid of entity embeddings, i.e., $\mathbf{c} = \frac{1}{7}\sum_{i=1}^{7} emb(\gamma(e_i))$, $f_0$ is the target property, and $f_1, \ldots, f_4$ are negative properties, with $T > 0$ a temperature parameter. We write $\mathcal{L}_1$ for the average classification loss across all target properties.

Note that the fine-tuning process explained thus far does not specifically focus on perceptual features, nor on the fact that we use the embeddings for ranking. Kumar et al. (2024) found that models which were fine-tuned on perceptual features generalized well to other, previously unseen perceptual features. As a secondary fine-tuning objective, we therefore also include the following *ranking loss*:

$$\sigma\left(-\alpha \cdot y_i \cdot [(\mathbf{e}_1 - \mathbf{e}_2) \cdot emb(\gamma(f))]\right)$$

where $y_i \in \{-1, +1\}$ indicates whether $e_1$ should rank above $e_2$ with respect to feature $f$, $\alpha$ is a scaling hyperparameter, $\mathbf{e}_1 = emb(\gamma(e_1))$, $\mathbf{e}_2 = emb(\gamma(e_2))$, and $\sigma$ denotes the sigmoid function. We write $\mathcal{L}_2$ for the average ranking loss across all entity pairs in our training set. The overall loss is then simply given by $\mathcal{L}_1 + \lambda\mathcal{L}_2$, where $\lambda$ is a hyperparameter.

## 4 Datasets

Following Kumar et al. (2024), we evaluate our approach on the following datasets:

**Taste:** a dataset, originally created by Martin et al. (2014), describing the taste of 590 food items, in terms of the following quality dimensions: sweetness, sourness, saltiness, bitterness, fattiness and umaminess. This dataset was first used for evaluating LLMs by Chatterjee et al. (2023), who rephrased some of the properties to make the more suitable for prompting. We use their cleaned version of the dataset.

**Rocks:** a dataset, originally created by Nosofsky et al. (2018), describing the physical appearance of 30 types of rocks, in terms of the following dimensions: lightness of colour, average grain size, roughness, shininess, organisation, variability of colour and density .

**Tag genome:** a dataset with human ratings of the extent to which a number of tags apply to different movies and books. Kumar et al. (2024) selected 38 tags for movies and 32 tags for books which can be viewed as ordinal features, all corresponding to adjectives (e.g. scary, quirky, suspenseful). The original movie ratings were obtained by Vig et al.

(2012), while the book ratings were obtained by Kotkov et al. (2022).

**Physical properties:** a dataset focused on three physical properties: mass, size and height. The data was originally created by Standley et al. (2017) and Liu et al. (2022b). It was used to evaluate LLMs by Li et al. (2023a) and subsequently cleaned by Chatterjee et al. (2023), who removed 7 items.

**Wikidata:** a dataset with 20 numerical features obtained from Wikidata, collected by Kumar et al. (2024) (e.g. the length of rivers, population of countries, and date of birth of people).

We will furthermore experiment on the following datasets, which have not yet been considered for evaluating LLMs, to the best of our knowledge:

**Odour:** a dataset of 200 odorants collected by Moss et al. (2016). A total of 103 participants rated odorants across nine dimensions. The authors reported that the following four were the most useful as normative data: familiarity, intensity, pleasantness, and irritability. We therefore also focus on these dimensions.

**Music:** a dataset of 364 music excerpts from different genres, collected by a panel of nine music experts (Strauss et al., 2024). The 517 participants rated the excerpts based on the emotions they felt, using the following dimensions from the Geneva Emotion Music Scale (GEMS) (Zentner et al., 2008): wonder, transcendence, tenderness, nostalgia, peacefulness, energy, joyful activation, sadness and tension.

# 5 Experiments

We refer to our proposed approach as *ProtoSim* (Prototype Similarity).[2] ProtoSim is clearly more practical than prompting LLMs to provide pairwise judgments, especially when large numbers of entities need to be ranked. Our main research question is whether or not the increased convenience of ProtoSim comes with a trade-off on performance.

## 5.1 Experimental Setup

**Models** We experiment with LLMs of different sizes and from different families: Llama3-8B (Dubey et al., 2024), Qwen3-8B and Qwen3-14B (Yang et al., 2025), Mistral-Nemo-12B, Mistral-Small-24B, OLMo2-7B, OLMo2-13B (OLMo et al., 2025) and Phi4-14B (Abdin et al., 2024). We furthermore experiment with the following pre-trained embedding models: E5-Mistral-7B (Wang et al., 2024), LLM2Vec-Llama3-8B, LLM2Vec-Llama3-8B-Sup, and LLM2Vec-Mistral-7B (BehnamGhader et al., 2024). We evaluate all models in two settings. First, we fine-tune the LLMs and pre-trained embedding models using the strategy from Section 3 (ProtoSim). Second, we fine-tune the LLMs as pairwise rankers, using the methodology from Kumar et al. (2024).

**Methodology** We evaluate the following variants of the fine-tuning strategy from Section 3. **Pre-trained**: we use the model without any fine-tuning. **Classification**: we only fine-tune the model with the classification dataset that was collected from GPT-4o (i.e. loss $\mathcal{L}_1$). **Rank-perc**: we only fine-tune on the ranking datasets (i.e. loss $\mathcal{L}_2$). As fine-tuning data, we use all perceptual datasets (i.e. Taste, Rocks, Odour, Music), apart from the dataset that is being evaluated. **Rank-full**: similar as before, but we fine-tune on all datasets (i.e. also on Tag Genome, Physical Properties and Wikidata), again excluding the dataset that is being evaluated. **Class + rank-perc**: use both the *Classification* and *Rank-perc* losses. **Class + rank-full**: use both the *Classification* and *Rank-full* losses. For the pairwise approach, only the ranking datasets can be used, i.e. *Rank-perc* and *Rank-full*. However, we also report results for pre-trained models with pairwise few-shot prompting. The prompts we used for this purpose are included in Appendix A.

**Benchmarks** The datasets discussed in Section 4 are used for both training and testing, using a leave-one-out strategy. In particular, when testing on a given dataset, we train on all the other datasets in the case of *rank-full* (and all the other perceptual datasets for *rank-perc*). Our experiments thus focus on the ability of the models to generalize to different domains than the ones they have seen during training. The *classification* dataset is open-domain, but this is a small dataset of 123 categories, which is not focused on perceptual properties and does not provide any information about ranking.

