Aligned at the Start: Conceptual Groupings in LLM Embeddings

Anonymous ACL submission

Abstract

This paper shifts focus to the often-overlooked input embeddings – the initial representations fed into transformer blocks. Using fuzzy graph, k-nearest neighbor (k-NN), and community detection, we analyze embeddings from diverse LLMs, finding significant categorical community structure aligned with predefined concepts and categories aligned with humans. We observe these groupings exhibit within-cluster organization (such as hierarchies, topological ordering, etc.), hypothesizing a fundamental structure that precedes contextual processing. To further investigate the conceptual nature of these groupings, we explore cross-model alignments across different LLM categories within their input embeddings, observing a medium to high degree of alignment. Furthermore, provide evidence that manipulating these groupings can play a functional role in mitigating ethnicity bias in LLM tasks.

1 Introduction

Large Language models (LLMs) are rapidly approaching human-level language abilities. However, these advancements raise important concerns - despite their immense potential, language models are susceptible to unexpected and potentially harmful behaviors such as hallucination (Ji et al., 2023), stereotyping (Liang et al., 2021), misinformation (Pan et al., 2023), and leakage of sensitive training data (Lukas et al., 2023). Moreover, the extent of their capabilities and the nature of their understanding is not yet known, thus capturing the attention of diverse groups, from policymakers to academics (Gu et al., 2023; Peng et al., 2022; Bender et al., 2021; Gurnee and Tegmark, 2024). While these issues may seem orthogonal, they stem from a common concern - the "black box" nature of large language models. The limited ability to understand how they reach their outputs raises doubts about their true capabilities, potential biases, and unintended consequences.

A crucial step towards addressing these challenges lies in understanding how LLMs internally represent and manipulate semantic concepts, the building blocks of human language and thought. Unraveling these internal representations is essential for not only improving LLM performance and robustness, but also for building trustworthy machine learning systems.

In the realm of contextual representations, some studies (Patel and Pavlick, 2021; Gurnee and Tegmark, 2024; Abdou et al., 2021) have investigated the ability of transformer-based models to learn representations of color, spatial, and temporal information. These studies often rely on analyzing contextual embeddings, which are intermediate outputs of LLMs. Moreover, contextual embeddings are inherently tied to the specific input context, making them highly variable, less generalizable, and less controllable/manipulatable.

In our work, we explore this critical area by investigating the emergence of conceptual structures within the **input embedding space** of LLMs. We aim to explore if these conceptual formations form independent of the context, and if such groupings are in alignment across LLMs, if these clusters exhibit internal organization, and if these patterns have functional implications for model behavior. Specifically, we examine:

- RQ1: Whether semantically related words and phrases are grouped together, forming identifiable conceptual clusters that are aligned with external world concepts and categories.
- RQ2: Do these groupings exhibit intra-cluster organization such as hierarchies, topological ordering, etc., thereby suggesting the formation of structured concepts.
- RQ3: Is there an inter-model alignment in semantic organization, across diverse transformer-based LLMs (Albert (Lan et al., 2019), T5 (Raffel et al., 2020), Llama3 (met, 2024)) irrespective of architecture, size, or pretraining data.

- RQ4: Whether these groupings play a functional role in LLMs. We test this by a case study that tries to mitigate ethnicity bias through cluster modification.

To uncover conceptual structures within the input embedding space of LLMs, we employ fuzzy graph construction (McInnes et al., 2020). Then, the fuzzy graph is analyzed using a community detection algorithm (Blondel et al., 2008) to reveal conceptual groupings and their categorical organization. We use Louvain community detection, which is effective at revealing hierarchical community structures (Blondel et al., 2008) in conjunction with multiple k choices (for k-nearest neighbors) in our approach. This systemic approach allows us to investigate both the existence and the hierarchical organization of conceptual clusters, directly addressing our research questions. Quantitative evaluation on external datasets (named entities (Remy, 2021; Gada, 2018), numerical tokens, and social structures) demonstrates that token embeddings exhibit significant categorical community structure aligned with real-world concepts.

The structure of this paper is as follows: We first establish necessary background on embeddings, semantic representations, and evaluation strategies (Section 2), followed by a description of our methodological approach (Section 3). We then present our core findings, starting with LLM-human alignment (Section 4), with a focus on within-cluster properties and hierarchical structure. Then, we explore LLM-LLM alignment within their input embedding space (Section 5). Section 6 demonstrates the practical implications of our work through embedding engineering and bias mitigation. Finally, we conclude in Section 7.

2 Preliminaries

2.1 Static, Contextual and Base Embeddings

In this section, we clarify the distinctions between static, contextual, and base embeddings, which are crucial for understanding modern language models.

- **Static embeddings** (e.g., Word2Vec(Mikolov et al., 2013), GloVe (Pennington et al., 2014)) are context-independent vector representations of words, meaning each word has a fixed embedding regardless of its surrounding text. This limits their ability to handle polysemy (words with multiple meanings). These embeddings are typically pre-trained on large corpora and can be used in various downstream tasks. Critically, for the context of this work, static embeddings are product of legacy LM models and not used as the input representations of modern transformer-based LLMs. They are not inputs to transformer blocks and thus have limited significance when it comes to applications such as mitigation techniques (e.g. embedding engineering) in LLMs.
- **Contextual embeddings** These are dynamic, context-dependent vectors. The embedding of a token is a function of its surrounding text, enabling the representation of nuanced meaning and resolving polysemy (e.g., Bert (Devlin et al., 2018), Albert (Lan et al., 2019), GPT variants (Raffel et al., 2020; met, 2024)). Different model layers capture different levels of contextual conditioning. However, this context-dependence limits its direct interpretability and generalizability of individual token embeddings outside of specific contexts.
- **Base Embeddings**: The process of generating contextual embeddings starts with base embeddings, which provide the initial vector representation for each input token. These differ from static embeddings such as GloVe and Word2Vec from the following perspectives. (1) Generation: Base embeddings are learned parameters within the LLM, as compared to separately trained static embeddings. (2) Usage: They are the direct input to the transformer blocks, forming the basis upon which contextualized representations are built through the model's subsequent layers. These embeddings are the focus of our study.

2.2 Previous Works on Embedding Interpretability

Previous research on interpretability in LLMs has primarily focused on analyzing either contextual embeddings (for modern LLMs) or static embeddings (for legacy language models which are not directly applicable to LLMs due to the architectural difference).

In the realm of contextual representations, initial research focused on the learning dynamics of linguistic features within LLMs (Tenney et al., 2018; Liu et al., 2019), the scope has expanded to explore how these models acquire and represent knowledge about the world. Some studies (Patel and Pavlick, 2021; Gurnee and Tegmark, 2024; Abdou et al., 2021) have investigated the ability of transformer-based models to learn representations of color, spatial, and temporal information. These studies often rely on analyzing contextual embeddings, which

are intermediate outputs of LLMs. Moreover, contextual embeddings are inherently tied to the specific input context, making them highly variable and less generalizable.

The closest work to our knowledge that focuses on context agnostic embeddings within the modern LLMs is (Bommasani et al., 2020) where the authors propose a method to create context agnostic word embeddings from contextualized word representations using (sub)-word pooling as well as context combination techniques, and tested on semantic similarity datasets. Furthermore, (Li et al., 2021) proposes a method for creating context-agnostic word representations by averaging the contextual embeddings derived from BERT, given a set of inferences on a masked target token within a corpus. In this methodology, the context is seen as a form of Gaussian noise that can be averaged out and hence produces a context-agnostic semantic representation. They observed such embedding represent richer semantic information than static word embedding counterparts (Word2Vec and GloVe) in intrinsic evaluation tasks. However, these works lack rigorous and extensive analysis of base embeddings of the LLMs to explores the intrinsic semantic organization within input embeddings of LLMs, possbile conceptual groupings and their alignments.

3 Concept Extraction

To investigate RQ1 and RQ2, we study the Human-LLM alignment of the input representations. To this end, we develop a method to first extract possible formed concepts within that space (refer to appendix A.1 for the discussion of conceptual groupings and semantic memory), and then to evaluate them against external datasets.

Our methodology consists of building the semantic graph, then using community detection to extract possible conceptual groupings.[1] Note that differentiable embedding functions guarantee a smooth, optimizable embedding space, but do not ensure a uniform distribution of concept instances. This unevenness, arising from factors like varying concept complexity and instance frequency,

[1]Note that we favor community detection over traditional clustering for identifying the conceptual clusters due to (1) its ability to align with the network-like structure of semantic representations, (2) its independence from the need for a predetermined number of clusters, and (3) its effectiveness in managing high-dimensional data by transforming it into a graph.

Algorithm 1: Concept Extraction

Data: All tokens in the input embeddings.
Result: A set of hierarchical communities.

```
Create a community list. The initial
  community is the entire space;
for k=[different neighbor sizes] do
    for all communities do
        - Generate knn graph from the input
          embedding weights;
        - Compute the edge weights of the graph
          using fuzzy simplex;
        - Apply Louvain community detection;
        - Add the identified communities to the list;
```

implies that while conceptual groupings may be locally uniform, the overall embedding space can be unevenly distributed. Therefore, effectively identifying these groupings requires mitigating this unevenness, which we address using a UMAP-based fuzzy graph construction.

