CoDA21: Evaluating Language Understanding Capabilities of NLP Models With Context-Definition Alignment

Anonymous ACL submission

Abstract

Pretrained language models (PLMs) have achieved superhuman performance on many benchmarks, creating a need for harder tasks. We introduce CoDA21 (Context Definition Alignment), a challenging benchmark that measures natural language understanding (NLU) capabilities of PLMs: Given a definition and a context each for k words, but not the words themselves, the task is to align the k definitions with the k contexts. CoDA21 requires a deep understanding of contexts and definitions, including complex inference and world knowledge. We find that there is a large gap between human and PLM performance, suggesting that CoDA21 measures an aspect of NLU that is not sufficiently covered in existing benchmarks.

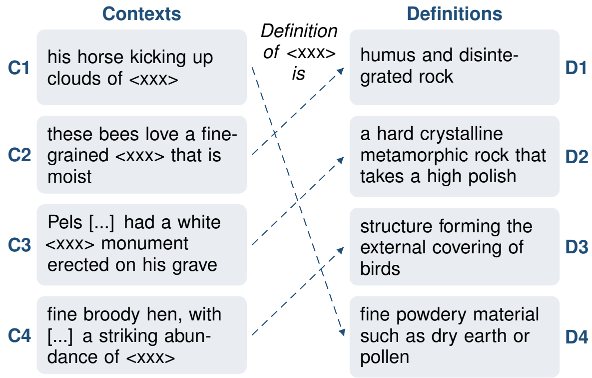

Figure 1: The CoDA21 task is to find the correct alignment between contexts and definitions: **C1-D4**, **C2-D1**, **C3-D2**, **C4-D3**. The target words for **C1-C4** ("dust", "soil", "marble", "feathers"; not given) are replaced with a placeholder <xxx>.

1 Introduction

Increasing computational power along with the design and development of large and sophisticated models that can take advantage of enormous corpora has drastically advanced NLP. For many tasks, finetuning pretrained transformer-based language models (Vaswani et al., 2017; Devlin et al., 2019; Radford et al., 2018) has improved the state of the art considerably. Language models acquire knowledge during pretraining that is utilized during task-specific finetuning. On benchmarks that were introduced to encourage development of models that do well on a diverse set of NLU tasks (e.g., GLUE[1] (Wang et al., 2018) and SuperGLUE[2] (Wang et al., 2019)), these models now achieve superhuman performance (He et al., 2020). The pretrain-then-finetune approach usually requires a great amount of labeled data, which is often not available or expensive to obtain, and results in specialized models that can perform well only on a single task. Recently, it was shown that generative language models can be applied to many tasks without finetuning when the task is formulated as text generation and the PLM is queried with a natural language prompt (Radford et al., 2019; Brown et al., 2020).

Motivated by recent progress in zero-shot learning with generative models as well as the need for more challenging benchmarks that test language understanding of language models, we introduce CoDA21 (**Co**ntext **D**efinition **A**lignment), a difficult benchmark that measures NLU capabilities of PLMs. Given a definition and a context each for k words, but not the words themselves, the task is to align the k definitions with the k contexts. In other words, for each definition, the context in which the defined word is most likely to occur has to be identified. This requires (i) understanding the definitions, (ii) understanding the contexts and (iii) the ability to match the two. Since the target words are not given, a model must be able to distinguish subtle meaning differences between different contexts/definitions to be successful. To illustrate the difficulty of the task, Figure 1 shows a partial example for $k = 4$ (see supplementary for the full example). We see that both complex inference (e.g., <XXX> can give rise to a cloud by being kicked up

[1] https://gluebenchmark.com/leaderboard
[2] https://super.gluebenchmark.com/leaderboard

⇒ <XXX> must be dry ⇒ <XXX> can be dust, but not soil) and world knowledge (what materials are typical for monuments?) are required for CoDA21.