## 5.2 Results

**Comparing Fine-tuning Strategies** We first determine the best fine-tuning strategy for each approach. For this analysis, we use Llama3-8B

[2]All our code and preprocessed datasets will be shared upon acceptance.

| | Sweetness | Saltiness | Sourness | Bitterness | Umaminess | Fattiness | Average |
|---|---|---|---|---|---|---|---|
| ProtoSim (Llama3-8B) | | | | | | | |
| Pre-trained | 55.6 | 57.6 | 50.6 | 47.1 | 62.1 | 48.2 | 53.5 |
| Classification | 77.6 | 78.8 | **70.3** | **64.4** | 70.3 | 72.6 | 72.4 |
| Rank-perc | 77.9 | 75.3 | 56.5 | 55.9 | 68.5 | 63.2 | 66.2 |
| Rank-full | 73.2 | 70.6 | 53.2 | 51.2 | 63.8 | 72.1 | 64.0 |
| Class + rank-perc | **78.2** | **79.1** | 70.0 | 60.6 | **72.9** | 75.0 | **72.6** |
| Class + rank-full | 77.1 | 75.6 | 68.5 | 58.8 | 68.5 | **76.2** | 70.8 |
| ProtoSim (LLM2Vec-Llama3-8B-Sup) | | | | | | | |
| Pre-trained | 70.0 | 57.1 | 62.7 | 48.5 | 57.7 | 60.9 | 59.5 |
| Classification | 76.2 | 74.1 | **67.9** | **62.6** | **67.1** | 70.0 | 69.7 |
| Rank-perc | 75.0 | 76.8 | 58.2 | 55.3 | 64.7 | **70.9** | 66.8 |
| Rank-full | 72.6 | 72.9 | 55.9 | 51.8 | 58.2 | 70.3 | 63.6 |
| Class + rank-perc | **77.6** | **77.4** | 66.8 | 61.2 | 66.2 | 70.3 | **69.9** |
| Class + rank-full | 76.2 | 76.2 | 65.0 | 61.2 | 66.5 | 69.4 | 69.1 |
| Pairwise approach (Llama3-8B) | | | | | | | |
| Few-shot | 52.4 | 52.6 | 47.1 | 51.8 | 51.2 | 52.4 | 51.2 |
| Rank-perc | 55.3 | 62.9 | 56.8 | 55.3 | 52.1 | 57.4 | 56.6 |
| Rank-full | **79.7** | **71.5** | **62.7** | **62.1** | **63.5** | **72.1** | **68.6** |

Table 1: Comparison of different fine-tuning strategies (accuracy % on pairwise comparisons). The best results within each block are highlighted in bold.

(for the variants based on pre-trained LLMs) and LLM2Vec-Llama3-8B-Sup (for the variants based on pre-trained embedding models). The results are summarized in Table 1 for the Taste dataset. For ProtoSim with Llama3-8B, we can clearly see the effectiveness of the classification dataset, enabling an increase from 53.5% to 72.4%. Despite its small size, it successfully allow us to align the embedding space of the entities with the embedding space of the prototypes. Only fine-tuning on the ranking objective also helps, but it underperforms the classification approach. The *Class + rank-perc* approach overall performs best, outperforming *Classification* in four of the six dimensions. For ProtoSim with LLM2Vec-Llama3-8B-Sup, the findings are broadly similar, with *Class + rank-perc* again performing best. For the remainder of the experiments, we will therefore fix *Class + rank-perc* as the fine-tuning strategy for the ProtoSim experiments. When it comes to the pairwise approach, *Rank-full* outperforms *Rank-perc*. In the following, we will thus fix *Rank-full* as the fine-tuning strategy for the experiments with the pairwise approach.

**Comparing Models** Table 2 compares the performance of a number of different models, for each of the considered approaches. For this analysis, we still focus on the Taste dataset, and fix the fine-

| | Sweetness | Saltiness | Sourness | Bitterness | Umaminess | Fattiness | Average |
|---|---|---|---|---|---|---|---|
| ProtoSim (LLMs) | | | | | | | |
| Llama3-8B | **78.2** | **79.1** | 70.0 | 60.6 | **72.9** | 75.0 | **72.7** |
| Qwen3-8B | 75.9 | 71.5 | 63.2 | 62.6 | 61.5 | 72.7 | 67.9 |
| Qwen3-14B | 74.7 | 70.3 | 66.2 | 60.6 | 63.4 | 72.4 | 67.9 |
| Mistral-12B | 76.8 | 72.9 | 70.9 | 64.1 | 64.4 | 75.9 | 70.8 |
| Mistral-24B | 77.9 | 76.2 | 70.3 | 59.1 | 62.7 | 74.7 | 70.2 |
| OLMo2-7B | 75.0 | 68.2 | **75.6** | **65.9** | 67.4 | 76.5 | 71.4 |
| OLMo2-13B | 76.8 | 70.0 | 69.1 | 63.8 | 56.5 | 74.7 | 68.5 |
| Phi4-14B | 75.9 | 69.4 | 67.4 | 61.5 | 65.0 | **76.8** | 69.3 |
| ProtoSim (Fine-tuned embedding models) | | | | | | | |
| E5-Mistral-7B | 74.7 | **77.1** | 64.4 | 62.4 | 62.9 | **75.6** | 69.5 |
| LLM2Vec (Llama3) | **76.5** | 76.2 | **65.3** | 60.6 | 66.8 | 72.4 | **69.6** |
| LLM2Vec (Mistral) | 71.5 | 74.7 | 62.4 | **65.0** | **70.3** | 72.4 | 69.4 |
| ProtoSim (Pre-trained embedding models) | | | | | | | |
| E5-Mistral-7B | **68.5** | **63.5** | **64.4** | 51.5 | **61.8** | **65.0** | **62.5** |
| LLM2Vec (Llama3) | **68.5** | 45.9 | 52.4 | 42.9 | 55.0 | 38.5 | 50.5 |
| LLM2Vec (Mistral) | 65.3 | 54.4 | 58.8 | **64.4** | 51.2 | 50.6 | 57.5 |
| Pairwise approach | | | | | | | |
| Llama3-8B | **79.7** | 71.5 | 62.6 | 62.1 | 63.5 | 72.1 | 68.6 |
| Qwen3-8B | 78.5 | 71.5 | 63.8 | 58.5 | 65.0 | 72.4 | 68.3 |
| Qwen3-14B | **79.7** | 73.5 | 61.5 | 55.9 | 64.7 | **77.6** | 68.8 |
| Mistral-12B | 79.4 | 73.8 | **67.6** | 56.5 | 63.5 | 72.4 | 68.9 |
| Mistral-24B | 76.8 | **77.4** | 66.2 | **67.6** | 67.4 | 75.9 | **71.9** |
| OLMo2-7B | 74.1 | 64.1 | 60.0 | 57.9 | 62.4 | 69.4 | 64.7 |
| OLMo2-13B | 79.4 | 71.8 | 62.4 | 64.4 | 64.7 | 70.6 | 68.9 |
| Phi4-14B | 75.6 | 68.2 | 60.9 | 57.1 | **70.6** | 69.1 | 66.9 |
| Zero-shot LLMs | | | | | | | |
| GPT-4o | 73.5 | 73.2 | 68.5 | 56.8 | 65.6 | 74.4 | 68.7 |
| GPT-4.1 | **79.4** | **76.2** | **71.2** | **58.5** | **70.3** | **78.2** | **72.3** |