3.1 Graph Construction

The first step in the Uniform Manifold Approximation and Projection (UMAP) algorithm is to approximate the manifold by constructing a fuzzy topological representation of the embeddings using nearest neighbor descent (McInnes et al., 2020). Inspired by that, we use the same nearest neighbor descent method to find the K nearest neighbors for every token embedding in the embedding space and then use the same equations used in UMAP's fuzzy graph construction to define the weight function of the edge between x_i and x_j nodes in the K-NN graph (McInnes et al., 2020):

$$\omega(x_i, x_j) = exp\left(\frac{-\max(0,\ d(x_i, x_j) - \rho_i)}{\sigma_i}\right) \tag{1}$$

where $d(\cdot, \cdot)$ is the distance function (cosine in our case) and ρ_i is calculated as:

$$\rho_i = \min\{d(x_i, x_j) \mid 1 \leq j \leq k,\ d(x_i, x_j) > 0\} \tag{2}$$

where k is the number of neighbors of node i. Finally, σ_i is calculated by setting the summation of weights of a node to be equal to a constant (i.e., $\log_2(k)$):

$$\sum_{j=1}^{k} \omega(x_i, x_j) = \log_2(k). \tag{3}$$

Building upon our theoretical arguments, the conceptual/categorical representations (if they exist),

should form fuzzy partitions that can be detected by the community detection algorithms. Note that since UMAP dimensionality reduction process can lead to information loss (Geiger and Kubin, 2012; Wang et al., 2021), potentially obscuring important nuances in the representations, we perform community detection in the high-dimensional space.

3.2 Louvain Community Detection

The Louvain method is a widely used algorithm for community detection in large networks. It finds the communities by optimizing a metric called modularity. The modularity of a partition is a scalar value between -1 and 1 that measures how much more densely connected the nodes within a community are compared to how connected they would be in a random network. (Blondel et al., 2008). For a weighted graph, modularity is defined as:

$$Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j) \qquad (4)$$

where A_{ij} represents the weight of the edge between i and j; k_i is the sum of the weights of the edges attached to vertex i; m is the sum of all of the edge weights in the graph; the δ-function $\delta(c_i, c_i)$ is 1 if $u = v$ and 0 otherwise; c_i is the community to which the nodes i belongs to.

Then, it aggregates the communities to identify possible hierarchical structures. In this phase, each community is considered as a single node and the links between the new nodes are calculated as the sum of the weight of the links between nodes in the corresponding two communities. More details are given in appendix C, algorithm 2.

3.3 Concept Extraction Algorithm

For our concept extraction algorithm, as the first step, we create and weight the adjacency graph using K-NN, UMAP-based weighing formula (mentioned in section 3.1), and then use Louvain algorithm. Algorithm 1 describes the concept extraction process (see appendix E for details on the algorithm methodology and considerations). We configured our algorithms to create k-NN graph iteratively for different values of k. This enables us to observe the communities/concepts formation at various granularities. Table 1 shows the number of identified clusters for $k = [100, 75, 50, 25, 12, 6]$.

Hierarchy Formation. When examining the broader perspective (i.e., k=100), the model primarily found groups of named entities (names of people and places), adverbs, sub-words, some number symbols, and etc. (appendix J, Figure 5 shows the overall concept hierarchies of the Albert vocab). Zooming in further (e.g., k=75), these communities revealed more specific clusters that are relatable to the real world. For instance, within named entities, clusters formed for personal vs. location names, even further pinpointing locations by country. As the granularity level increases (approaching a smaller k value), clusters exhibit a stronger association with word forms.

Table 1: Number of communities with different granularities of nearest neighbors for Albert, T5, Glove, and Llama3. For Glove, we only used the subset of GloVe that present in Albert vocabulary

	K settings						Vocab
Models	100	75	50	25	12	6	Size
T5	1	115	1203	4551	8137	9407	32000
Albert	8	133	1058	4442	7718	8626	30000
GloVe	9	207	1157	3521	6237	7200	25869
Llama-3	7	23	844	6044	18644	32535	128256

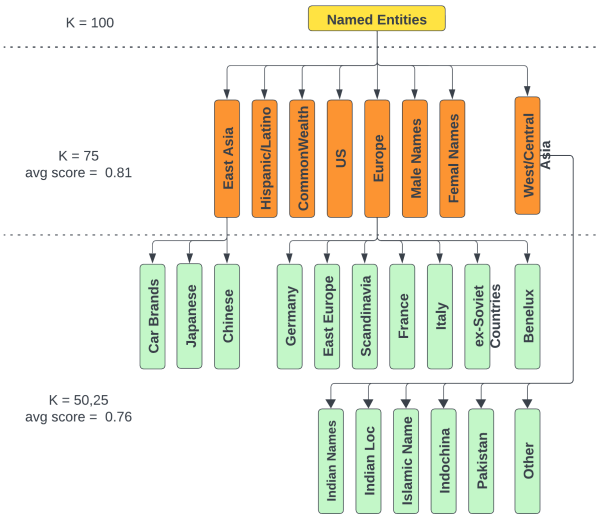

Figure 1: The identified name and location communities for different k granularity for Albert model. At the left-side the average precision score for the extracted graph within each granularity is given. For the more detailed tables, see H (Note that the results for other models are also available in appendix I).

4 Evaluation: Alignment with External Knowledge

We extracted several meaningful conceptual communities, ranging from symbo-numerical groups, to concrete objects and Named-Entities (such as plant groups, animals, car brands, names, and locations), to more abstract groupings such as colors, social roles and structures, currencies, etc. (more

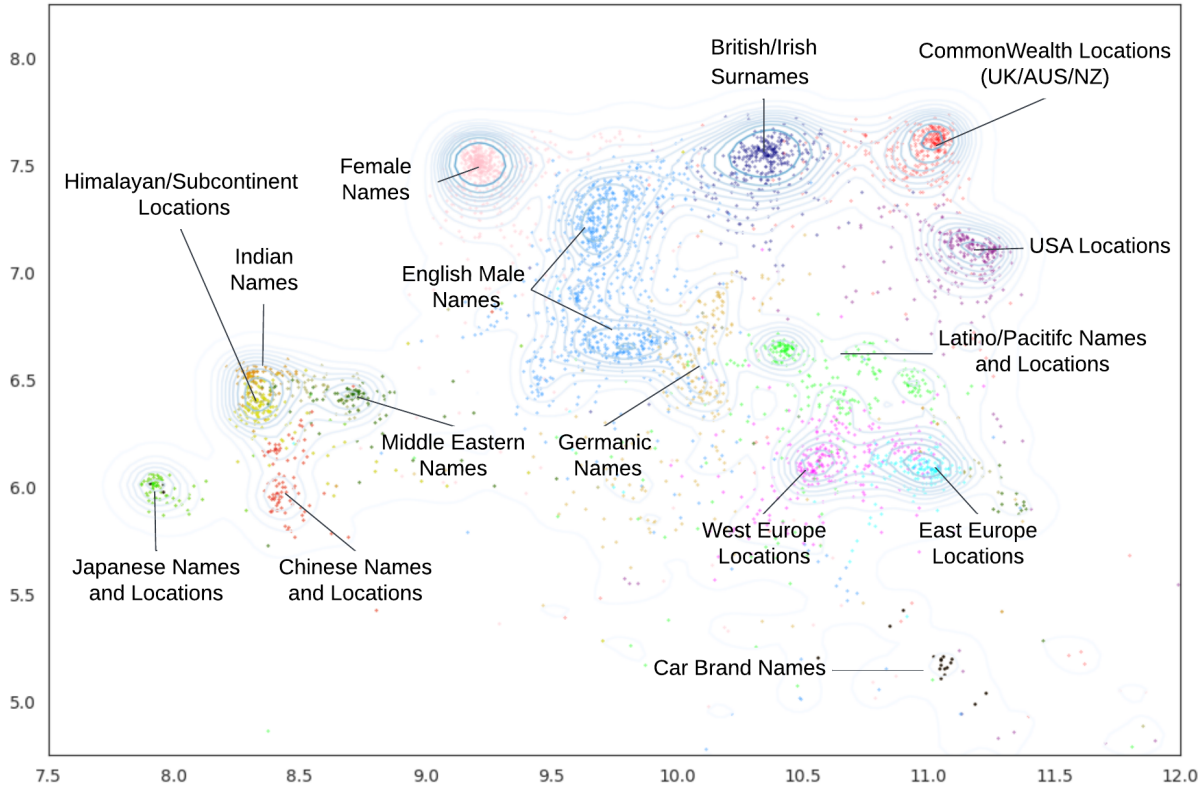

Figure 2: Visualization of the identified name and location communities of size larger than 10 entities. UMAP projection along with Seaborn (Waskom, 2021) is used for the visualization.

details in appendix J).