We formulate the alignment task as a text prediction task and evaluate, without finetuning, three PLMs on CoDA21: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019) and GPT-2 (Radford et al., 2019). Poor performance of the PLMs and a large gap between human and PLM performance suggest that CoDA21 is an important benchmark for designing models with better NLU capabilities.

2 CoDA21

2.1 Dataset

We construct CoDA21 by first deriving a set $\mathcal{G}$ of *synset groups* $\{G_1, G_2, \ldots\}$ from Wordnet (Miller, 1995). A synset group G_i is a group of synsets whose meanings are close enough to be difficult to distinguish (making the task hard), but not so close that they become indistinguishable for human and machine. In a second step, each synset group G_i is converted into a *CoDA21 group* G_i^+ – a set of triples, each consisting of the synset, its definition and a corpus context. A CoDA21 group can be directly used for one instance of the CoDA21 task.

Synset groups. Each synset group G consists of $5 \leq k \leq 10$ synsets. To create a synset group, we start with a *parent synset* $\hat{s}$ and construct a co-hyponym group $\bar{G}(\hat{s})$ of its children:

$$\bar{G}(\hat{s}) = \{s \mid s < \hat{s}, s \notin D\}$$

where $<$ is the hyponymy relation between synsets and D is the set of synsets that have already been added to a synset group. The intuition for grouping synsets with a common parent is that words sharing a hypernym are difficult to distinguish (as opposed to randomly selected words).

We iterate $\hat{s}$ through all nouns and verbs in WordNet. At each iteration, we get all hyponyms of $\hat{s}$ that have not been previously added to a synset group; not reusing a synset ensures that different CoDA21 subtasks are not related and so no such relationships can be exploited. We extract synset groups from co-hyponym groups by splitting them into multiple chunks of size k, where each chunk contains synsets whose definitions are most dissimilar from each other (see Appendix for details).

CoDA21 groups. For each synset s, we extract its definition $d(s)$ from WordNet and a context $c(s)$

Dataset	# of G^{noun}	# of G^{verb}
CoDA21-*clean-hard*	106	102
CoDA21-*clean-easy*	274	103
CoDA21-*noisy-hard*	691	350
CoDA21-*noisy-easy*	1188	370

Table 1: CoDA21 group (G) statistics

in which it occurs from SemCor.[3] SemCor[4] is an English corpus tagged with WordNet senses. Let $C(s)$ be the set of contexts of s in SemCor. If $|C(s)| > 1$, we use as $c(s)$ the context in which *bert-base-uncased* gives s the highest log probability (averaged for multi-token instances) – this favors contexts that are specific to the meaning of the synset. Finally, we convert each synset group G_i in $\mathcal{G}$ to a CoDA21 group G_i^+:

$$G_i^+ = \{(s_j, d(s_j), c(s_j)) \mid s_j \in G_i\}$$

That is, a CoDA21 group G_i^+ is a set of of triples of sense, definition and context. In PLM evaluation, each CoDA21 group G_i^+ gives rise to one context-definition alignment subtask.

We name the resulting dataset *CoDA21-noisy-hard*: *noisy* because if $|C(s)|$ is small, the selected context may not be informative enough to identify the matching definition; *hard* because the synsets in a CoDA21 group are taxonomic sisters, generally with similar meanings despite the clustering-based limit on definition similarity. We construct a *clean* version of the dataset by only using synsets with $|C(s)| \geq 5$. We also construct an *easy* version by taking the "hyponym grandchildren" s of a parent synset $\hat{s}$ ($s < m \wedge m < \hat{s}$) instead of its hyponym children. This reduces the similarity of synsets in a CoDA21 group, making the task easier. Table 1 gives dataset statistics.