Table 2: Comparison of different models (accuracy % on pairwise comparisons). The best results within each block are highlighted in bold. ProtoSim results are obtained with `Class + rank-perc`, results for the pairwise model are for `Class + rank-full`.

tuning strategies as explained above. For ProtoSim, Llama3-8B achieves the best results for three dimensions, with OLMo2-7B for two dimensions and Phi4-14B for one dimension. Surprisingly, increasing model size does not seem to improve results. For instance, the performance of Qwen3-8B and Qwen3-14B is almost identical, Mistral-12B outperforms Mistral-24B, and OLMo2-7B outperforms OLMo2-13B (on average). ProtoSim can be used with LLMs and with pre-trained embedding models. We might expect that starting from a model such as LLM2Vec would have some advantages, as the model has already been pre-trained to generate embeddings. However, we found such models to underperform Llama3-8B. When using the pre-trained embedding models without any fine-

| | Lightness | Grain size | Roughness | Shininess | Organisation | Variability | Density | Familiarity | Intensity | Pleasantness | Irritability | Wonder | Transcendence | Tenderness | Nostalgia | Peacefulness | Energy | Joyful activation | Sadness | Tension | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Rocks** | | | | | | | **Odour** | | | | **Music** | | | | | | | | | |
| | PROTOSIM (LLMS) | | | | | | | | | | | | | | | | | | | | |
| Llama3-8B | 79.7 | 62.6 | 60.3 | 64.7 | 59.4 | **68.2** | 78.8 | 42.6 | **62.9** | 72.6 | 64.4 | 53.5 | 61.2 | 69.7 | 60.0 | 66.5 | 60.0 | 63.2 | 60.0 | **64.7** | 63.8 |
| Mistral-24B | 79.7 | 70.6 | 58.8 | 64.1 | 55.9 | 60.3 | 77.3 | **64.7** | 60.0 | **74.4** | **66.2** | 53.8 | 62.6 | 69.1 | 58.8 | 65.9 | 60.3 | 60.9 | **65.0** | 63.2 | 64.6 |
| | PAIRWISE APPROACH | | | | | | | | | | | | | | | | | | | | |
| Llama3-8B | 79.4 | 70.9 | 60.9 | 60.6 | 58.2 | 50.0 | 69.7 | 53.8 | 52.1 | 58.8 | 56.8 | **59.1** | 57.9 | 71.5 | 59.4 | 62.6 | 59.7 | 55.6 | 64.7 | 61.8 | 61.2 |
| Mistral-24B | 79.7 | **80.0** | 62.6 | **67.9** | 50.3 | 67.9 | 78.0 | 57.4 | 53.8 | 58.2 | 59.1 | 55.6 | 60.9 | 68.8 | 52.1 | 63.2 | 58.8 | 50.9 | 62.1 | 56.8 | 62.2 |
| | ZERO-SHOT LLMS | | | | | | | | | | | | | | | | | | | | |
| GPT-4o | 59.7 | 75.6 | 55.9 | 63.2 | **65.0** | 52.4 | 69.7 | 58.5 | 48.5 | 58.8 | 51.2 | 50.6 | **63.8** | 66.5 | 51.8 | 72.1 | 59.4 | 62.9 | 55.0 | 60.6 | 60.1 |
| GPT-4.1 | **80.6** | 77.6 | **68.5** | 67.1 | 56.2 | 61.5 | **84.8** | 58.5 | 53.2 | 72.1 | 62.4 | 54.7 | 61.8 | **75.3** | **62.9** | **75.0** | **65.6** | **63.8** | 60.6 | **64.7** | **66.3** |

Table 3: Comparison of different models (accuracy % on pairwise comparisons). The best overall results for each quality dimension are highlighted in bold. ProtoSim results are obtained with `Class + rank-perc`, results for the pairwise model are for `Class + rank-full`.

| | WD1 | WD2 | Movies | Books | Size | Mass | Height | Average |
|---|---|---|---|---|---|---|---|---|
| | **WD** | | **TG** | | **Phys** | | | |
| | PROTOSIM (LLMS) | | | | | | | |
| Llama3-8B | 65.6 | 68.6 | 71.1 | 61.6 | 75.3 | 58.4 | 78.3 | 68.4 |
| Mistral-24B | 66.0 | 71.6 | **72.5** | 58.0 | 66.9 | 53.6 | 65.7 | 64.9 |
| | PAIRWISE APPROACH | | | | | | | |
| Llama3-8B | 64.8 | 58.6 | 64.0 | 51.6 | 75.3 | 59.6 | 83.7 | 65.4 |
| Mistral-24B | 65.4 | 64.0 | 62.6 | 53.8 | 88.0 | 61.4 | 92.2 | 69.6 |
| | ZERO-SHOT LLMS | | | | | | | |
| GPT-4o | 68.0 | 79.2 | 67.4 | 61.1 | 92.2 | 50.0 | 85.5 | 71.9 |
| GPT-4.1 | **81.0** | **89.4** | 72.1 | **67.1** | **98.2** | **64.5** | **97.0** | **81.3** |

Table 4: Comparison of different models (accuracy % on pairwise comparisons). The best overall results for each quality dimension are highlighted in bold. ProtoSim results are obtained with `Class + rank-perc`, results for the pairwise model are for `Class + rank-full`.

