Diving Under the Hood: Exploring LLM Conceptual Understanding Through Latent Embeddings

Stanford CS224N {Custom} Project

Kelvin Nguyen
Department of Computer Science
Stanford University
kelvinkn@stanford.edu

Abstract

Concepts form the mental representations of words, and are used for high-level thinking, reasoning, and decision making, representing a core difference between humans and contemporary large language models (LLMs), which are trained at the token-level. While work has been conducted to evaluate LLMs' conceptual understanding and to endow them with conceptual awareness, this work has mainly worked on prompting models, and not their latent embedding spaces. Yet, concepts have been encoded in embeddings for years: early embedding methods such as word2vec encoded certain conceptual relationships (ex: hypernym-hyponym), as apparent in the parallelogram rule.

Therefore, I explore whether a contemporary model can identify and extract different aspects of in-context, concepts from their embeddings. To do that, I take very polysemous words, which contain high contextual diversity and therefore aspects, cluster different sentences containing each word, and used an LLM to describe embeddings. Results show that the LLM can encode context-dependent aspects of concepts in its embeddings, can determine the highlighted aspect of a word in context, and can understand if a cluster of sentences is conceptually cohesive.

This analysis suggests that LLMs can generate accurate descriptions of their conceptual aspects in context, and present an evaluation method to verify their correctness and another method to prove a model's understanding of cohesive clusters. I also confirm that there is no simple metric to identify noisy and incohesive clusters, but do find a direct correlation between the number of clusters for a word and their frequency.

1 Key Information to include

- Mentor: Chen Shani cshani@stanford.edu
- External Collaborators (if you have any): None
- Sharing project: No

2 Introduction

Concepts play an important part in human speech and reasoning. They provide the sustenance from which words aim to convey, and transcend the linguistical differences between languages. They are, essentially, the backbone for human thought [1]. When communicating, humans consider both the concepts and the tokens necessary to convey them, which can lead to the same concepts having various word representations.

On the other hand, present-day Large Language Models (LLMs) are currently trained and operate only on the token level. This can lead to an unnaturally limited range of speech, as models operate

only the probabilities of a word, and not their underlying concepts. Token-centric training can also lead to a phenomenon known as *surface form competition*, where the probability for a given concept is spread among its various word forms. [2], artificially hurting its chances of correctly being output. Work has been conducted in creating concept-aware LLMs, and it has been demonstrated that LLMs have some conceptual understanding when prompted directly with tests of logic, and it has been determined that some models [3] do possess conceptual knowledge. In addition, conceptual relationships (such as the hypernym-hyponym relationship) have historically been observed to be mapped in early vector embeddings such as word2vec [4], specifically as seen in the *parallelogram rule*. With the transformative potential of endowing LLMs with concepts, it is interesting to see if contemporary models' embeddings already do encode awareness of conceptual relationships, and more excitingly, *does a model's latent embedding space map different aspects of the same concept given varying contexts?*

As such, I explore two research questions:

1. Does a model's latent space encode the highlighted aspect(s) of a word given a context?
2. Given the noise present in the embedding space, is there a metric to easily detect or fix clusters of embeddings that are conceptually incohesive?

I specifically examine very polysemous words, as these correlate to higher word frequency [5], and higher word frequency leads to more contextual diversity [6], which would give us more aspects to examine.

I present my findings that:

- We can find aspects of the same concept in a model's contexualized embedding space, and models are able to recognize and can extract a specific conceptual aspect of words in context, and also understand when a cluster of sentences are not conceptually cohesive.
- There is a strong, significant correlation between the number of senses for a word, and the number of embedding clusters generated. There is a direct correlation between the number of clusters for a word and the frequency of the word in text.
- There is no observed simple metric from the data to determine or remedy the incohesivity of a cluster.