2.2 Alignment

Recall the CoDA21 task: given a definition and a context each for k words (but not the words themselves), align the k definitions with the k contexts. That is, we are looking for a bijective function (a one-to-one correspondence) between definitions and contexts. Our motivation in designing the task is that we want a hard task (which can guide us in developing stronger natural language understanding models), but also a task that is solvable by humans. Our experience is that humans can at

[3]We do not consider synsets without contexts in SemCor.
[4]http://lcl.uniroma1.it/wsdeval/home

least partially solve the task by finding a few initial "easy" context-definition matches, removing them from the definition/context sets and then match the smaller remaining number of definitions/contexts.

The number of context-definition pairs scales quadratically ($O(k^2)$) with k and the number of alignments factorially ($O(k!)$). We restrict k to $k \leq 10$ to make sure that we do not run into computational problems and that humans do not find the task too difficult.

Let t be a target word, c a context in which t occurs and m a made-up word. To test PLMs on CoDA21, we use the following two patterns:

$$Q_{\text{noun}}(c, m) = c_m \text{ Definition of } m \text{ is}$$
$$Q_{\text{verb}}(c, m) = c_m \text{ Definition of } m \text{ is to}$$

where c_m is c with each occurrence of t replaced by m.

We calculate the *match score* of a context-definition pair (c, d) as $\log P(d \mid Q(c, m))$, i.e., as the log generation probability of the definition d conditioned on $Q(c, m)$ where Q is either Q_{noun} or Q_{verb}, depending on the target word. Our objective is to maximize the sum of the k match scores in an alignment. We find the best alignment by exhaustive search. The accuracy for a CoDA21 group G_i^+ is then the accuracy of its best alignment, i.e., the number of contexts in G_i^+ that are aligned with the correct definition, divided by the total number of contexts $|G_i^+|$.

2.3 Baselines

We calculate $P(d \mid Q(c, m))$ for a masked language model (MLM) M and an an autoregressive language model (ALM) A as follows:

$$P_M(d \mid Q') = \prod_{i=1}^{|d|} P(d_i \mid Q', d_{-i})$$
$$P_A(d \mid Q') = \prod_{i=1}^{|d|} P(d_i \mid Q', d_1, \ldots, d_{i-1})$$

where $Q' = Q(c, m)$, d_i is the i^{th} word in definition d and d_{-i} is the definition with the i^{th} word masked.

We evaluate the MLMs BERT and RoBERTa and the ALM GPT-2. We experiment with both base and large versions of BERT and RoBERTa and with all four sizes of GPT-2 (small, medium, large, xl), for a total of eight models, to investigate the effect of model size on performance.

The made-up word m should ideally be unknown so that it does not bias the PLM in any way. However, there are no truly unknown words for the models we investigate due to the word-piece tokenization they apply to the input. Any made-up word that is completely meaningless to humans will have a representation in the models' input space based on its tokenization. To minimize the risk that the meaning of the made-up word may bias the model, we use m = *bkatuhla*, a word with an empty search result on Google that most likely never appeared in the models' pretraining corpora.

In addition to PLMs, we also evaluate 2 recent sentence transformer models[5] (Reimers and Gurevych, 2019), *paraphrase-mpnet-base-v2* (mpnet) and *paraphrase-MiniLM-L6-v2* (MiniLM), and fastText static embeddings[6] (Mikolov et al., 2018). To calculate the match score of a context-definition pair, we first remove the target word from the context and represent contexts and definitions as vectors. For sentence transformers, we obtain these vectors by simply encoding the input sentences. For fastText, we average the vectors of the words in contexts and definitions. We then calculate the match score as the cosine similarity of context and definition vectors.