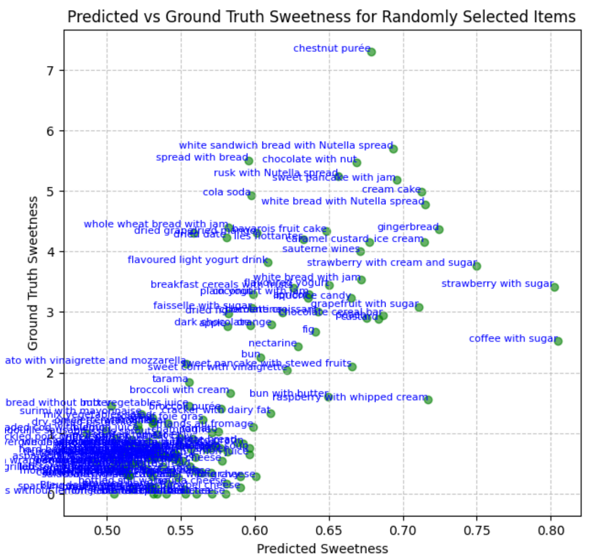


Figure 2: Scatter plot showing the predicted sweetness of a food item (X-axis) and the ground truth rating (Y-axis).

tuning, performance is substantially lower. In that case, we also see clear differences between E5 and the LLM2Vec models. However, after fine-tuning these differences disappear. When comparing ProtoSim and the pairwise approach, their relative performance depends on the LLM which is used. The best results overall are obtained by ProtoSim with Llama3-8B. ProtoSim is also better when Mistral-12B, OLMo2-7B or Phi4-14B is used. Conversely, the pairwise approach is better for the Qwen models, Mistral-24B and OLMo2-13B. Finally, we also report zero-shot results with GPT-4o and GPT-4.1 in the table. We found GPT-4.1 to consistently improve on GPT-4o, while performing slightly worse than ProtoSim with Llama3-8B on average.

**Evaluation on Different Domains** We now analyze the results on the other datasets. First, Table 3 shows the results for the remaining perceptual datasets. As before, the ProtoSim models are trained using *Class* + *rank-perc* and the pairwise models using *Class* + *rank-full*, based on our findings from Table 1. Based on the results from Table 2 we focus this analysis on Llama3-8B (as the best-performing model for ProtoSim and a representative smaller model) and Mistral-24B (as the best-performing model for the pairwise approach and a representative larger model). We find that ProtoSim outperforms the pairwise approach on average, although there is some variation between the three considered domains: the pairwise approach with Mistral-24B outperforms ProtoSim on Rocks; ProtoSim outperforms the pairwise approach on Odour, especially for Mistral-24B; and both approaches perform relatively similarly on Music, with ProtoSim being slightly better on average. We find that Mistral-24B outperforms Llama3-8B even for ProtoSim (especially for Odour), in contrast to our earlier findings on Taste. GPT-4.1 achieves the best results on Rocks and Music, but underperforms ProtoSim with Mistral-24B on Odour.

Table 4 summarizes the results for the non-perceptual datasets. ProtoSim outperforms the pairwise approach on the Tag Genome dataset and, to a lesser extent, on Wikidata, but the pairwise approach performs better on Physical Properties. GPT-4.1 substantially outperforms the other methods on Wikidata and Physical properties, which are the two datasets that involve factual numerical attributes. For Tag Genome, which involves subjective labels, the performance of GPT-4.1 is more in line with ProtoSim and the pairwise approach.

### 5.3 Analysis

**Predicting Degrees of Sweetness** For the main experiments, we have only focused on ranking. However, in contrast to the pairwise approach, ProtoSim associates a numerical score *emb*$(\gamma(e)) \cdot$ *emb*$(\gamma(f))$ with every entity $e$ and feature $f$, which we can interpret as the coordinate of a conceptual space dimension. As such, we can also use this method for predicting the *degree* to which an entity has some feature. We analyze this for the particular example of *sweetness* from the Food dataset. Figure 2 compares the predicted sweetness score with the ground truth sweetness values (which were obtained as the average sweetness rating that was assigned by all annotators). For this analysis, we have used the ProtoSim model with Llama3-8B (trained using *class* + *rank-perc*). The figure shows a random sample of 150 food items. The figure shows a clear correlation between the predicted and ground truth scores (Pearson correlation for the full set of 590 food items: 0.752). In the bottom-left corner of the plot, we can see a large set of items which are considered to be clearly non-sweet, both by the human annotators and by the model. The items that are rated to be sweetest by the human annotators are all predicted to be sweet by the model as well (with *chestnut purée* as an outlier). However, food items with intermediate levels of sweetness can be more challenging. For instance, *coffee with sugar* is far less sweet than predicted by the model, while *cola soda* and *whole wheat bread with jam* are sweeter than predicted.

**Qualitative Analysis** To better understand which kinds of features can be modelled using ProtoSim, we carried out a qualitative analysis using a question-answering dataset about recipes (Zhang et al., 2023). Each question specifies a preference for a particular type of food (e.g. *a quick breakfast for a rushed school morning*), and the task is to select the most appropriate option among 5 listed alternatives. We select the option whose embedding is most similar to the stated preference. We found that the model was generally ably to handle a variety of commonsense properties (e.g. *a toddler-friendly fried snack for a birthday party*). However, we also noticed three key limitations: difficulties with negative preferences, being overly sensitive to lexical overlap, and sometimes focusing too much on one aspect of the query. A detailed analysis can be found in Appendix D.

## 6 Conclusions

We have shown that LLM embeddings can serve as conceptual space representations of perceptual features. While previous work had already shown the potential of LLMs for modelling perceptual features, this was based on pairwise comparison prompts, which are not practical when representations for large numbers of entities are needed. To model a given quality dimension (e.g. sweetness) we obtain an LLM embedding of a corresponding prototype description (e.g. "a sweet food"). The main idea is that we can then simply compare this embedding with the embeddings of the entities of interest. However, we found this to perform poorly with pre-trained LLMs (including LLM-based embedding models), due to the fact that the embeddings of the prototype descriptions and the entities are not aligned. To address this, we align the embeddings by fine-tuning the LLM on a small synthetically generated dataset. After this alignment step, we found the proposed strategy to be highly effective, matching and often even surpassing the performance of the pairwise approach.