To evaluate RQ1 and RQ2, we selected named entities as well as numbers, primarily because the existence of external evaluation datasets for NE as well as the well defined nature of numbers allow us to examine (1) the alignment with external world knowledge, and (2) to observe within group properties of the conceptual communities. To assess the consistency of conceptual groupings across different LLMs, we applied our method to the embedding spaces of GloVe (Pennington et al., 2014), Albert (Lan et al., 2019), T5 (Raffel et al., 2020), and Llama-3 (met, 2024) (See Appendix F). For brevity, we present the overview of the results for Albert in the main section (for more detailed table for the all aforementioned models please refer to Appendix H and I)

4.1 Named Entities

To evaluate the quality and the alignment of the extracted conceptual grouping with the external world, we used we leveraged the name-dataset (Remy, 2021) and the country-state-city database (Gada, 2018). These datasets provide comprehensive information on names (including gender and country-specific popularity) and locations (including hierarchical relationships between countries, states, and cities), enabling a rigorous assessment of the conceptual groupings (see appendix H for more details).

The granularity of observed clusters was influenced by both the number of constituent samples and conceptual attributes. For instance, English/American named entities and locations tended to form distinct clusters at higher levels of granularity (k values) due to their high frequency within the vocabulary. However, upon further increasing granularity (higher k values), hierarchical structure emerged within these clusters. The conceptual attributes contributing to this hierarchical organization within named entities included (1) entity type (human name or location), (2) name part (first or last), gender, and (3) regional/national origin.

Note that since there's no one-to-one mapping between names/locations in LLM vocabulary and external datasets (e.g., a name may appear in multiple countries, the external dataset is also a superset), recall is less relevant. We prioritize precision to evaluate accuracy, as it better reflects our ability to identify correct matches. We observe high precision score across the identifies name entity cluster. Figure, 1 shows the main groupings for Al-

bert Model. We got average precision score of 0.81 for K=75 and 0.76 for K=50/25. Appendix H.1 contains the detailed tables for different models. Furthermore, we observed a degree of geographic ordering within the identified communities. As illustrated in Figure 2, there appears to be a general trend from east to west as we move across the communities from bottom-left to top-right. The leftmost communities are predominantly associated with Japanese locations and names, while those on the rightmost side are primarily linked to Europe and the United States. This suggests that the model's internal representations in the input embedding layer may inherently capture geographical relationships.

Table 2: Topological ordering score of different communities of the numbers. Years refer to the cluster of numbers between 1816 - 2021. Support refers to the number of samples within the cluster.

Category	k-1	k-3	k-5	Support
0-100	0.68	0.89	0.92	100
100-262	0.75	0.94	1.0	115
263-300	0.46	0.78	0.89	37
300-400	0.42	0.71	0.84	82
400+	0.53	0.75	0.83	110
Years	0.86	0.96	0.98	203

4.2 Symbols-Numbers

At a high level, symbols and numbers are distinctly separated from other tokens in the embedding space. Within their own domain they form communities according to (i) years, (ii) integer values, (iii) tokens indicating monetary values (e.g., \$1), (iv) ratio/time (e.g., 2:1, 3:30), (v) fractions (e.g., '/8'), (vi) large values (e.g., '100,000'), and (vii) percentages (e.g., 42%). In these communities, integers create sub-communities based on their hundreds.

To align with the human notion of numbers, it is essential to identify a topological ordering within the numerical embeddings. As the embedding manifold may not conform to a Euclidean structure (Law et al., 2019; Chen et al., 2021; Cai et al., 2020), conventional distance-based order measures were inapplicable. Lemma 5 formulates the local ordering of embeddings within a embedding manifold.

Lemma 1. *Local Ordering on a Manifold: Let M be a manifold and let d(a, b) denote the distance between points a and b on M. For a given positive integer k, we say that a point x on M is locally ordered if and only if:*

$$x \in top_k(x+1) \cap top_k(x-1) \tag{5}$$

where $top_k(a)$ denotes the set of k-nearest neighbors of point a on M.

This localized approach captures the intuitive concept of topological ordering within a small neighborhood, rather than relying on a global ranking.
Topological Ordering Score is defined as:

$$S = \frac{1}{n}\sum_{i=0}^{n} f_k(x_i) \tag{6}$$

where n indicates the number of embeddings in the given cluster, k is an integer that controls the strictness of ordering, and $f_k(x_i)$ is boolean function that returns 1 if x_i hold lemma 5 condition.

Table 2 presents the topological ordering scores of the communities identified by our algorithm. The "k-1" score represents strict topological ordering, where $x+1$ lies exactly between x and $x+2$ in the embedding space. The "k-3" and "k-5" scores measure relaxed topological ordering, where $x+1$ falls within the 3 or 5 nearest neighbors of both x and $x+2$, respectively. As the results indicate, all detected communities exhibit a high degree of local topological ordering, regardless of the chosen level of strictness. This finding is significant because it suggests that LLMs may possess not only the ability to categorize heterogeneous input entities but also the capacity to construct meaning within smaller, internally consistent structures (internally homogeneous sub-structures). This implies the potential for LLMs to move beyond simply classifying information to actively interpreting and generating meaning within specific clusters.

5 LLM-LLM alignment

Investigating LLM-LLM alignment in the input embedding layer is crucial to answer RQ3, because it reveals how well different LLMs represent the same concepts in their initial processing stages. This analysis provides valuable insights into the influence of model architecture, size, and training regimes on the formation of language representations. To quantify this alignment, we defined an alignment score that specifically captures the discrete overlaps of nearest neighbors in embedding spaces. This score offers a precise assessment of how similarly the LLMs represent shared tokens, enabling a more nuanced understanding of their

Table 3: Pair-wise alignment score for various LLM sizes, and training regimes. higher alignment score is better.

Model-1	Model-2	k=3	k=5	k=10	k=50	Support
Albert-XXL	Albert-base	0.501	0.490	0.472	0.438	30000
Albert-XXL	Albert-L	0.56	0.54	0.52	0.48	30000
Albert-XXL	Albert-XL	0.60	0.58	0.55	0.51	30000
T5-11B	T5-small	0.61	0.577	0.52	0.38	32100
T5-11B	T5-base	0.68	0.66	0.60	0.51	32100
T5-11B	T5-large	0.74	0.71	0.66	0.61	32100
T5-11B	T5-3B	0.79	0.76	0.71	0.63	32100
Llama3-70B	Llama3-1B	0.57	0.55	0.50	0.31	128256
Llama3-70B	Llama3-3B	0.56	0.53	0.48	0.27	128256
Llama3-70B	Llama2-70B	0.65	0.61	0.56	0.47	22430

representational alignment.

We define **Alignment Score** as:

$$S = \frac{1}{n} \sum_{j=0}^{j=n} \frac{|top\_k(LLM_j^1) \cap top\_k(LLM_j^2)|}{k} \tag{7}$$

where LLM^1 and LLM^2 denote the two LLMs under investigation, n represents the number of shared tokens between the LLM pair, and $top\_k(LLM_j^1)$ and $top\_k(LLM_j^2)$ represent the sets of the top k nearest neighbors of the j-th token in LLM^1 and LLM^2 respectively. The alignment score for two randomly chosen embeddings with the same vocabulary is 0 (assuming $k << n$).

We calculated this alignment score across a range of LLMs, encompassing diverse architectures and training objectives. we first identify the set of tokens shared across the vocabularies of a given pair of LLMs. For each shared token, we compute its k nearest neighbors ($k = 3, 5, 10, 50$) in the embedding space of both models. We use pairwise cosine similarity measure to find top-k tokens within an LLM. This selection included Albert (encoder-based), T5 (encoder-decoder), Llama 70B (decoder-only) to examine the impact of high-level model architecture. To investigate the effect of model size, we included T5 models of different scales: T5-small, T5-base, T5-large, T5-3B, and T5-11B. Analyzing results from tables 3 and 4, we observe:

- Table 3 shows that model size is a contributing factor in achieving higher alignment scores when comparing within the same architecture. Since larger models also demonstrate better performance on benchmark tasks, we can infer that the quality of the concepts formed in the input embedding layer is positively correlated with model size, assuming architecture and training regimes are held constant.
- Both tables 3 and 4 show that alignment scores generally decrease as the value of k increases. This trend suggests that while models share a core understanding of semantic similarity at smaller k values, they diverge when considering more generic concepts at larger k values. The alignment drop is larger in decoder-only models, likely due to their unidirectional context [2].
- Interestingly, the alignment scores between Llama2-70B and Llama3-70B are comparable to those between Llama3-70B and T5-11B (refer last rows of table 3 and table 4). There are minimal architectural differences between Llama2 and Llama3-70B (primarily in the context window size). This implies that factors such as training regimes (e.g., dataset size and composition) and context window size are as influential as model architecture in achieving strong alignment.
- Table 4 shows the cross-architecture alignment scores for different LLMs, comparing models of similar sizes but varying architectures. Despite architectural differences, models of similar sizes exhibit moderate to high alignment scores (mostly above 0.5), suggesting a consistent semantic organization in LLMs.

Overall, we observed moderate to high alignment across LLMs, regardless of their size, architecture, or pretraining, indicating that the findings in Section 4 may generalize to other LLMs.

Table 4: Cross architecture alignment score for various LLM architectures, sizes, and training regimes. higher alignment score is better. Support column indicates the number of shared tokens between the LLM pair.