	clean hard		clean easy		noisy hard		noisy easy		S20
Model	N	V	N	V	N	V	N	V	N
BERT$_b$	.20	.21	.22	.25	.21	.22	.22	.24	.24
BERT$_l$	.22	.22	.19	.21	.19	.20	.20	.20	.22
RoBERTa$_b$	.24	.26	.26	.32	.25	.25	.28	.27	.29
RoBERTa$_l$	.26	.30	.30	.30	.27	.29	.30	.33	.29
GPT-2$_s$	.31	.32	.42	.40	.35	.32	.40	.36	.35
GPT-2$_m$	.37	.35	.45	.39	.38	.35	.43	.39	.39
GPT-2$_l$	.38	.34	.47	**.42**	.39	**.37**	**.46**	.41	.47
GPT-2$_{xl}$	**.42**	.36	**.49**	**.42**	**.40**	.36	**.46**	**.43**	.48
mpnet	**.42**	**.39**	.48	**.42**	**.40**	**.37**	**.46**	.40	.51
MiniLM	.35	.34	.40	.36	.34	.30	.38	.32	.34
fastText	.18	.17	.20	.20	.18	.18	.18	.18	.17
Random	.15	.15	.14	.14	.16	.15	.14	.14	.14
Human	–	–	–	–	–	–	–	–	**.86**

Table 2: Average accuracy on the noun (N) and verb (V) subsets of CoDA21 for eight PLMs, two sentence transformers, fastText embeddings and (on S20) for humans

3 Results

Table 2 presents average accuracy of the investigated models on the four CoDA21 datasets. As can be seen, fastText performs only slightly bet-

[5] https://www.sbert.net/docs/pretrained_models.html

[6] We use the *crawl-300d-2M-subword* model from https://fasttext.cc/docs/en/english-vectors.html

ter than random. MLMs also perform better than random chance by only a small margin. This poor performance can be partly explained by the generation style setup we use, which is not well suited for masked language models. Even the smallest GPT-2 model performs considerably better than RoBERTA-large, the best performing MLM. Performance generally improves with model size. GPT-2_{xl} achieves the best results among the LMs on almost all datasets. Interestingly, sentence transformer *all-mpnet-base-v2* performs comparably to GPT-2_{xl} on most datasets despite its simple, similarity based matching compared to generation based matching of GPT-2 models. *Based on this observation it can be argued that current state of the art language models fail to perform complex, multi-step reasoning and inference which are necessary to solve the CoDA21 tasks.* Overall, MLMs perform slightly better on verbs than nouns while the converse is true for GPT-2. As expected, all models perform better on the *easy* datasets. Performance on *noisy* and *clean* datasets are comparable; this indicates that our contexts are of high quality even for the synsets with only a few contexts.

To investigate the **effect of the made-up word** m, we experiment with several other words on the noun part of CoDA21-*clean-easy* using GPT-2_{xl}. When m is a frequent word like "orange" or "cloud", performance drops (0.41 and 0.40 accuracy, respectively) due to the effect of prior knowledge models have about these words. The single letter "x" results in better performance (0.45 accuracy), possibly due to not having a strong specific meaning. Another nonce word "opyatzel" performs worse than "bkatuhla" (0.44 vs 0.49 accuracy), which indicates some random variation.

We compared our patterns Q_{noun} and Q_{verb} to two alternatives, but the difference in performance was minimal. See supplementary for details.

Human performance on CoDA21. We asked two NLP PhD students[7] to solve the task on S20, a random sample of size 20 from the noun part of CoDA21-*clean-easy*. Table 2 shows results on S20 for these two subjects and our models. Human performance is 0.86 – compared to 0.48 for GPT-2_{xl}, the best performing model. This difference indicates that there is a large gap in NLU competence between current language models and humans and that CoDA21 is a good benchmark to track progress on closing that gap.

[7] Both are proficient (though not native) English speakers.

To get a better sense of why the task is hard for PLMs, we give an example, from the CoDA21 subtask in Figure 1, of a context-definition match that is scored highly by GPT-2_{xl}, but is not correct. **Context:** "these bees love a fine-grained <XXX> that is moist". **Definition:** "fine powdery material such as dry earth or pollen". GPT-2_{xl} most likely gives a high score because it has learned that *bees* and *pollen* are associated. It does not understand that the mutual exclusivity of "moist" and "powdery" makes this a bad match.