3 Related Work

Conceptual awareness in models is nuanced and desireable goal with the potential to make LLMs more robust and flexible to unimportant semantic variations. Current literature points to the notion that models possess varying levels of conceptual understanding, and it has been determined possible to train models on subsets of tokens which represent certain concepts, and not singular tokens, both during the pretraining stage and, especially excitingly, the finetuning stage [7]. Conceptual understanding has also been pursued in the field of visual question answering (VQA), with work demonstrating that labeling images with their high level concepts improves image captioning and sentence question answering [8]. The idea of focusing on concepts to handle semantic variation has also been explored in computer vision, with recent work showing a marked improvement when training vision models using visual concepts to help them map these to semantic concepts [9]. In addition, work has been conducted to gauge LLM's self-awareness of their capabilities and their social intelligence [10], both of which contain very high-level, in-context conceptual knowledge. In addition, researchers have introduced into some models a system to map input text to hidden neurons that are trained to identify certain high-level human concepts, such as food and service quality, to create bottlenecks that can then be used to perform downstream tasks such as sentiment analysis, with the goal of improving model interpretability [11]. However, these works focus on determining conceptual awareness or training concepts into models on the word-level, with not much attention on the latent embedding spaces that could encode meaningful conceptual knowledge.

4 Approach

Task: Given a set χ of polysemous, multifaceted words (as defined by WordNet's number of senses for them), for a given word $x \in \chi$, I want to find all the different aspects of the word that can be

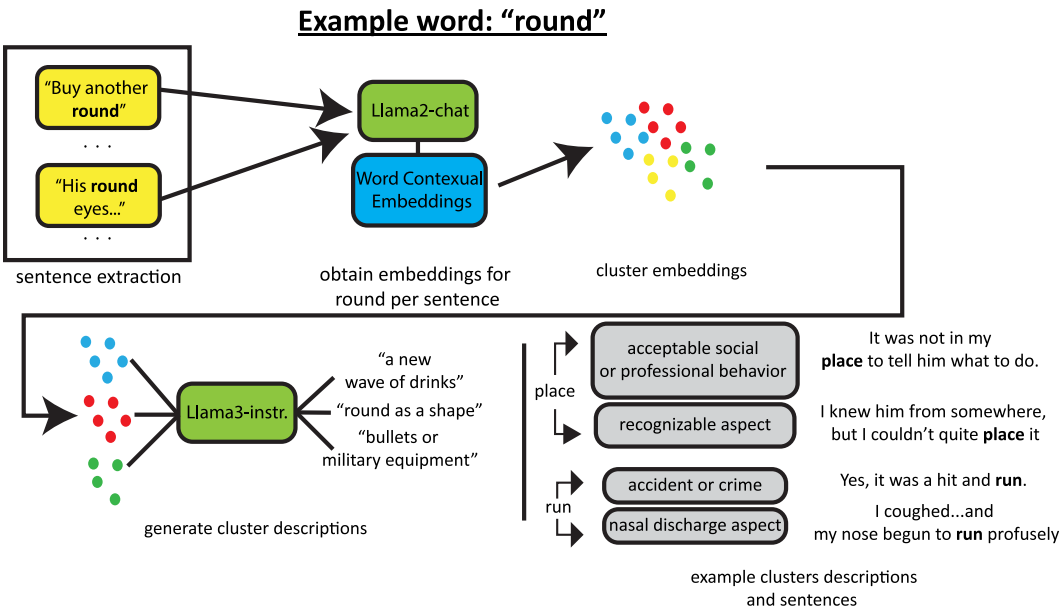

Figure 1: Approach Architecture.

represented and encoded in the model's embeddings. I need to generate the contexualized embeddings for the word x, cluster them, and have the model describe the aspect of the word being represented in these clusters. I then want to evaluate the quality of these descriptions. The complete architecture is seen in Figure 1, and each section below corresponds to a step in the figure.

An aspect of a word can include the various contexts in which it is used/interacted with and the various scenarios that it inhabits, and also includes its formally defined senses. For example, the word "coffee" can be used to describe a "coffee shop", be modified by a color such as "black coffee," or even be a color itself, such as "coffee colored." Of course, a very polysemous word is also a very multi-aspected one, as in the word "run". The distinction between whether a model describes an aspect of a word as being an entirely distinct sense or simply a modification based on its context is not significant; however, the model should recognize how a word affects and is affected by its context.