Model-1	Model-2	k=3	k=5	k=10	k=50	Support
Albert-XXL	Llama3-3B	0.55	0.52	0.49	0.44	18640
Albert-XXL	T5-3B	0.63	0.61	0.59	0.54	12307
Llama3-3B	T5-3B	0.66	0.62	0.58	0.53	20603
Llama3-70B	T5-11B	0.64	0.61	0.56	0.47	20603

6 Bias Mitigation: Case Study of Cluster Modification

To investigate RQ4 and demonstrate the practical impact of our findings, we conduct a case study on modifying conceptual clusters to mitigate ethnicity bias. A key challenge here is balancing bias reduction with the preservation of the model's overall

[2]This is consistent with human studies showing that readers' eye movements are bidirectional: forward to absorb new information, and backward to resolve comprehension issues or correct errors (Staub and Rayner, 2007).

utility and linguistic integrity. Our approach focuses on embedding engineering, targeting clusters of tokens associated with stereotype-prone identities, such as proper nouns from the Indian subcontinent (e.g., Indian and Pakistani human names). We hypothesize that by modifying these token embeddings, we can disrupt learned biases without sacrificing the tokens' semantic roles within the language model. For example, we aim to investigate if the token "Sharma", a member of the Indian proper nouns cluster, retains its identity as a proper noun after token engineering. To evaluate fairness score, we use Bias Benchmark for Question Answering (BBQ) dataset (Parrish et al., 2022), the fairness evaluation methodology is based on the WinoBias evaluation(Zhao et al., 2018). To assess the linguistic preservation, we chose the part-of-speech (POS) tagging task as a proxy indicator on CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003) as well as a subset of Wikimedia dataset (Wikimedia) (Further evaluation on the robustness of embedding engineering on GLUE, SuperGLUE, and SQUAD (Wang et al., 2018, 2019) benchmarks is provided in Appendix D to provide a more comprehensive assessment of general language model quality after the token manipulations).

Our approach begins by selecting a cluster of token embeddings from the original model. For this experiment, we used the communities associated with Indian and Pakistani human names that was identified by our concept extraction algorithm. Then, we calculated the mean and standard deviation of the joint cluster. Finally we modified these clusters to form a single cluster with Gaussian distribution of the joint cluster (i.e. samples from the $Gaussian(\mu, 0.7 * \sigma)$). Then, we fine-tuned the model for POS tagging task for CoNLL-2003 dataset fro 5 epochs (see Appendix B for more details). For the POS tagging task, we utilize the CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003), a widely-used benchmark for named entity recognition and POS tagging. Each model is fine-tuned using a standard supervised fine-tuning approach.

As shown in Table 5, all token-engineered models exhibit over 90% token overlap with their base counterparts. This high overlap (like the >90% reported) suggests that even though the embeddings have been modified, they still largely represent the same underlying POS tags. Furthermore, for those POS tags where the base and modified models show disagreement, we still observe the same overall quality. This means that even where the model does change its POS tag assignment, the new assignment is just as likely to be correct as the original. At the same time, we are able to mitigate bias ranging from 44% for the Albert-xxl-large and T5-3b models to 52% for the T5-11b model. These highlight the potential of token engineering for bias mitigation while keeping the general purpose utility of the model intact.

Table 5: Fairness and POS Tagging Performance of Various Models. Fairness: higher score implies less bias. POS: higher score implies better accuracy.

Model	Fairness		POS Tagging		
	Base Score	Ours Score	Base Acc	Our Acc	Overlap %
Albert-base	0.26	0.74	0.91	0.91	0.90
Albert-xxl	0.28	0.72	0.93	0.91	0.92
T5-3b	0.28	0.72	0.92	0.91	0.94
T5-11b	0.24	0.76	0.94	0.93	0.94

7 Conclusion

In this paper, we propose a modular concept extraction mechanism that uncovers the emergence of distinct conceptual communities within the entire input embedding space. Using our methodology, we observe that LLMs form organized conceptual structures within their input embedding spaces. We demonstrate that the input embeddings of LLMs form categorical semantic structures that align with external world representations. We quantitatively analyze several properties of these structures, with a particular focus on categorical structures related to named entities. Additionally, we observe that numerical structures within the input embedding layer align with human notion of numerical values, including a topological ordering of numbers. We also discussed that LLMs inherently exhibit a degree of alignment with one another, suggesting the potential to extend the observed human-LLM alignment to other models. This study opens new avenues for further exploration and intervention in LLMs, especially within the realm of embedding engineering in several key areas, including bias detection and mitigation.

8 Limitations and Risks

Limitations: The model forms the conceptual communities that are meaningful but its priority is not exactly the same as that of humans. The model vocabulary is a contributing factor to the way the model prioritizes the formation of conceptual clusters in its embedding layer. For example, the number of English names is much higher than the other languages and this has caused the model to form high-level communities (e.g., k=75) specified for names vs. less frequent names/locations a high-level community contains the combination of regions personal and location names. This limits our method to associate the KNN resolution with the abstraction level of the extracted concepts/categories.

Risks: This work provides detailed information about (1) the formed clusters/concept in the input embedding layer, and (2) the separation of memory from reasoning in Albert. As the methodology can also be applied to other models, it can potentially facilitate more advanced adversarial attacks and content manipulation in LLMs.

References

2024. Introducing meta llama 3: The most capable openly available LLM to date.

Mostafa Abdou, Artur Kulmizev, Daniel Hershcovich, Stella Frank, Ellie Pavlick, and Anders Søgaard. 2021. Can language models encode perceptual structure without grounding? a case study in color. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 109–132.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Jeffrey R Binder and Rutvik H Desai. 2011. The neurobiology of semantic memory. *Trends in cognitive sciences*, 15(11):527–536.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

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.

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. 2020. Isotropy in the contextual embedding space: Clusters and manifolds. In *International conference on learning representations*.

Boli Chen, Yao Fu, Guangwei Xu, Pengjun Xie, Chuanqi Tan, Mosha Chen, and Liping Jing. 2021. Probing bert in hyperbolic spaces. *arXiv preprint arXiv:2104.03869*.

Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 1–6, Berlin, Germany. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Darshan Gada. 2018. Countries states cities database. https://github.com/dr5hn/countries-states-cities-database.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Peter Gärdenfors. 2020. Primary cognitive categories are determined by their invariances. *Frontiers in Psychology*, 11:584017.

Bernhard C Geiger and Gernot Kubin. 2012. Relative information loss in the pca. In *2012 IEEE information theory workshop*, pages 562–566. IEEE.

E.B. Goldstein. 2009. *Sensation and Perception*. Cengage Learning.

Yuling Gu, Bhavana Dalvi Mishra, and Peter Clark. 2023. Do language models have coherent mental models of everyday things? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1892–1913, Toronto, Canada. Association for Computational Linguistics.

Nishant Gurnani. 2017. Hypothesis testing based intrinsic evaluation of word embeddings. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 16–20, Copenhagen, Denmark. Association for Computational Linguistics.

Wes Gurnee and Max Tegmark. 2024. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*.

James A Hampton. 2007. Typicality, graded membership, and vagueness. *Cognitive Science*, 31(3):355–384.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Marc Law, Renjie Liao, Jake Snell, and Richard Zemel. 2019. Lorentzian distance learning for hyperbolic representations. In *International Conference on Machine Learning*, pages 3672–3681. PMLR.

Na Li, Zied Bouraoui, Jose Camacho-Collados, Luis Espinosa-Anke, Qing Gu, and Steven Schockaert. 2021. Modelling general properties of nouns by selectively averaging contextualised embeddings. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 3850–3856. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. Towards understanding and mitigating social biases in language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6565–6576. PMLR.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

Bradley C Love and Todd M Gureckis. 2007. Models in search of a brain. *Cognitive, Affective, & Behavioral Neuroscience*, 7(2):90–108.

N. Lukas, A. Salem, R. Sim, S. Tople, L. Wutschitz, and S. Zanella-Beguelin. 2023. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363, Los Alamitos, CA, USA. IEEE Computer Society.

Ben Mann, N Ryder, M Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Leland McInnes, John Healy, and James Melville. 2020. Umap: Uniform manifold approximation and projection for dimension reduction. *Preprint*, arXiv:1802.03426.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

John Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander Rush. 2023. Text embeddings reveal (almost) as much as text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12448–12460, Singapore. Association for Computational Linguistics.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4658–4664, Florence, Italy. Association for Computational Linguistics.

Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and William Wang. 2023. On the risk of misinformation pollution with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1389–1403, Singapore. Association for Computational Linguistics.

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel Bowman. 2022. BBQ: A hand-built bias benchmark for question answering. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2086–2105, Dublin, Ireland. Association for Computational Linguistics.

Roma Patel and Ellie Pavlick. 2021. Mapping language models to grounded conceptual spaces. In *International Conference on Learning Representations*.

Hao Peng, Xiaozhi Wang, Shengding Hu, Hailong Jin, Lei Hou, Juanzi Li, Zhiyuan Liu, and Qun Liu. 2022. Copen: Probing conceptual knowledge in pre-trained language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5015–5035.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word

representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Jason Phang, Phil Yeres, Jesse Swanson, Haokun Liu, Ian F. Tenney, Phu Mon Htut, Clara Vania, Alex Wang, and Samuel R. Bowman. 2020. `jiant` 2.0: A software toolkit for research on general-purpose text understanding models. `http://jiant.info/`.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Philippe Remy. 2021. Name dataset. `https://github.com/philipperemy/name-dataset`.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rita Sevastjanova, Aikaterini-Lida Kalouli, Christin Beck, Hanna Schäfer, and Mennatallah El-Assady. 2021. Explaining contextualization in language models using visual analytics. 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 464–476, Online. Association for Computational Linguistics.