4 Related Work

There are many datasets (Levesque et al., 2012; Rajpurkar et al., 2016; Williams et al., 2018) for evaluating language understanding of models. Many adopt a text prediction setup: Lambada (Paperno et al., 2016) evaluates the understanding of discourse context, StoryCloze (Mostafazadeh et al., 2016) evaluates commonsense knowledge and so does HellaSwag (Zellers et al., 2019), but examples were adversarially mined. LAMA (Petroni et al., 2019) tests the factual knowledge contained in PLMs. In contrast to this prior work, CoDA21 goes beyond prediction by requiring the matching of pieces of text. WIC (Pilehvar and Camacho-Collados, 2019) is also based on matching, but CoDA21 is more complex (multiple contexts/definitions as opposed to a single binary match decision) and is not restricted to ambiguous words. WNLaMPro (Schick and Schütze, 2020) evaluates knowledge of subordinate relationships between words, and WDLaMPro (Senel and Schütze, 2021) understanding of words using dictionary definitions. Again, matching multiple pieces of text with each other is much harder and therefore a promising task for benchmarking NLU.

5 Conclusion

We introduced CoDA21, a new challenging benchmark that tests natural language understanding capabilities of PLMs. Performing well on CoDA21 requires detailed understanding of contexts, performing complex inference and having world knowledge, which are crucial skills for NLP. All models we investigated perform clearly worse than humans, indicating a lack of these skills in the current state of the art in NLP. CoDA21 therefore is a promising benchmark for guiding the development of models with stronger NLU competence.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*. Citeseer.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

George A Miller. 1995. WordNet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534, Berlin, Germany. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning 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 1267–1273, Minneapolis, Minnesota. Association for Computational Linguistics.

A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. In *Technical Report*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

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

Timo Schick and Hinrich Schütze. 2020. Rare words: A major problem for contextualized embeddings and how to fix it by attentive mimicking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:8766–8774.

Lutfi Kerem Senel and Hinrich Schütze. 2021. Does she wink or does she nod? a challenging benchmark for evaluating word understanding of language models. In *Proceedings of the 16th Conference of* *the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 532–538, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

A Appendices

A.1 Extracting Synset Groups from Co-hyponym Groups

In an initial exploration, we found that the task is hard to solve for human subjects if two closely related hyponyms are included, e.g., “clementine” and “tangerine”. We therefore employ clustering to assemble a set of mutually dissimilar hyponyms. We first compute a sentence embedding for each hyponym definition using the *stsb-distilbert-base* Sentence Transformer[8] model. We then cluster the embeddings using complete-link clustering, combining the two most dissimilar clusters in each step. We stop merging before the biggest cluster exceeds the maximum group size ($k = 10$) or before the similarity between the last two combined clusters exceeds the maximum similarity ($\theta = 0.8$). The largest cluster G is added to the set $\mathcal{G}$ of synset groups. We then iterate the steps of (i) removing the synsets in the previous largest cluster G from $\bar{G}(\hat{s})$ and (ii) running complete-link clustering and adding the resulting largest cluster G to $\mathcal{G}$ until fewer than five synsets remain in $\bar{G}(\hat{s})$ or no cluster can be formed whose members have a similarity of less than θ.

A.2 Effect of Pattern

We compared our pattern Q_{noun} with two alternative patterns by evaluating GPT-2_{xl} on the noun part of CoDA21-*clean-easy*. Patterns and the evaluation results are shown in Table 3. The results suggest that the **effect of the pattern** on performance is minimal.