4.1 Extracting Sentences

Correlating to the first step in Figure 1, I start by extracting every sentence that contains the given word x from our dataset. I also included instances of sentences that contained the plural form of the word (i.e. suffixed with -s), if such a form is a valid word. Preliminary tests determined that the cosine similarity between the singular and plural form of a noun was high (across a test of six words, averaged .6-.7), and so should not significantly affect the final result. Due to ambiguity over whether other forms of a word represent an aspect of the word or a completely different word (ex: open vs opening), no other forms were included. For example, using the word "round", I would take all sentences with the word "round" or "rounds", but not "rounding", or "around".

4.2 Generating Embeddings

As per the second step of Figure 1, I feed all of these sentences with the target word into a Llama2 [12] chat model, and extract the contextualized embedding for the word. For words that are tokenized into multiple tokens, the resulting embeddings are averaged together.

4.3 Clustering Embeddings

For the third step, the embeddings were then clustered utilizing nVIDIA's cuML library's implementation of HDBSCAN, the algorithm of which is described in [13]. This algorithm finds areas of high and low density. This algorithm was chosen because concepts are inherently hierarchical, and so this would represent their relationships more accurately. This algorithm was also chosen over K-Means

and Agglomerative clustering because these require a k cluster hyperparameter, which was inherently indeterminate as I do not have, nor can expect, a fixed set of aspects to find. I chose a small minimum sample size and small minimum cluster size relative to the size of the dataset, as these would produce the most finegrained clusters that would better assist the model in extracting a singular aspect for this cluster (exact numbers in 5.3).

4.4 Generating Cluster Descriptions

Once the clusters have been determined, each embedding in a cluster still corresponds to its original sentence. I then pass all the sentences in a given cluster into a Llama3 [14] instruct model, and prompt it to describe the aspect of the target word that is being represented in the cluster as well as to determine if the cluster is cohesive. All clusters were clipped to 240 sentences, an empirically-derived value that alleviated massive inputs that would exceed the model's context length and caused indeterminate behavior.[1]

5 Experiments

5.1 Data

The corpus I used consisted of the BookCorpus [15] dataset, a collection of about 7,000 self-published works of literature from the website SmashWords. To find very polysemous, multifaceted words to evaluate, 65 of NLTK's WordNet [16] 100 most polysemous words were randomly chosen. Random selection ensured that there was a slight variation in the polysemity of the words, in order to see its effects on the clusters, and 65 was simply chosen due to the smaller scale scope of the project.

5.2 Evaluation method

Because I am working with concepts, it is difficult to establish a perfectly quantitative metric. Still, to automate evaluation, I passed all the sentences associated with a cluster of embeddings into a Llama3 8B instruct model and prompted it: **You are given a list of sentences, all containing the word <word>. State in up to five words what aspect of <word> is being represented in these sentences in one line in the form: RESPONSE: <aspect>. Also state in only one word if there is only one aspect of the word represented in the list of sentences in the format: ONE ASPECT: <TRUE OR FALSE>. "** Clusters that the model labeled as having one aspect, specifically containing the string "ONE ASPECT: TRUE", were marked as cohesive.

Evaluation Task 1 To verify the accurateness of these generated descriptions as well as the correctness of the model's determination of cohesive versus non-cohesive clusters, I created my own evaluation method. In this method, I randomly sampled a set of k_{words} words, and within them randomly sampled k_{sents} sentences from a random $k_{clusters}$ clusters the model labeled as cohesive. For each sentence, the $k_{cluster}$ descriptions of the clusters were presented to a human evaluator alongside the sentence, and they would select either one description if they confidently could tell which cluster the sentence belonged to, multiple if it was ambigious (such as due to duplicate titles), or none if they believed none of the descriptions were accurate. See Figure 2 for an example. A high number of correct, single responses indicates strong cohesivity within clusters and distinctiveness between them, and a high number of times the user correctly chose the cluster but with multiple choices chosen indicates high cohesivity but low distinctiveness.