Congzheng Song and Ananth Raghunathan. 2020. Information leakage in embedding models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 377–390.

Adrian Staub and Keith Rayner. 2007. Eye movements and on-line comprehension processes. *The Oxford handbook of psycholinguistics*, 327:342.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2018. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Avijit Thawani, Biplav Srivastava, and Anil Singh. 2019. SWOW-8500: Word association task for intrinsic evaluation of word embeddings. In *Proceedings of the 3rd Workshop on Evaluating Vector Space Representations for NLP*, pages 43–51, Minneapolis, USA. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. 2021. Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *Journal of Machine Learning Research*, 22(201):1–73.

Michael L. Waskom. 2021. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

Wikimedia. English wikipedia dump.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018. Gender bias in coreference resolution: Evaluation and debiasing methods. *Preprint*, arXiv:1804.06876.

Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from labeled and unlabeled data with label propagation. *ProQuest Number: INFORMATION TO ALL USERS*.

A Semantic Representation Space

A.1 conceptual groupings, Representation Learning and Semantic Memory

A simplified model of human understanding can be described as a multi-step process in which the incoming sensory information is integrated and turned into a brain-constructed interpretation of external objects/stimuli. This results in a mental construct (or percept) (Goldstein, 2009). Then, the newly formed mental construct gets compared, integrated, and associated with the existing knowledge already retained in the semantic memory. These connections help us to conceptualize and understand the new information based on what we already know (as shown in Figure 3). While diverse reasoning mechanisms exist, a common thread among them seems to be their reliance on and interaction with the semantic memory. This interaction likely leaves enduring traces within such a memory (Binder and Desai, 2011). The integration of newly encountered mental constructs within an existing cognitive framework is often guided by their relations to established internal constructs (Gärdenfors, 2020). This process facilitates the creation, refinement, or expansion of semantically similar clusters (Love and Gureckis, 2007), implying a degree of inherent categorization[3].

Representation Learning and Conceptual roupings) Representation learning is a fundamental aspect of language models, where the goal is to learn distributed representations (embeddings) for words or subword units that capture their semantic and syntactic relationships (Bengio et al., 2013). Early models like word2vec (Mikolov et al., 2013) utilized shallow neural networks to learn these embeddings from large text corpora. More recent models, such as Large Language Models (LLMs) like GPT-3 (Mann et al., 2020), employ transformer architectures with self-attention mechanisms (Vaswani et al., 2017), enabling more accurate and dynamic representation learning. From the distributional hypothesis, models can form "concepts" by identifying patterns and relationships within data, particularly through recognizing the approximate invariance of shared features across different data instances (Gärdenfors, 2020). It should be noted that "conceptual groupings" is a byproduct of the learning process, not a guaranteed outcome. Furthermore, the formed "concepts" might not always align with human-defined concepts. Thus, it is crucial to investigate and evaluate the nature of conceptual groupings within models and their alignment with human understanding.

[3]This notion implies an interrelation between the recognition and categorization.

A.2 Evaluation Methods

The analysis, measurement, and interpretability of semantic representation learning in language models have been the subject of extensive research. Various methods have been proposed to evaluate how well these models capture semantic meaning.

Intrinsic evaluation methods, assess semantic representations by measuring word embedding similarity, comparing them to human judgments of relatedness, (e.g. (Mikolov et al., 2013; Gurnani, 2017; Thawani et al., 2019; Niven and Kao, 2019; Tenney et al., 2019)). While these methods provide valuable insights, they often suffer from the limitations that probing techniques impose, and may not fully capture how well a model forms conceptual hierarchies or grasps the full spectrum of semantic relationships, as word embeddings typically represent individual words rather than higher-level categories (Chiu et al., 2016). This limitation can hinder the evaluation of a model's ability to understand broader semantic relationships and its capacity for abstract reasoning.

Another approach is to use *extrinsic evaluation methods*, which measure the performance of language models on downstream tasks that rely on semantic understanding, such as sentiment analysis, question answering, and machine translation (Wang et al., 2018, 2019). The performance on these tasks can indirectly indicate the quality of the learned semantic representations. The main challenge with using such extrinsic evaluation metrics to assess conceptual groupings in the embedding space is that they do not directly measure the quality of the concepts themselves. Instead, they measure how well the model performs on specific tasks that rely on those concepts, which does not necessarily prove that the model has formed robust, human-like concepts. For instance, a model might accurately classify sentiment without truly grasping the nuances of emotions like sarcasm or irony (McCoy et al., 2019).

In addition to evaluation metrics, researchers have also employed visualization methods to interpret learned representations. These methods often involve projecting high-dimensional embeddings into lower-dimensional spaces for visualization (Sevastjanova et al., 2021; Rogers et al., 2021).

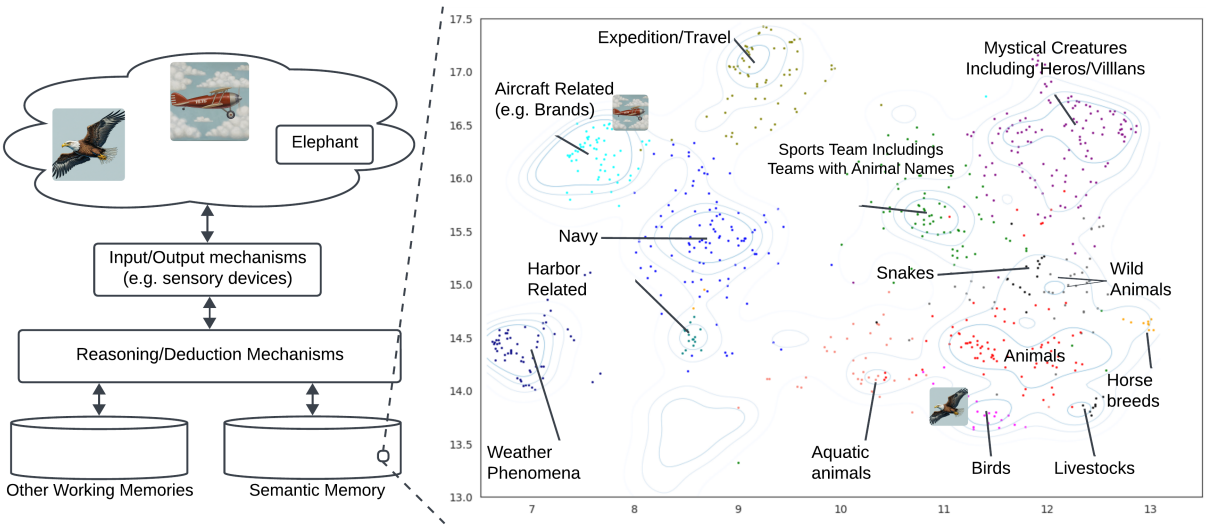

Figure 3: Simplified steps on how external information is understood and retained. Upon understanding a newly encountered word/entity, it is typically stored in the semantic memory. The existence of semantic memory (on the left) allows the previously encountered words/entities to have a form of meaning even without requiring an external context. The scatter box on the right is the community (primarily associated with moving creatures) we extracted from the Albert model (Lan et al., 2019).

However, several challenges arise with these approaches. Firstly, they offer indirect and subjective assessments of model understanding, lacking a quantitative basis for evaluation. Secondly, the dimensionality reduction process can lead to information loss (Geiger and Kubin, 2012; Wang et al., 2021), potentially obscuring important nuances in the representations. Finally, different visualization techniques can produce conflicting results, making it difficult to reach definitive conclusions about the model's true comprehension. To mitigate these shortcomings, we propose a concept extraction mechanism that identifies communities in higher-dimensional space. This approach enables both quantitative evaluation and mitigation of potential information loss incurred during dimensionality reduction.

B Bias Mitigation details

We evaluated the models on subsets of the Wikimedia dataset (Wikimedia), where we sample only those sentences that contain the tokens from our token-engineered cluster. For the fairness task, we adopt the Bias Benchmark for Question Answering (BBQ) dataset (Parrish et al., 2022), focusing on the nationality split. We modify the data set for masked language modeling (MLM) by replacing the interrogative questions with a token '[MASK]'. To ensure the quality of evaluation, each sample is manually checked for grammatical correctness following this transformation. For evaluating fairness, we compare each base model with its corresponding token-engineered model that utilizes Gaussian-sampled embeddings. We first filter out the evaluation samples that pass the fairness test on the base model, thus isolating only the problematic cases. Our evaluation of fairness is inspired by the evaluation metrics used for the winobias dataset (Zhao et al., 2018).For the remaining biased samples, we calculate which model (base or token-engineered) is more likely to generate a biased output by examining their output logit probability.

C Lovain Algorithm

Evaluation Metrics. conceptual groupings involves the creation of abstract internal representations through the clustering of inputs with shared features. While numerous intrinsic clustering metrics exist to assess cluster formation quality, their application to our use case is limited by two factors:

- The formed clusters are not situated within Euclidean space (Law et al., 2019; Chen et al., 2021; Cai et al., 2020), rendering geometric properties such as cluster distances inadequate indicators of concept well-formedness and distinctiveness.
- Concepts inherently possess a degree of vagueness (Hampton, 2007), thus metrics like compactness or separation do not reliably reflect the

Algorithm 2: Louvain

Data: The initial input is a weighted network of all the nodes exist in the entire space.