## Limitations

The problem of aligning vector spaces has been extensively studied within the context of cross-lingual word embeddings (Mikolov et al., 2013; Xing et al., 2015; Artetxe et al., 2020). Such methods essentially learn a linear transformation to align two monolingual vector spaces. It is possible that a similar approach might be affective for aligning prototype and entity embedding spaces we well, which would mean that the fine-tuning step could be avoided. Apart from being more efficient (e.g. in terms of storing model parameters), this might also help to prevent any catastrophic forgetting. Indeed, in preliminary experiments, we found that increasing the size of the classification fine-tuning dataset led to reduced performance, but a further investigation of this strategy is left for future work.

In our experiments, we have focused on ranking, rather than measuring the degree to which features are satisfied. We illustrated the potential of our model to predict degrees of sweetness, but a formal evaluation was left for future work. More generally, conceptual spaces are commonly used for evaluating similarity. For instance, we expect that a learned conceptual space of taste, composed of the six considered taste dimensions, would allow us to estimate human similarity judgments more reliably than is possible with the original LLM embeddings. Note that the problem of estimating similarity judgments can also be related to the problem of estimating causal inner products in LLM embeddings spaces (Park et al., 2024).

## References


Marah I Abdin, Jyoti Aneja, Harkirat S. Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *CoRR*, abs/2412.08905.

Janet Aisbett and Greg Gibbon. 2001. A general formulation of conceptual spaces as a meso level representation. *Artif. Intell.*, 133(1-2):189–232.

Maria Antoniak and David Mimno. 2021. Bad seeds: Evaluating lexical methods for bias measurement. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1889–1904, Online. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Hadi Banaee, Erik Schaffernicht, and Amy Loutfi. 2018. Data-driven conceptual spaces: Creating semantic representations for linguistic descriptions of numerical data. *J. Artif. Intell. Res.*, 63:691–742.

Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *CoRR*, abs/2404.05961.

Or Raphael Bidusa and Shaul Markovitch. 2025. Concept layers: Enhancing interpretability and intervenability via LLM conceptualization. *CoRR*, abs/2502.13632.

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781, Online. Association for Computational Linguistics.

Usashi Chatterjee, Amit Gajbhiye, and Steven Schockaert. 2023. Cabbage sweeter than cake? analysing the potential of large language models for learning conceptual spaces. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11836–11842, Singapore. Association for Computational Linguistics.

Joaquín Derrac and Steven Schockaert. 2015. Inducing semantic relations from conceptual spaces: A data-driven approach to plausible reasoning. *Artif. Intell.*, 228:66–94.

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

Igor Douven. 2023. The role of naturalness in concept learning: A computational study. *Minds Mach.*, 33(4):695–714.

Igor Douven, Sylvia Wenmackers, Yasmina Jraissati, and Lieven Decock. 2017. Measuring graded membership: The case of color. *Cogn. Sci.*, 41(3):686–722.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela

Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Katrin Erk and Marianna Apidianaki. 2024. Adjusting interpretable dimensions in embedding space with human judgments. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2675–2686, Mexico City, Mexico. Association for Computational Linguistics.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Jamie Forth, Geraint A. Wiggins, and Alex McLean. 2010. Unifying conceptual spaces: Concept formation in musical creative systems. *Minds Mach.*, 20(4):503–532.

Amit Gajbhiye, Luis Espinosa-Anke, and Steven Schockaert. 2022. Modelling commonsense properties using pre-trained bi-encoders. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3971–3983, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Peter Gärdenfors. 2000. *Conceptual Spaces - the Geometry of Thought*. MIT Press.

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, Online. Association for Computational Linguistics.

Gabriel Grand, Idan Asher Blank, Francisco Pereira, and Evelina Fedorenko. 2022. Semantic projection recovers rich human knowledge of multiple object features from word embeddings. *Nature Human Behaviour*, 6(7):975–987.

Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. 2015. Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21, Lisbon, Portugal. Association for Computational Linguistics.

Ting Jiang, Shaohan Huang, Zhongzhi Luan, Deqing Wang, and Fuzhen Zhuang. 2024. Scaling sentence embeddings with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3182–3196, Miami, Florida, USA. Association for Computational Linguistics.

Yasmina Jraissati and Ophelia Deroy. 2021. Categorizing smells: A localist approach. *Cogn. Sci.*, 45(1).

Denis Kotkov, Alan Medlar, Alexandr V. Maslov, Umesh Raj Satyal, Mats Neovius, and Dorota Glowacka. 2022. The tag genome dataset for books. In *CHIIR '22: ACM SIGIR Conference on Human Information Interaction and Retrieval, Regensburg, Germany, March 14 - 18, 2022*, pages 353–357. ACM.

Nitesh Kumar, Usashi Chatterjee, and Steven Schockaert. 2024. Ranking entities along conceptual space dimensions with LLMs: An analysis of fine-tuning strategies. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7974–7989, Bangkok, Thailand. Association for Computational Linguistics.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *CoRR*, abs/2405.17428.

Lei Li, Jingjing Xu, Qingxiu Dong, Ce Zheng, Xu Sun, Lingpeng Kong, and Qi Liu. 2023a. Can language models understand physical concepts? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11843–11861, Singapore. Association for Computational Linguistics.

Na Li, Hanane Kteich, Zied Bouraoui, and Steven Schockaert. 2023b. Distilling semantic concept embeddings from contrastively fine-tuned language models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, pages 216–226. ACM.

Fangyu Liu, Julian Eisenschlos, Jeremy Cole, and Nigel Collier. 2022a. Do ever larger octopi still amplify reporting biases? evidence from judgments of typical colour. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 210–220, Online only. Association for Computational Linguistics.

Fangyu Liu, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2021a. Fast, effective, and self-supervised: Transforming masked language models into universal lexical and sentence encoders. In *Proceedings of the*

*2021 Conference on Empirical Methods in Natural Language Processing*, pages 1442–1459, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qianchu Liu, Fangyu Liu, Nigel Collier, Anna Korhonen, and Ivan Vulić. 2021b. MirrorWiC: On eliciting word-in-context representations from pretrained language models. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 562–574, Online. Association for Computational Linguistics.