Evaluation Task 2 Additionally, to conversely verify that clusters labeled as having various aspects or as being incohesive were correct, I also randomly sampled a set of words, clusters, and numbers similar to the above evaluation metric. I then took the k_{sents} random sentences for each cluster, and displayed them alongside the description for the associated cluster, and then prompted the evaluator to label the cluster as cohesive or incohesive. See Figure 3 for an example.

[1] In hindsight, we should have randomly sampled sentences rather than purely choosing the first k sentences. The value 240 was also chosen through limited experimenting, but no rigorous testing was done to verify I had the optimal value

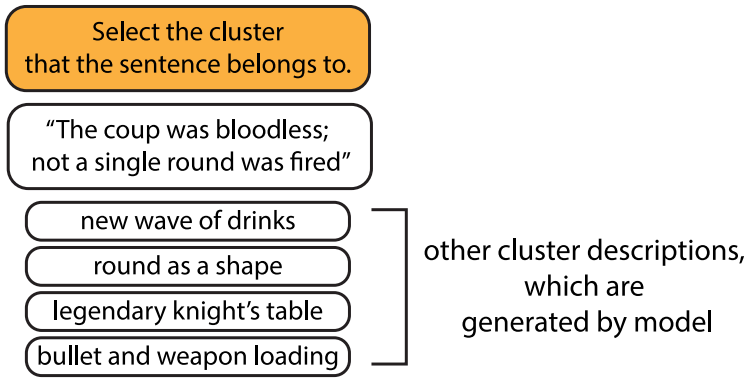

Figure 2: Example of Evaluation Question for Verifying Cluster Descriptions (Task 1)

The evaluator looks at sentences from a cohesive cluster, and chooses the right cluster description to verify that the descriptions match the sentences .

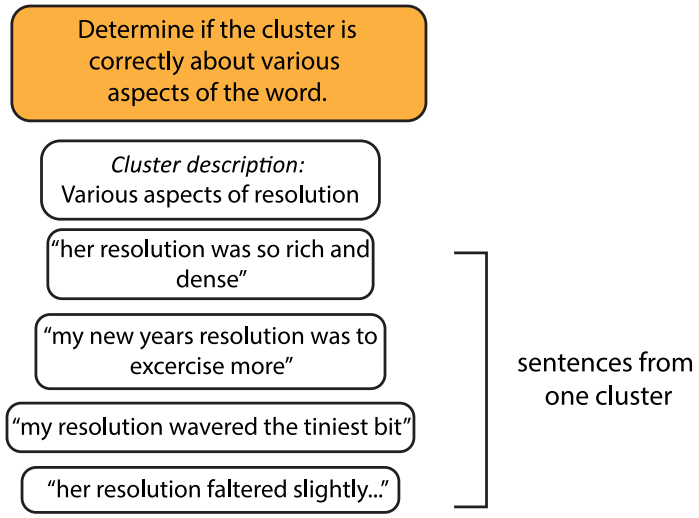

Figure 3: Example of Evaluation Question for Verifying Model's Incohesive Cluster Label (Task 2)

The evaluator looks at a incohesive (ie having multiple aspects) cluster description and it's sentences, and verifies that the cluster is indeed incohesive.

For both metrics, scores were calculated based on a simple ratio of correct answers to total answers.

5.3 Experimental details

For the embeddings, a Huggingface Llama2-7b-Chat model [12] was used. To evaluate the clusters, a HuggingFace Llama3-8B-Instruct[2] model was used, with a temperature of .2 and a top probability of .9. For clustering, nVIDIA's cuML's HDBSCAN was used with the minimum number of samples was set to 8, the minimum cluster size was set to 30, and the distance metric set to euclidean. For plotting the embedding vectors, cuML's UMAP was used to reduce the dimensionality to 2, using a cosine distance metric, 40 neighbors, 240 epochs, and a random state of 2024.

For both human evaluation tasks, a sample of 10 words, each with 5 clusters, and for each cluster 5 sentences were used. For the purposes of this project, I was the human evaluator.

[2]I extracted Llama2 embeddings before I learned that Llama3 had been released just a month prior. Given that Llama3's instruct had better performance, I reasoned it would do better in the cluster description generation task.