Result: A set of hierarchical communities.

```
Community Detection;
Create a community list; assign a different community id to each node of the network;
while a local maxima of the modularity is not attained do
    for each node i do
        for each neighbor j do
            evaluate the gain of modularity if i moved to the community of node j;
            keep the maximum gain and community id;
        if the maximum gain is positive then
            Move node i to the community with maximum gain.
Community Aggregation;
if Number of Communities > 1 then
    Reduce each community to a single node;
    go to 1
```

quality of formed concepts.

D Knowledge-Reasoning Separation

Thus far, we have demonstrated that the LM constructs a knowledge base (mental representation) directly within its input embedding layer. Furthermore, we have established a degree of human-LM alignment in both hierarchical structure and semantic meaning. Now, we are interested to see the extent to which the knowledge learned during pretraining is modular and separable from the reasoning mechanisms employed by the language model in downstream tasks. For example, can the knowledge learned during the pretraining phase be selectively removed or modified without significantly impacting the model's performance on finetuning? The modularity can also impact the effectiveness of Language Model Inversion (Morris et al., 2023; Song and Raghunathan, 2020) techniques, which aim to extract private information such as names or other sensitive details learned during the pretraining[4].

To investigate this, we selected GLUE, SuperGLUE, and SQUAD benchmarks as downstream tasks to assess language model performance. We then systematically removed within-community information by calculating and assigning the embedding space center (mid-point value) of each community to all its members. For example, in a community of names like "James," "John," and "Alex," all members would share the same embedding. Table 6 shows the results on major LM benchmarks GLUE, SuperGLUE, and SQUAD for the Albert base model. Although our experiment does not prove the separation of knowledge and reasoning, it indicates that at least the granular information acquired during the pretraining is not required for the model's performance on the aforementioned LM benchmarks. This is significant because it opens the door for embedding engineering of private or harmful information that is learned during the pretraining.

E Hierarchical Community Extraction: Methodology and Considerations

There are multiple viable strategies to extract hierarchical communities in our methodology; the first one is to use algorithms such as Louvain that inherently generate hierarchical communities (Blondel et al., 2008). However, the summation of the weights as a new weight in the community aggregation phase of Louvain algorithm skews the weighted graph in favor of merging smaller communities in the next phase. This detaches the community detection from the actual values in the semantic representation space (i.e. graph weights higher in the Louvain hierarchy no longer reflect the geometrical affinity of the nodes). Thus, for our concept extraction algorithm, we use 1-2 community aggregation and rather use KNN iteratively with different granularity for extracting hierarchical concepts (more details are described in Algorithm 1).

Note that we only use well-established methods such as K-NN, UMAP-based weighing formula, as well as Louvain (Blondel et al., 2008) and label propagation (Zhu and Ghahramani, 2002) community detection algorithms to capitalize on

[4]For example, if knowledge is found to be highly modular, it may be possible to develop targeted interventions that obscure or remove specific sensitive information without significantly impacting the model's overall performance on downstream tasks.

Table 6: Huggingface Albert base model on GLUE, SuperGLUE, and SQUAD tasks. For the baseline, the model was finetuned without altering the embeddings. For the mid-point, the embedding layer entries are assigned the mid-point embedding of their associated community. We used (Phang et al., 2020) repo for benchmarking.

mid-point	GLUE								
Method/Tasks	mnli	mrpc	qnli	qqp	rte	sst	stsb	wnli	
baseline	0.827	0.841	0.902	0.858	0.765	0.915	0.872	0.549	
mid-point	0.849	0.865	0.910	0.875	0.783	0.922	0.890	0.563	
mid-point	SuperGLUE							SQUAD	SQUAD
Method/Tasks	boolq	cb	copa	multirc	record	wic	wsc	v1 (f1)	v2 (f1)
baseline	0.622	0.512	0.59	0.350	0.586	0.595	0.528	83.72	70.9
mid-point	0.621	0.478	0.55	0.372	0.588	0.626	0.634	84.4	74.9

the established generalizability of these algorithms. Although, as we show in the next sections that our method produces amazingly good categories, it should be noted that we intend to focus our analysis on "if the language model forms concepts" rather than creating the most optimal concept extraction mechanism. Thus, as an extension to this work, one can focus on further optimizing our proposed method. Notably, our concept extraction is algorithm-agnostic; alternatives could be readily employed.

F Models Under Investigation

We used huggingface repository for all our models.

F.1 Albert

Albert (A Lite BERT) (Lan et al., 2019) is a transformer-based model for language representation learning, designed to be more efficient than its predecessor, BERT (Devlin et al., 2018). While it shares the same basic architecture as BERT, it incorporates two main key modifications:

- **Factorized embedding parameterization**. The benefit of factorized embedding parameterization in Albert is the significant reduction in the number of parameters compared to models like BERT. In BERT, the word embedding size (E) is tied to the hidden layer size (H), leading to a large embedding matrix as H increases. Albert instead factorizes this embedding into two smaller matrices, one projecting token ID vectors to a lower-dimensional space (E) and another projecting from this space to the hidden layer (H). This allows H to be much larger than E without increasing the parameter count of the embedding layer substantially, resulting in a more efficient use of parameters. This is particularly beneficial for large models, where memory limitations can hinder training and deployment.
- **Cross-layer parameter sharing**. Parameter sharing acts as a form of regularization, preventing the model from overfitting to specific layers or features in the data. This can lead to improved generalization performance on unseen data.

Note that reducing the number of parameters and sharing information across layers can force the model to learn more general representations, thus indirectly contributing to better conceptual groupings. **datasets.** Albert is pretrained on English text datasets, namely the English Wikipedia and BookCorpus, using self-supervised learning objectives. **Tokenization.** It uses Sentencepiece tokenizer (Kudo and Richardson, 2018) on the uncased corpus with a vocabulary size limit of 30K tokens.

F.2 T5

T5, or Text-to-Text Transfer Transformer, is a transformer-based architecture that casts all natural language processing (NLP) tasks into a text-to-text format. This means the model takes text as input and generates text as output, regardless of the specific task. At its core, T5 is an encoder-decoder model with the following key components:

- **Encoder:** This component takes the input text and processes it into a sequence of hidden representations. It uses multiple transformer layers, each consisting of self-attention mechanisms and feedforward neural networks.
- **Decoder:** This component generates the output text auto-regressively, conditioned on the encoder's hidden representations. It also uses multiple transformer layers with self-attention and feedforward networks, as well as cross-attention mechanisms to attend to specific parts of the input sequence.

Datasets. T5 is pre-trained on a massive dataset called C4 (Colossal Clean Crawled Corpus), which contains around 750 GB of clean English text.

Tokenization. It uses Sentencepiece tokenizer on the cased corpus with a vocabulary size limit of 30K tokens.

F.3 GloVe

GloVe (Global Vectors for Word Representation) is a method for obtaining vector representations for words. Unlike context-based models like transformer-based LM models, GloVe leverages global word co-occurrence statistics across a corpus to learn word vectors. The GitHub repository[5] provides an implementation of the GloVe model for learning word representations (word vectors or embeddings). We used the default embedding provided by the python API [6].

Tokenization. They used Stanford tokenizer [7], a form of BPE tokenization scheme (Sennrich et al., 2016; Gage, 1994) that constructs unigram counts from a corpus, and optionally thresholds the resulting vocabulary based on total vocabulary size (2.2M [8] most frequent words in the case of GloVe embeddings) or minimum frequency count (Pennington et al., 2014).

F.4 Llama3

LLaMA 3 (Large Language Model Meta AI) is a decoder-only large language model (met, 2024), which Grouped Query Attention (GQA) that allows the model to effectively handle longer contexts. The model is available in various sizes, including 1B, 3, and 70B parameters. The larger models exhibit significantly improved capabilities in reasoning and complex language tasks.

Training Dataset. LLaMA 3 is pretrained on an extensive dataset, including over 15 trillion tokens sourced from diverse text corpora, such as books, articles, and websites. This large-scale training ensures comprehensive language understanding across different domains .

G Case Sensitivity Analysis

The T5 model's vocabulary preserves case information, enabling us to examine how formed concepts align with capitalization differences. We identified 4,328 tokens with varying case appearances (total of 8,887 tokens). We found that for highly granular concepts (k=6), 80% of these tokens belong to the same community. This ratio increases to 85% for k=25 before plateauing, suggesting that case variations generally do not drastically alter the semantic grouping of tokens. This finding supports the notion that the model learns to associate words with their meanings regardless of capitalization, particularly for more abstract or broader concepts (larger k values). However, the initial increase in alignment ratio with increasing k implies that case sensitivity might still play a minor role in differentiating highly specific or nuanced concepts.