Xiao Liu, Da Yin, Yansong Feng, and Dongyan Zhao. 2022b. Things not written in text: Exploring spatial commonsense from visual signals. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2365–2376, Dublin, Ireland. Association for Computational Linguistics.

Raja Marjieh, Ilia Sucholutsky, Pol van Rijn, Nori Jacoby, and Thomas L Griffiths. 2024. Large language models predict human sensory judgments across six modalities. *Scientific Reports*, 14(1):21445.

Christophe Martin, Michel Visalli, Christine Lange, Pascal Schlich, and Sylvie Issanchou. 2014. Creation of a food taste database using an in-home "taste" profile method. *Food Quality and Preference*, 36:70–80.

Stephen McGregor, Kat Agres, Matthew Purver, and Geraint A. Wiggins. 2015. From distributional semantics to conceptual spaces: A novel computational method for concept creation. *J. Artif. Gen. Intell.*, 6(1):55–86.

Tomás Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

Andrew G Moss, Christopher Miles, Jane V Elsley, and Andrew J Johnson. 2016. Odorant normative data for use in olfactory memory experiments: Dimension selection and analysis of individual differences. *Frontiers in Psychology*, 7:1267.

Matthieu Moullec and Igor Douven. 2025. Cheaper spaces. *Minds Mach.*, 35(1):6.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Robert M Nosofsky, Craig A Sanders, Brian J Meagher, and Bruce J Douglas. 2018. Toward the development of a feature-space representation for a complex natural category domain. *Behavior Research Methods*, 50:530–556.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2025. 2 olmo 2 furious. *CoRR*, abs/2501.00656.

Matías Osta-Vélez and Peter Gärdenfors. 2022. Non-monotonic reasoning, expectations orderings, and conceptual spaces. *J. Log. Lang. Inf.*, 31(1):77–97.

Matías Osta-Vélez and Peter Gärdenfors. 2024. Analogy as a search procedure: a dimensional view. *J. Exp. Theor. Artif. Intell.*, 36(7):1135–1154.

Cory Paik, Stéphane Aroca-Ouellette, Alessandro Roncone, and Katharina Kann. 2021. The World of an Octopus: How Reporting Bias Influences a Language Model's Perception of Color. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 823–835, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Carita Paradis. 2015. *Conceptual Spaces at Work in Sensory Cognition: Domains, Dimensions and Distances*, pages 33–55. Springer International Publishing, Cham.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Roma Patel and Ellie Pavlick. 2022. Mapping language models to grounded conceptual spaces. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

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

Alex Rosenfeld and Katrin Erk. 2023. An analysis of property inference methods. *Nat. Lang. Eng.*, 29(2):201–227.

Trevor Standley, Ozan Sener, Dawn Chen, and Silvio Savarese. 2017. image2mass: Estimating the mass of an object from its image. In *1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017, Proceedings*, volume 78 of *Proceedings of Machine Learning Research*, pages 324–333. PMLR.

Hannah Strauss, Julia Vigl, Peer-Ole Jacobsen, Martin Bayer, Francesca Talamini, Wolfgang Vigl, Eva Zangerle, and Marcel Zentner. 2024. The emotion-to-music mapping atlas (emma): A systematically organized online database of emotionally evocative music excerpts. *Behavior Research Methods*, 56(4):3560–3577.

Jesse Vig, Shilad Sen, and John Riedl. 2012. The tag genome: Encoding community knowledge to support novel interaction. *ACM Trans. Interact. Intell. Syst.*, 2(3):13:1–13:44.

Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. 2021. LexFit: Lexical fine-tuning of pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5269–5283, Online. Association for Computational Linguistics.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11897–11916, Bangkok, Thailand. Association for Computational Linguistics.

Yueqi Wang, Yoni Halpern, Shuo Chang, Jingchen Feng, Elaine Ya Le, Longfei Li, Xujian Liang, Min-Cheng Huang, Shane Li, Alex Beutel, and 1 others. 2023. Learning from negative user feedback and measuring responsiveness for sequential recommenders. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1049–1053.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1006–1011, Denver, Colorado. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Marcel Zentner, Didier Grandjean, and Klaus R Scherer. 2008. Emotions evoked by the sound of music: characterization, classification, and measurement. *Emotion*, 8(4):494.

Haochen Zhang, Anton Korikov, Parsa Farinneya, Mohammad Mahdi Abdollah Pour, Manasa Bharadwaj, Ali Pesaranghader, Xi Yu Huang, Yi Xin Lok, Zhaoqi Wang, Nathan Jones, and 1 others. 2023. Recipe-mpr: A test collection for evaluating multi-aspect preference-based natural language retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2744–2753.

Shu Zhong, Elia Gatti, Youngjun Cho, and Marianna Obrist. 2024a. Exploring human-ai perception alignment in sensory experiences: Do LLMs understand textile hand? *CoRR*, abs/2406.06587.

Shu Zhong, Zetao Zhou, Christopher Dawes, Giada Brianza, and Marianna Obrist. 2024b. Sniff AI: is my 'spicy' your 'spicy'? exploring LLM's perceptual alignment with human smell experiences. *CoRR*, abs/2411.06950.

## A Experimental Details

**Models** Table 5 provides the details of the models that were used in our experiments. Experiments with GPT-4o and GPT-4.1 were carried out using the OpenAI API[3]. We used versions `gpt-4o-2024-11-20` and `gpt-4.1-2025-04-14` respectively.

**Fine-tuning Methodology** To fine-tune the base models, we used the QLoRa method, which allows converting the floating-point 32 format to smaller data types. In particular, for all the models, we used 4-bit quantization for efficient training. In the QLoRa configuration, $r$ (the rank of the low-rank matrix used in the adapters) was set to 32, $\alpha$ (the scaling factor for the learned weights) was set to 64, and dropout was set to 0.05. We applied QLoRa to all the linear layers of the models, including *q_proj*, *k_proj*, *v_proj*, *o_proj*, *gate_proj*, *up_proj*, *down_proj*, and *lm_head*. In all our experiments, the temperature parameter $T$ was set to $0.25$ for the classification loss, the scaling factor $\alpha$ was set to 10, and $\lambda$ was set to $0.25$.