5.4 Results and Analysis

5.4.1 Model Performance on Human Evaluation

I present the results for our previously-defined evaluation task here, summarized in Table 1.

Table 1: Cluster Cohesivity Evaluation Results

Choices	**Accuracy**
Correct Cluster, one cluster chosen	56%
Correct Cluster, multiple clusters chosen	18%
Total Accuracy	**74.8**%

**Note: Guessing baseline for one cluster is 20%*

As described before in 5.2, I wanted to verify that the model can correctly produce accurate cluster descriptions, as well as correctly identify when a cluster is cohesive or not. My previously created and defined evaluation method aims to do that.

Based on the human evaluator, the model was able to classify a sentence (using the task from Figure 2 into solely the correct cluster 56% (noting that randomly guessing has a baseline accuracy of 20%) of the time. However, there were some duplicate and slightly overlapping titles, so 18% of the time the cluster assigned by the model was one of the multiple chosen by the evaluator, meaning that the model was able to **generate correct descriptions for the clusteres 74.8%** of the time. **This suggests that the model is acceptably capable of extracting the nuanced aspects of certain concepts, and can also identify that these extracted aspects indeed represent a cluster of sentences.**

Some examples from this task are given above in Table 2. In the first row, the reader can see an example where the model and user agree on the description. In the second row, the model generated a very close but not quite correct description, and the user chose instead another description that the model generated for a different cluster. In, the third row, the second user-chosen description is from a different cluster. This illustrates how some of the descriptions the model generated for other clusters can sometimes overlap with multiple clusters, leading to ambiguity, hence why the evaluator chose two responses. Some of the descriptions also highlight the dynamic nature of these aspects of concepts, and the LM's ability to identify them: for instance, the word "wing" was used as a name of both a geological place and a person's name, and the clustering algorithm was able to separate these instances and the model was able to detect this difference, despite the fact that using "wing" as a name is not common nor most likely pretrained for.

In addition, the model, based on a human evaluator (which for the purposes of the project was me), **correctly classified clusters as being incohesive or having various aspects of a word represented 72.5%** of the time, using the evaluation metric (ie, being presented with sentences from a incohesive cluster, and having the evaluator verify the sentences do indeed show multiple aspects of the same word) of which is described above, and again an example is given in Figure 3.

5.4.2 Cluster Characteristics and Noise

Inital Attempts to Remedy Incohesive Clusters The model would have a difficult time describing certain clusters, often outputting "various aspects of <word> represented", or would produce duplicate cluster descriptions. To remediate this, reclustering these groupings was attempted. For indeterminate descriptions, I attempted to run HDBSCAN on just the indeterminate clusters, and for the duplicate clusters, I pooled all the embeddings from these clusters and reran HDBSCAN. However, it was

Table 2: Example of User Responses to Evaluation Task

word	**sentence**	**model-assigned description**	**user chosen description**
medium	"My steak's medium rare."	"cooking temperature"	"cooking temperature"
issue	"that's not the issue'	"disagreement or distraction"	"problematic situation"
medium	"no', said the medium"	"communication aspect"	["communication aspect", "aspect of medium as a person"]

**Note: All descriptions are model generated.*

observed that there was only a marginal improvement, usually one or two better-described clusters. As such, as per research question 2, I was also interested to see if there was any metric or variable that could be used to easily identify and recluster incohesive clusters.

This was especially important because our clustering algorithm labeled a significant amount of points as noise, ranging from 30% to 80%[3]. Empirically, it has been previously shown that polysemity correlates positively to word frequency [5], and my clusters include senses for words and the in-context ways those senses are used, so if the number of clusters was not somehow correlated to word frequency, then these variables could be responsible for incohesive clusters. To analyze this, I present my resulting data on the relationships between number of senses, clusters, frequency of a word, and also other related variables.

Overall, there was no significant relationship between the cluster size and number of clusters), with a correlation score of -.24. There was also a significant overlap between the length of clusters that were determined to be incoherent and those determined to be coherent, with the median cluster length of 59 and 57 respectively. Words most frequently had below 50 clusters.