H ALBERT Human-Model Aligment

H.1 Names and Locations

Figure 2 visualizes the major identified communities of locations and human names. Generally, the formed clusters associate with specific regions or cultures and contain both location and personal names. Within these, even more granular subclusters emerge, characterized by distinct communities of location and personal names. Interestingly, we observed a degree of geographic ordering within the identified communities. As illustrated in Figure 2, there appears to be a general trend from east to west as we move across the communities from bottom-left to top-right. The leftmost communities are predominantly associated with Japanese locations and names, while those on the rightmost side are primarily linked to Europe and the United States. This suggests that the model's internal representations in the input embedding layer may inherently capture geographical relationships.

To mitigate the subjectivity risk of assessing the semantic structure, we further used external datasets in our evaluations. For Names, we used name-dataset (Remy, 2021) which consists of a comprehensive set of names (730K first names and 983K last names), their associated genders, and their popularity rank for each country. For locations, we used the country-state-city database (Gada, 2018) which contains information on all countries, 5K+ states, and 150K+ cities. Table 9 shows the high-level communities that our approach detected. Most of the high-level communities are a mix of names/locations associated with certain geographical/cultural regions. Within these clusters, names and locations form distinct subcommunities which we discuss in more detail in the following subsections.

[5] https://github.com/stanfordnlp/GloVe

[6] Common Crawl (840B tokens, 2.2M vocab, cased, 300d vectors, 2.03 GB download), and a context window size of 10.

[7] https://nlp.stanford.edu/software/tokenizer.shtml

[8] 2196016 cased tokens

Note that since there's no one-to-one mapping between names/locations in LLM vocabulary and external datasets (e.g., a name may appear in multiple countries, the external dataset is also a superset), recall is less relevant. We prioritize precision to evaluate accuracy, as it better reflects our ability to identify correct matches.

Names. To determine whether the sub-clusters are associated with names, we pre-filtered the clusters that at least 70% of their tokens are in the top 1000 names (of any country), with gender confidence of above 0.8. We primarily use gender confidence to distinguish between first-name from last-name clusters. Then we cross-referenced all identified name communities for all countries in the dataset and assigned the country name with the highest score [9] as the cluster names.

Table 8 shows the precision score of the identified granular communities. The first column shows the ratio of the community members that are indeed human names (overall precision), while the second column shows the ratio with respect to specific countries. It should be noted that the country-wise scores of the countries with similar cultures and languages were similar. The reported precision score shows a high degree of categorization based on the country/culture of origin. [10]

The formation of these clusters in the input embedding space, particularly those containing ethnic minority names, presents an opportunity for token engineering to mitigate potential ethnicity biases. (As our focus here is on interpretability and conceptual groupings and alignment, we provide an example of such a token engineering approach in Appendix 6 for interested readers.)

Locations. It should be noted that, although the dataset is majorly comprehensive, the LLM token space associates only one token to each location, leading to an artificial decrease in precision for multi-token location names. Despite these limitations, we were able to identify communities within the input embedding space that are associated not only with the location category but also with spe-

[9] For some clusters we picked the second highest country name if the scores were similar. Due to the cosmopolitan nature of countries like the USA, They tend result in a high score across the board.

[10] It should be noted that we only included clusters with sizes larger than 10 and country-wise precision of more than 0.5 due to space limitation. The list of identified cluster names goes far beyond the aforementioned table. Clusters such as character names from books, mythology, and car brand names were also identified which were not included due to space limitations.

Table 7: Precision of the largest identified name and location communities with respect to name and location databases. Note that these are at a higher level in the cluster hierarchies. Table 8 shows the identified granular sub-clusters and their associated precision.

Category	Precision	Support	Note
US/UK/AUS/NZ	0.882	1011	Human & Location
Male	0.854	946	Human Names
Female	0.866	552	Human Names
West-Asia	0.684	390	Human & Location
Hispanic/Latino	0.685	282	Human & Location
US	0.720	267	Location Names
Europe	0.739	215	Human & Location
East-Asia	0.741	178	Human & Location

Table 8: Precision of communities based on the identified categories with respect to name-database. First and last indicates category of first and last names.

Country	Overall Precision	Country Precision	Support	Note
USA/UK	0.857	0.725	211	First
UK/Canada	0.886	0.698	116	First
Saudi/Arabic	0.82	0.76	94	First
Spain/Mexico	0.977	0.78	87	First
USA/Hebrew	0.835	0.568	81	First
Italy/Swiss	0.887	0.625	80	First
Belgium/France	0.928	0.789	76	First
German/Sweden	1.0	0.709	55	First
German/Austria	0.962	0.717	53	First
India	0.82	0.56	39	First
Russian	0.896	0.724	29	First
France	0.9	0.737	76	Last
Mexico	0.951	0.855	83	Last
China	0.9	0.9	30	Last
Denmark	0.88	0.64	25	Last
Germany	0.95	0.95	20	Last
Japan	0.928	0.857	14	Last

Table 9: Precision of communities based on the identified categories with respect to the location database.

Country-Region	Precision	Support
United States	0.80	240
Germany	0.412	80
France	0.409	66
Africa	0.690	55
India	0.580	50
Italy	0.590	44
Mexico	0.424	33
Spain	0.592	27
China	0.500	20
Japan	0.736	19
Philippines	0.460	15
Pakistan	0.461	13
Netherlands	0.636	11
North-Africa	0.800	10

cific regions/countries. The precision numbers in Table 9 suggest that the model groups the locations with respect to the borders of the countries, which, to a certain degree, implies a subjective perception of geographical knowledge aligned with the external world [11], wherein it approximates borders and associates nationalities with specific clusters/communities. It should be noted that, due to space constraints, we only present the communities containing more than 10 entities [12]. Note that dealing with multi-token location names is more challenging. For instance, "Carolina" was correctly clustered within the United States community by the LLM, while our reference dataset misclassified it as a city in Brazil. Additionally, not all location names with English spellings are included in the dataset; for example, 'Wurttemberg' (or 'Nuremberg'), a region within Germany, is absent, leading to an artificial decrease in precision.

H.2 Social Structures

Our methodology identifies a cluster of 903 members with the theme of social structures. In order to have a reference dataset, we annotated the dataset using a combination of GPT-4 and human annotators [13]. Given the potential human subjectivity in analyzing these concepts, we ask GPT-4 to identify the theme of each category in the datasets. (as shown in Table 10) for mitigating such uncertainity. In order to reduce subjectivity (we formed the class names based on the GPT-4 recommendation. Then, we asked GPT-4 and human annotator to classify each word with respect to given class names (we added another class named "Other" to avoid forcing the annotators to miscalassify).

Then, we calculated the precision score of the identified clusters with respect to our annotated dataset. We see the precision scores shown in Table 10 as evidence that the model forms an idea on different aspects of social structure in its semantic memory. When it comes to more granular clusters (k=25), the sub-clusters are mostly word-forms or

[11] We refer the definition of the mental map to American Psychology Association dictionary. It is defined as "a mental representation of the world or some part of it based on subjective perceptions rather than objective geographical knowledge.

[12] The complete set of clusters is included as the supplementary material.

[13] A human annotator was used alongside GPT-4 to correct GPT-4 misclassifications. Label correction by a human happens faster than the label suggestion task. We estimated our approach is more time/cost-effective while resulting in the same quality annotations.

Table 10: The precision score of community members belonging to the identified categories.

Category	Precision	Support
Religious (Christianity)	0.818	258
Military and Law Enforcement	0.842	133
Administration and business	0.788	129
Political Ideologies/Movements	0.648	125
Monarchy and Aristocracy	0.64	107
Legislature and Election	0.736	80

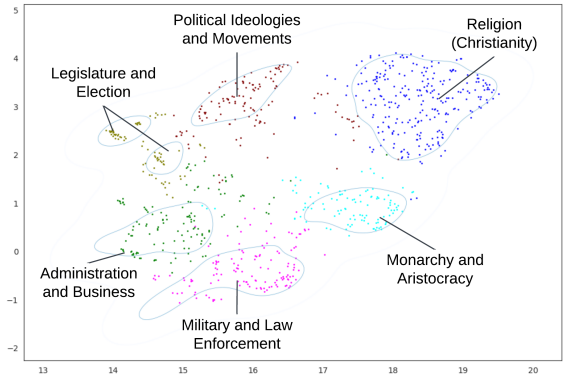

Figure 4: Visualization of the social structure cluster and its associated identified sub-clusters.

semantically similar words.

Intriguingly, the vocabulary model grouped words like "God" and "divinity" within the same community as concepts and structures associated with Christianity. Conversely, terms like "Islam," "Judaism," and "Talmud" formed a distinct cluster. This finding warrants further investigation to determine whether it reflects potential biases within the underlying semantic memory.

I Human-LM Alignments For Glove, T5, Llama

We performed our concept extraction algorithm on Glove, Albert, and T5 models. We observed that conceptual groupings happens in all the models, however, the quality of the formed concepts is better in Albert model. Glove embeddings contain more than 200K tokens, in order to enable apple-to-apple comparison with Albert, we found the intersection of these tokens with the huggingface Albert-base-v2 token set, and, applied our concept extraction algorithm. We the formed clusters have weak correlation with the concepts formed in Albert model [14] (they similarities are stronger for

[14] if we assume Glove-Albert mapping exists between two clusters if more than half of their members are equivalent,

Table 11: Location communities founds in T5 tokens. Note that most of the communities are member of 0_0_5, and 0_0_13 super communities. SOAM stands for South America.