Pattern	Acc
<CTXT> Definition of <XXX> is	0.49
<CTXT> <XXX> is defined as	0.51
<CTXT> <XXX> is	0.49

Table 3: Effect of the pattern on the performance of GPT2-$_{xl}$ on the noun part of CoDA21-*clean-easy*

A.3 Effect of Alignment Setup

We constructed CoDA21 as an alignment dataset which uses the fact that matching between the definitions and contexts is one-to-one. This setup makes the task more intuitive and managable for humans. However, context-definition match scores

[8] https://huggingface.co/sentence-transformers/stsb-distilbert-base

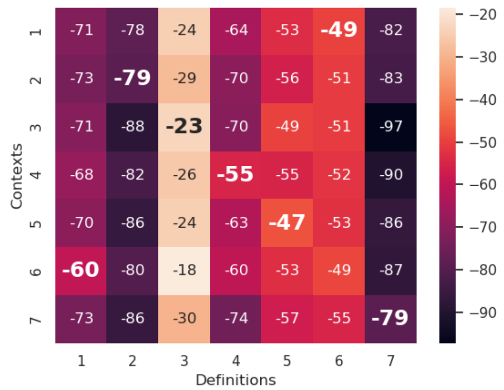

Figure 2: Match scores from GPT2-xl model for the context definition pairs for the sample given in Table 4. Match scores shown in bold correspond the context-definition pairs that are in the predicted alignment by the model that yields maximum total match score.

can be used to evaluate models on CoDA21 samples also without the alignment setup by simply picking context-definition pairs with the highest match score for each definition. We additionally evaluated GPT-2_{xl} model on CoDA21-*clean-easy* dataset using this simple matching approach which yielded 0.38 average accuracy compared to the 0.49 accuracy achieved with the alignment setup. This result suggests that language models can also make use of the alignment style evaluation, similar to humans.

Table 4 presents a sample of size 7 from the noun part of the CoDA21-*clean-easy* dataset. Figure 2 displays all 49 match scores of the context-definition pairs for this sample obtained using GPT-2_{xl}. 5 of the 7 definitions (2,3,4,5,7) are matched with correct contexts with the alignment setup while 4 definitions (4,5,6,7) are matched correctly for the simpe matching setup. Alignment setup enabled the model to match second and third definitions with their corresponding contexts even though thier match scores are not the highest ones.

Hidden word	Context
dust	1. He came spurring and whooping down the road , his horse kicking up clouds of <XXX> , shouting :
marble	2. Pels also sent a check for $ 100 to Russell 's widow and had a white <XXX> monument erected on his grave .
wastewater	3. The high cost of land and a few operational problems resulting from excessive loadings have created the need for a <XXX> treatment system with the operational characteristics of the oxidation pond but with the ability to treat more organic matter per unit volume .
feathers	4. It was a fine broody hen , white , with a maternal eye and a striking abundance of <XXX> in the under region of the abdomen .
fraction	5. It was then distilled at least three times from a trap at - 78 ' to a liquid air trap with only a small middle <XXX> being retained in each distillation .
soil	6. The thing is that these bees love a fine-grained <XXX> that is moist ; yet the water in the ground should not be stagnant either .
cards	7. And the coffee shop on Drexel Street , where the men spent their evenings and Sundays playing <XXX> , had a rose hedge beneath its window .
Synset	**Definition**
dust.n.01	1. fine powdery material such as dry earth or pollen that can be blown about in the air
marble.n.01	2. a hard crystalline metamorphic rock that takes a high polish; used for sculpture and as building material
effluent.n.01	3. water mixed with waste matter
feather.n.01	4. the light horny waterproof structure forming the external covering of birds
fraction.n.01	5. a component of a mixture that has been separated by a fractional process
soil.n.02	6. the part of the earth's surface consisting of humus and disintegrated rock
card.n.01	7. one of a set of small pieces of stiff paper marked in various ways and used for playing games or for telling fortunes

Table 4: A sample CoDA21 question taken from the noun part of the CoDA21-*clean-easy* dataset. The synsets are grandchildren of the parent synset 'material.n.01' whose definition is "the tangible substance that goes into the makeup of a physical object".