The cluster sizes tended to be around 50-100 sentences. The relatively high number of clusters also signifies that the model already encodes a large range of aspects of concepts. This points to the fact that some concepts are highly modified by their context, and also that the model is able to detect variations in it.

Below in Table 3, I also present the correlation scores between the number of clusters, frequency of a word, the number of a word's senses, and their median cluster size.

X var	Y var	Correlation	Significance
# clusters	frequency of word	**.94**	1.8e-21
# senses	# clusters	**.69**	1.9e-07
# senses	frequency of word	**.62**	5.4e-06
# clusters	median cluster size	-.37	.002
# senses	median cluster size	-.24	.02

Table 3: Relationships between number of clusters, number of senses, frequency of word, and median cluster size.

As shown in Table 3 and displayed in Figure 4, these demonstrate that the number of senses of a word can be used as a rough indicator of how many clusters one should expect. Yet, the fact that I show high correlation between number of clusters and word frequency, coupled with the demonstrated constant difference between the number of cohesive and incohesive clusters 5.4.2 across all sense levels, **shows that we cannot look at number of clusters and word frequency to predict and remedy incohesive clusters**. Instead, one can only merely look at the number of senses for a word and the frequency of the word to understand if enough clusters have been identified.

As for the other potential variables that could explain cohesivity, the number of sentences in each cluster did not appear to have an consistent effect on the cohesivity of the cluster, with an exception of extremely large clusters (>4000 sentences), with the median absolute difference between the number of cohesive clusters and incohesive clusters being 1; however, with a high standard deviation of 67. Therefore, I find that there is no easy metric to predict and fix incohesive clusters.

However, these correlations do demonstrate our conclusion for research question 1. Specifically, it is known that the word frequency positively correlates to the number of senses a word has, which I demonstrate. We know that the number of senses correlates, but not completely, to the number of clusters, which are composed of our embeddings. That means I can safely assume that the other part of the correlation is contributed by the aspects of the word (meaning, the same sense used in different contexts), which we know as I have established that the aspects are represented in the cluster descriptions. **Therefore, this demonstrates again that the senses and contexual aspects of a word directly correlate to the clusters, or embeddings, of the model.** This table also serves as a sanity check to verify that our data is consistent with previously-observed semantic patterns.

[3]Density-based clustering algorithms allow for points to be classified as noise, i.e. not belonging to any cluster.

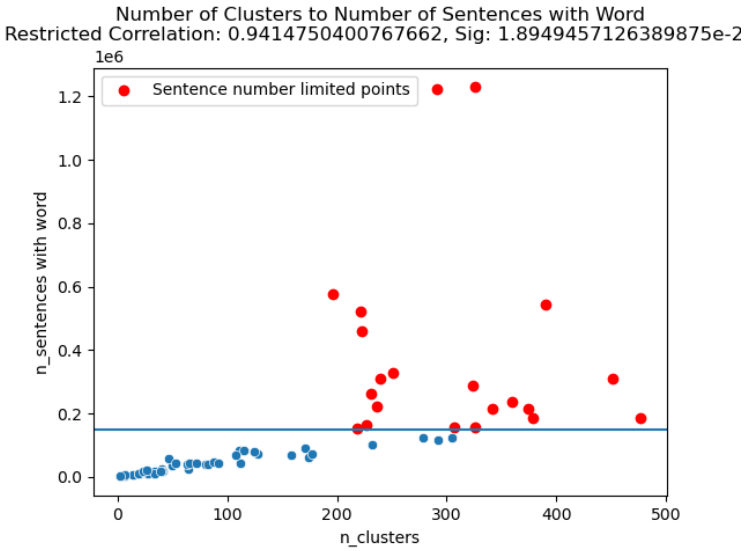

Figure 4: Number of Clusters per Word to its Frequency

**Note: Due to performance limitations, words were accidentally limited to 150,000 sentences. These points are depicted in red. All correlation values were calculated on the non-limited points.*

6 Conclusion

Based on the experiments, our chosen **LLM cleary encodes certain conceptual knowledge in its latent embedding spaces, even though it was never explicitly optimized to do so**. The LM can also identify conceptual aspects that are present in the text that was clustered using its own embeddings, signifying a direct relationship between the conceptual knowledge in its latent spaces and in its output. I present a simple and robust model-agnostic evaluation method to verify the accuracy of a model's description of clusters, and use this to verify that the LLMs have the ability to automate the extraction of concepts from context and to determine the conceptual cohesivity of text.