	Precision	**Support**	**Cluster Name**
USA	0.843	159	0_0_5_4
Britain/Ireland	0.843	118	0_0_5_5
Africa/SOAM	0.672	64	0_0_13_2
Germany	0.619	21	0_0_13_0_0
Australia	0.842	19	0_0_5_10
France	0.867	15	0_0_13_0_2
Canada	1.000	13	0_0_5_14
Balkan	0.833	12	0_0_13_1_1
Indochina	0.818	11	0_0_13_3_1
Benelux	0.700	10	0_0_13_0_3
Canada	0.778	9	0_0_5_24
India	0.750	8	0_0_13_4_0
Romania	0.714	7	0_0_22_1_4
Central Europe	0.800	5	0_0_13_0_5
Israel/Palestine	0.800	5	0_0_13_5_3
Arab Countries	0.800	5	0_0_13_5_5
Nordic	0.750	4	0_0_13_1_6
Baltic	1.000	4	0_0_13_1_8

concrete names/entities).

Tables 11 and 12 show the T5 location and name communities detected by our algorithm. High precision numbers for these cluster indicate clear conceptual groupings. However,since the pretrained HuggingFace T5 uses cased token set for the pretraining, the number of tokens in associated with location and names are much smaller than Albert and Glove. Table 13 shows the Llama3 name communities detected by our algorithm. Tables 14 and 15 show the GloVe location and name communities detected by our algorithm. Although, it shows the categories are formed in GloVe embeddings as well, the numbers suggests the quality of the formed categories have less quality than both Albert and T5 counterpart.

Table 12: Name communities founds in T5 tokens. Note that most of the communities are member of 0_0_5 super community.

Gender	Overall Precision	Detected Country	Country Precision	Support	Cluster Name
Male	0.942	United States	0.962	209	0_0_5_0
Female	0.888	United States	0.858	162	0_0_5_3
Last Name	0.904	United States	0.914	198	0_0_5_1
Male	0.837	France	0.612	49	0_0_5_7
Male	0.909	Peru	0.758	33	0_0_5_8
Mix	0.960	Germany	0.84	25	0_0_5_9
Male	0.846	Russian/Italy	0.615	13	0_0_5_15
Politicians	1.0	N/A	N/A	14	0_0_5_13

%69 percent of the Glove clusters have a corresponding Albert cluster for K=6.

J Extracted Concept Hierarchies

Figure 5 shows the overall structure of hierarchical communities extracted by our proposed method. The cluster names were suggested by GPT-4 and corrected by a human supervisor. The green blocks are the ones that are discussed in this paper.

Table 13: Examples of Name Entity clusters with size greater than 25 tokens found in Llama3 input embedding size. Note that we only evaluate against external datasets with English named entities.

Type	Precision	Support	Note	Cluster names
female	0.757	393	US/UK	0_0_0_0_0_0
male	0.811	291	US/UK	0_0_0_0_0_2
male	0.6639	171	US/UK	0_0_0_0_0_3
male	0.495	105	Saudi/UAE	0_0_0_0_0_5
male	0.6875	53	US/South Africa	0_0_0_0_0_11
male	0.5588	46	UK/Canada	0_0_0_0_0_12
female	0.571	41	US/UK	0_0_0_0_0_13
male	0.551	37	Mexico	0_0_0_0_0_14
male	0.619	30	Netherlands/US	0_0_0_0_0_16
male	0.666	30	US/UK	0_0_0_0_0_17
male	0.653	27	Biblical	0_0_0_0_0_18
last names	0.724	340	US/UK	0_0_0_0_0_1
last names	0.528	135	Canada/US	0_0_0_0_0_4
last names	0.509	65	UK/US	0_0_0_0_0_8
last names	0.717	39	Mexico/Chile	0_0_0_0_6_0
locations	0.75	485	Mostly American	0_0_0_8_0

Table 14: Location communities founds in the subset Glove tokens that exists in Albert Vocab.

Country	Precision	Support	Cluster Names
US	0.674	522	0_4
UK	0.626	174	0_8
Europe	0.513	39	0_3_8
China	0.625	32	0_3_10
Italy	0.720	25	0_3_14
Philippines	0.556	18	0_3_15
Spain	0.722	18	0_3_15
Japan	0.688	16	0_3_16
France	0.857	14	0_3_9_0
Africa	0.750	12	0_3_3_1
Indochina	0.600	10	0_3_11_1
Netherlands	0.833	6	0_3_17_0

Table 15: Name communities founds in the subset Glove tokens that exists in Albert Vocab.

Country	Overal Precision	Country Precision	Gender	Support	Cluster Name
USA	0.835	0.555	female	575	0_2_1
UK	0.695	0.641	male	223	0_2_0_0
USA	0.737	0.337	male	95	0_2_6
Italy	0.942	0.692	male	52	0_2_4_1
Mexico/Colombia	0.889	0.711	male	45	0_2_4_2
Mexico/Peru	0.844	0.6	male	45	0_2_4_3
Austria	0.897	0.793	male	29	0_2_3_3
US/Nigeria	0.821	0.429	male	28	0_2_0_1_2
Russia	0.926	0.481	male	27	0_2_3_4
Saudi Arabia	0.8	0.84	male	25	0_1_6_1
Switzerland/Belgium	1	0.846	male	13	0_2_3_0_0
UAE	0.727	0.727	male	11	0_1_6_3
Saudi-Arabia	0.8	0.7	male	10	0_1_6_0_1
Germany	0.917	0.75	no-gender	12	0_2_3_2_2
UK/Canada	0.917	0.33	no-gender	12	0_2_0_1_5
UAE	0.6	0.467	no-gender	15	0_1_6_2

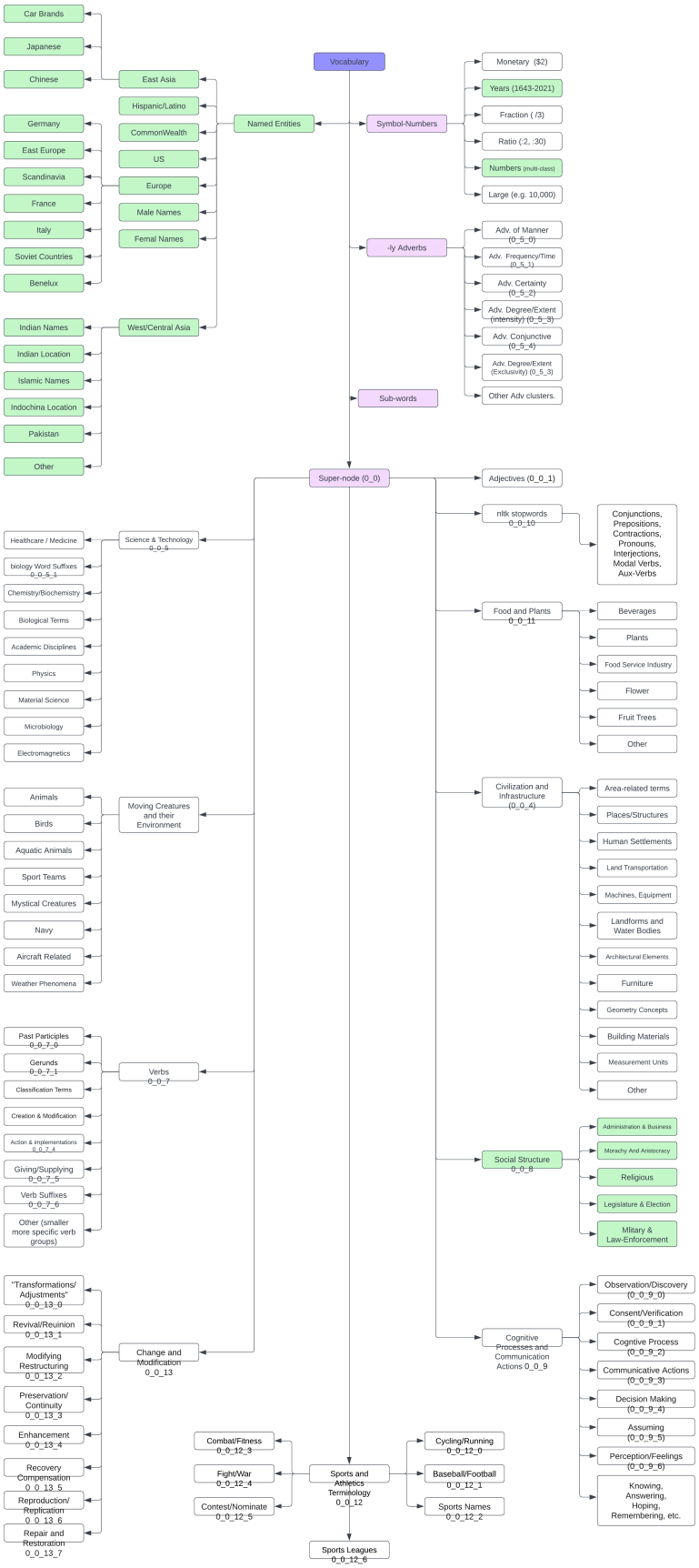

Figure 5: Visualization of the hierarchical Communities from Albert. The green blocks show the clusters that being evaluated and discussed in this paper.