**Computing Infrastructure** The fine-tuning experiments were conducted on a workstation equipped with an NVIDIA GeForce RTX 4090 GPU with 24GB of VRAM.

**Prompts** For the few-shot configuration in Table 1, we used the following prompt with three in-context demonstrations:

```
The task is to answer questions that involve
    comparing perceptual features of two
    entities. Please answer with Yes or No only.
     In the worst case, if you do not know the
    answer then choose randomly between Yes and
    No.
```

[3] https://platform.openai.com

| Model Name | Hugging Face URL | License |
|---|---|---|
| Llama3-8B | meta-llama/Meta-Llama-3-8B | Llama 3 |
| Qwen3-8B | Qwen/Qwen3-8B | Apache 2.0 |
| Qwen3-14B | Qwen/Qwen3-14B | Apache 2.0 |
| Mistral-Nemo-12B | mistralai/Mistral-Nemo-Base-2407 | Apache 2.0 |
| Mistral-Small-24B | mistralai/Mistral-Small-24B-Base-2501 | Apache 2.0 |
| OLMo2-7B | allenai/OLMo-2-1124-7B | Apache 2.0 |
| OLMo2-13B | allenai/OLMo-2-1124-13B | Apache 2.0 |
| Phi4-14B | microsoft/phi-4 | MIT |
| E5-Mistral-7B | intfloat/e5-mistral-7b-instruct | MIT |
| LLM2Vec-Llama3-8B | McGill-NLP/LLM2Vec-Meta-Llama-3-8B-Instruct-mntp | MIT |
| LLM2Vec-Llama3-8B-Sup | McGill-NLP/LLM2Vec-Meta-Llama-3-8B-Instruct-mntp-supervised | MIT |
| LLM2Vec-Mistral-7B | McGill-NLP/LLM2Vec-Mistral-7B-Instruct-v2-mntp | MIT |

Table 5: Details of the models used in the experiments.

```
This question is about two surfaces: Is mirror
    more reflective than still water surface?
Yes
This question is about two materials: Is silk
    fabric more lustrous than polished metal?
No
This question is about two sounds: Is operatic
    aria more melodious than car alarm?
Yes
```

We used the following prompt for the experiments with GPT-4o and GPT-4.1:

```
Answer the following with Yes or No only. In the
     worst case, if you don't know the answer
    then choose randomly between Yes and No.
```

## B Fine-tuning Dataset

The fine-tuning dataset for classification was synthetically generated using GPT-4o. We provided a few manually created examples and asked GPT-4o to generate additional similar datapoints. Each datapoint was manually checked, and GPT-4o was also prompted to re-examine the datapoints it generated as part of the quality assurance process. Multiple prompts were used interactively to guide the model in generating datapoints that cover diverse domains. In total, 517 datapoints were generated; however, we randomly selected 123 datapoints to be used for fine-tuning, as the model was overfitting to this dataset when the full set of 517 data points were used. Table 6 shows some examples of data points from the dataset.

## C Evaluation Datasets

For Taste, Rocks, Tag Genome, Physical Properties and Wikidata, we use the preprocessed datasets from Kumar et al. (2024), which are available from https://github.com/niteshroyal/RankingUsingLLMs. For the Odour and Music datasets, we obtained the datasets from the original publications. In particular, the Odour dataset is available as supplemental data at https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2016.01267/full. The Music dataset is available from https://osf.io/7ptmd/.

For the Taste, Rocks and Physical properties datasets, we could not find any information about licensing. The Tag Genome dataset was released under CC BY-NC 3.0. Wikidata is available under a CC0 license. The Odour dataset was released under a CC BY 4.0 license.

**Query**: *a quick breakfast for a rushed school morning.*
**Options**:

1. **Any cereal with milk**
2. Eggs benedict - poached eggs, prosciutto on top of English muffins topped with a creamy Hollandaise sauce
3. Instant ramen with eggs, spinach and pickled cabbage
4. Breakfast pizza with sausage, cheddar, sour cream and jalapenos
5. Classic salted french fries made of only potatoes

Figure 3: Example question from the recipe dataset.

## D Qualitative Analysis

The dataset from Zhang et al. (2023) contains 500 multiple-choice questions, each with 5 alternatives. To evaluate our models, we first converted each question to a descriptive phrase (expressing the same preference as the original question) using GPT-4o. Figure 3 shows a problem instance from the resulting dataset.

We first evaluated a number of LLMs on the original question answering benchmark, using a zero-shot prompt, achieving 91.4% accuracy with GPT-4o and 89.4% with Llama3-8B. This shows that, while many of the instances appear challeng-

| Target Property | Examples | Negative Properties |
|---|---|---|
| long river | Nile River, Amazon River, Yangtze River, Yenisei River, Yellow River, Ob-Irtysh River, Congo River | short river, polluted river, dry river, small city |
| influential artist | Pablo Picasso, Leonardo da Vinci, Vincent van Gogh, Claude Monet, Michelangelo, Rembrandt, Andy Warhol | unknown artist, amateur artist, unpopular artist, dry river |
| loyal dog | German Shepherd, Labrador Retriever, Golden Retriever, Collie, Boxer, Beagle, Bulldog | independent dog, aloof dog, aggressive dog, small city |
| energy efficient appliance | LED Light Bulbs, Smart Thermostats, Energy Star Refrigerators, Dual Flush Toilets, Solar Panels, High-Efficiency Washers, Electric Vehicles | high-energy consumption appliance, inefficient lighting, old model refrigerators, mild spice |
| water sport | Swimming, Water Polo, Diving, Synchronized Swimming, Rowing, Canoeing, Surfing | land sport, winter sport, individual sport, dry desert |
| transparent material | Glass, Acrylic, Polycarbonate, Quartz Crystal, Diamond, Clear Resin, Sapphire Crystal | opaque material, metallic material, porous material, poisonous flower |
| rail transportation | Train, Tram, Monorail, Subway, High-speed Rail, Funicular, Light Rail | air transport, road transport, water transport, ancient language |
| international law | Geneva Conventions, United Nations Charter, Hague Convention, UNCLOS, Treaty of Rome, Kyoto Protocol, Vienna Convention | domestic law, criminal law, civil law, ballroom dance |
| domesticated animal | Dog, Cat, Horse, Cow, Sheep, Goat, Chicken | wild animal, exotic animal, marine animal, modern software architecture |
| metaphysics philosophical branch | Ontology, Cosmology, Theology, Epistemology, Phenomenology, Existentialism, Dualism | logic, ethics, aesthetics, binary mathematical operation |
| acidic chemical compound | Hydrochloric Acid, Sulfuric Acid, Acetic Acid, Citric Acid, Nitric Acid, Phosphoric Acid, Carbonic Acid | basic compound, neutral compound, alkaline compound, military alliance |
| phonological linguistic phenomenon | Assimilation, Elision, Lenition, Vowel Harmony, Consonant Mutation, Metathesis, Assimilation | syntactic phenomenon, semantic phenomenon, morphological feature, freshwater ecosystem |