I also discover that there is no easy metric from our data to determine if a cluster is incohesive or a way to easily remedy it. However, I show that the direct relationship between the number of clusters and the frequency of a word can be used to verify that one has gathered enough clusters for a certain word, and this relationship and our data is supported by previously-observed semantic patterns.

Of course, it would be interesting to see if our results hold up for other datasets as well, given that BookCorpus is a relatively small and niche corpus. The use of external human evaluators to run our task on larger selections of data would also help to verify our findings. Duplication of model labels was also not addressed; however, one could possibly use the fact that models can somewhat understand concepts to prompt the model to identify and recluster clusters that have near-identical labels. Finally, a considerable amount of data is removed due to noise. so testing our findings on other datasets is important. Additionally, I only used Llama models in our implementation, and I used a different Llama version to extract the embeddings and to generate cluster descriptions, so verifying our results with other models would prove informative.

7 Ethics Statement

Because the main corpus of the project is a mainly fiction literature, some concepts analyzed may not be factually correct, and can contain harmful language. Also, since I directly cluster word embeddings, discriminatory concepts encoded in these latent spaces can accidentally be clustered together (such as gendered language and certain careers having a low vector distance). To mitigate this, I would include an option in my human evaluation program to flag problematic clusters and sentences, and also label which sentences come from fictional sources.

References

[1] Gregory L. Murphy. *The big book of concepts*. Bradford Book, 2004.

[2] Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[3] Chen Shani, Jilles Vreeken, and Dafna Shahaf. Towards Concept-Aware Large Language Models. arXiv, November 2023. arXiv:2311.01866 [cs].

[4] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality, 2013.

[5] George Kingsley Zipf. The meaning-frequency relationship of words. 33(2):251–256.

[6] Mark Steyvers and Kenneth Malmberg. The effect of normative context variability on recognition memory. *Journal of experimental psychology. Learning, memory, and cognition*, 29:760–6, 09 2003.

[7] Xiaojun Chen, Shengbin Jia, and Yang Xiang. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, 141:112948, 2020.

[8] Qi Wu, Chunhua Shen, Lingqiao Liu, Anthony Dick, and Anton van den Hengel. What value do explicit high level concepts have in vision to language problems?, 2016.

[9] Yi Zhang, Ce Zhang, Yushun Tang, and Zhihai He. Cross-modal concept learning and inference for vision-language models. *Neurocomputing*, 583:127530, 2024.

[10] Yuan Li, Yue Huang, Yuli Lin, Siyuan Wu, Yao Wan, and Lichao Sun. I think, therefore i am: Benchmarking awareness of large language models using awarebench, 2024.

[11] Zhen Tan, Lu Cheng, Song Wang, Yuan Bo, Jundong Li, and Huan Liu. Interpreting pretrained language models via concept bottlenecks. *ArXiv*, abs/2311.05014, 2023.

[12] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[13] Ricardo J. Campello, Davoud Moulavi, and Joerg Sander. Density-based clustering based on hierarchical density estimates. *Advances in Knowledge Discovery and Data Mining*, page 160–172, 2013.

[14] AI@Meta. Llama 3 model card. 2024.

[15] Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, 2015.

[16] George A. Miller. Wordnet: A lexical database for english, 1995. Princeton University.

A Appendix (optional)

If you wish, you can include an appendix, which should be part of the main PDF, and does not count towards the 6-8 page limit. Appendices can be useful to supply extra details, examples, figures, results, visualizations, etc. that you couldn't fit into the main paper. However, your grader *does not* have to read your appendix, and you should assume that you will be graded based on the content of the main part of your paper only.