Table 6: Examples from the fine-tuning dataset that was collected using GPT-4o.

ing, LLMs are generally capable of identifying the correct option. We then tested our Llama3-8B ProtoSim model (fine-tuned without the taste dataset), as follows. We used the descriptive version of the query as the verbalization of the property. The five options are treated as the verbalization of entities. We then simply predict the option whose embedding is closest to the embedding of the query. The accuracy of this approach was 67.6%.

Analyzing the results, we noticed that the model generally performs well on commonsense properties. For instance, the following queries were all answered correctly: (i) a quick breakfast for a rushed school morning, (ii) a toddler-friendly fried snack for a birthday party, (iii) diabetes-friendly cookies. However, Tables 7, 8 and 9 illustrate three types of common errors that are made by the model (ProtoSim with Llama3-8B).

Table 7 shows examples where the model focuses too much on one particular aspect of the specification. In the first example, the words *post-cardio* and *muscle* lead the model to select the *protein smoothie* option, despite the fact that the description was asking for a *snack*. Similarly, in the second example, the word *antioxidants* leads to the model to the vitamin-rich smoothie, ignoring the fact that the query was asking for a *salad*.

In Table 8, it is evident that the model is distracted by the lexical overlap between the query and some of the options. In the first example, the model selects an option that mentions *brown rice*, which also occurs in the query, despite the fact that the chosen option is not a dessert. Similarly, due to significant lexical overlap with the final option, the model fails to acknowledge the term *green* in the second example. In the final example, the model chose *Low fat crab chowder made with imitation crabmeat and different vegetables* over the correct option *Lighter clam chowder with bacon and vegetables, made with milk instead of cream* due to the presence of the words *low fat* and *chowder*, which also occur in the query.

Table 9 illustrates how the model struggles to handle negative requirements, such as *without cranberry sauce*, *non-greasy* or *lactose-free*. Such negative requirements can be critically important for recommendation systems (Wang et al., 2023), but they are challenging to capture with embeddings.

Table 7: Error analysis of the ProtoSim model. The table shows examples where the model focuses too much on one particular aspect of the query. The incorrect option chosen by the model is highlighted in red.

| **Recipe Query** | **ProtoSim response** |
|---|---|
| post-cardio snacks<br>for lean muscle maintenance | 1.Fruit salad with peaches, blackberries, strawberries and lime<br>2.Strawberry and banana protein smoothie<br>3.Classic chicken tenders - deep fried boneless chicken strips<br>4.Fragrant pilaf made from quinoa<br>5.Stir fried Japanese Shirataki noodles (low calorie noodles) |
| a salad rich with antioxidants. | 1.Potato salad with extra virgin olive oil dressing<br>2.Vitamin-rich soup made with vegetables<br>3.Vitamin-rich smoothies made with cranberries, carrot, mango, strawberries, and cantaloupe<br>4.Easy chicken legs made with Italian salad dressing<br>5.Caesar salad dressing recipe made from scratch using raw cashews |

Table 8: Error analysis of the ProtoSim model. The table shows examples where the model relies too much on lexical overlap. The incorrect option chosen by the model is highlighted in red.

| **Recipe Query** | **ProtoSim response** |
|---|---|
| a dessert made with brown rice | 1. Blueberry crisp containing blueberries, brown rice, rice bran, and walnuts<br>2. Long-grain white rice dish with onions<br>3.Jasmine rice cooked with coconut milk<br>4.Brown rice and mushrooms cooked with vegetable stock, olive oil, and rice vinegar<br>5.Dessert treat made with butter, mini marshmallows, and Rice Krispie cereal |
| a post-workout green smoothie | 1.Garden veggie smoothie containing tomatoes, celery, parsley, and spinach<br>2.Green chili made with bell peppers, beef stew meat, and chili peppers<br>3.Pineapple smoothie containing buttermilk<br>4.Frittata containing onions, zucchini, squash, red peppers, broccoli, and cauliflower<br>5.Berry post workout smoothie containing fresh raspberries strawberries, blueberries, and bananas |
| a low-fat clam chowder recipe | 1.Lighter clam chowder with bacon and vegetables, made with milk instead of cream<br>2.Low fat crab chowder made with imitation crabmeat and different vegetables<br>3.Creamy linguine noodles with clams and onions<br>4.Clam chowder made with half-and-half cream<br>5.Clam chowder made with heavy whipping cream |

Table 9: Error analysis of the ProtoSim model. The table shows examples where the model fails to interpret negative requirements. The incorrect option chosen by the model is highlighted in red.

| Recipe Query | ProtoSim response |
|---|---|
| grandma's thanksgiving dinner without cranberry sauce | 1. Roast turkey with plum sauce<br>2. Roast turkey with sweet cranberry sauce<br>3.Baked chicken drumsticks in tomato sauce<br>4.Chinese style crispy roast duck with hoisin sauce<br>5.Classic seasoned roast beef with red pepper flakes |
| solid, non-greasy food for a severe hangover | 1.Toast with seasonings<br>2.Pizza margherita - basic pizza with tomato sauce and mozzarella cheese<br>3.Hot dogs with hot pepper sauce and green chillies<br>4.Chickpea and mexican chilli soup<br>5.Miso based Shijimi clam broth for hangover prevention |
| a quick, lactose-free breakfast recipe | 1.Boiled oats made with water<br>2.Oats boiled in milk<br>3.Microwaved oatmeal in milk<br>4.Milk boiled oats with cheese and syrup<br>5.Enchiladas containing breakfast sausage, cheddar cheese, and a variety of vegetables |