Towards Concept-Aware Language Models

Anonymous ACL submission

Abstract

Concepts play a pivotal role in various human cognitive abilities. However, there has been relatively little work on endowing machines with the ability to form and reason with concepts. In particular, pretrained language models (LMs) work at the level of *words*, not concepts. This is problematic as different words relating to the same concept compete for probability mass.

Here we take the first step towards developing a *concept-aware LM*. Instead of rethinking the training process, we adapt existing LMs. We build a proof-of-concept LM outputting a ranked list of *concepts*, and show that they are relatively coherent and diverse. We demonstrate that concepts could help improve the LM's ranking and robustness. While this work is rather preliminary, we believe concept-aware LM can benefit many downstream tasks.

1 Introduction

Concepts are the glue that holds our mental model of the world together. It is hard to see how any intelligent agent could do without them. They are what enables us to comprehend new situations in terms of previous ones: when we walk into a new situation (e.g., a restaurant) full of new objects and people, we interpret it using learned concepts.

Concepts can be concrete ("soup") or abstract ("tasty"). They can also be complex, e.g., "good winter beach destinations". While there is a lively debate on the exact nature of concepts, researchers agree they **play a pivotal role in various cognitive abilities** such as categorization, learning, communicating, planning, and decision-making (Murphy, 2004). Thus, they are of interest to AI researchers wishing to endow machines with such abilities.

The representation of concepts has been studied in ML, NLP, and knowledge representation (Fumagalli and Ferrario, 2019; Davis and Marcus, 2015; Gardenfors, 2014). However, they often view concepts as fixed, shallow structures representing some set of entities. Recent studies suggest concepts are more flexible and dynamically influenced by context (Gabora et al., 2008). Unfortunately, AI still struggles with accounting for the creative, context-sensitive manner in which people employ concepts.

Here we focus on adding concepts to language models (LMs). Recently, pretrained large LMs (Yang et al., 2019; Raffel et al., 2020; Floridi and Chiriatti, 2020) have gained immense popularity, achieving SOTA results across the board. A fundamental LM task is *text completion*. However, using tokens (rather than concepts) leads to *surface form competition*: different surface forms compete for the same the probability mass, even if they share the same meaning, e.g., "mother" and "mom" (Holtzman et al., 2021), which distorts the ranking.

Here we take the first step towards developing **concept-aware LMs**. Instead of rethinking LMs' training, we take the simpler approach of adapting *existing* ones. Our method is model-agnostic, operating on any pretrained LM. Previous works showed it was possible to enhance pretrained LMs without further training and improve their performance on tasks such as word sense disambiguation, factualness and consistency (Levine et al., 2020; Liu et al., 2022). We believe concept-aware LMs could similarly enhance downstream tasks.

2 Problem definition

We focus on the fundamental LM task of text completion, namely fill-mask. Given a masked sentence $\mathcal{S} \in \Omega$ and an LM, our goal is to return a *ranked list of concepts* $C_1, ..., C_N$ (representing completions). Each concept C_i is a non-empty set of surface-level tokens $t \in \Omega$. Ideally, the concepts and their ranking should correspond to human intuition.

3 Algorithm

Overview. As we wrote above, rather than rethinking LM's architecture and training process, we take

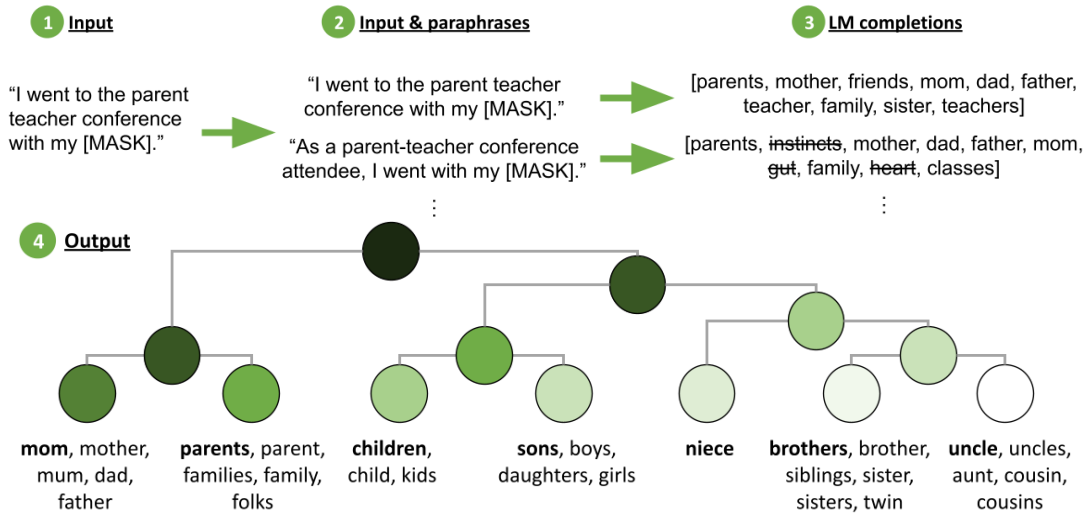

Figure 1: Overview of our algorithm. The input is a masked sentence. We augment it by paraphrasing and predict the top k completions for each of the paraphrases. Next, we filter out rare and unlikely tokens (strikethrough) and perform agglomerative clustering using the token-contextual embeddings from the input LM (centroid in bold). We assign new weights to each node in the dendrogram (darker ranked higher, sorted according to weight).

the simpler, proof-of-concept approach of building concepts on top of the output of existing LMs.

Figure 1 demonstrates our algorithm. In short, given a masked sentence $\mathcal{S}_0$, we retrieve top completions using the LM (coming up with several paraphrases of $\mathcal{S}_0$ as an augmentation technique, to increase robustness). To form concepts, we perform agglomerative clustering using the contextual embeddings. Each node in the dendrogram is assigned with a weight based on its tokens' both weights and repetitions across augmentations.

For clarity of presentation, Figure 1 shows clusters (using a distance threshold), rather than singleton tokens. In this example (parent-teacher conference), the most likely concept contains tokens such as "mom", "mother" and "dad", followed closely by a concept containing "parents" and "family". Next concepts refer to children and other family members in general. The top node indicates the completion is probably a family member.

We present our implementation (code can be found at [URL redacted for anonymity]). We give a succinct overview, for details see the Appendix.

Augmentation. To augment the input $\mathcal{S}_0$ we first retrieve the LM's top-k completions.[1] We replace the [MASK] token with the first completion that is not a stopword or a sub-word and paraphrase using wordtune.[2] We then mask $\{\mathcal{S}_0, ..., \mathcal{S}_{M-1}\}$.

Top-k completions retrieval. We retrieve the top-k (k=100) completions for each sentence in $\{\mathcal{S}_0, ..., \mathcal{S}_{M-1}\}$. We count how often each completions appears and remove infrequent ones.[1] We extract the contextual embeddings for all remaining completions (the token embedding from the last hidden layer using $\mathcal{S}_0$). We use the contextual embedding due to the importance of context.

Clustering & Ranking. We reduce the dimensionality using PCA and t-SNE[1], and use agglomerative clustering to cluster the completions into concepts. We use agglomerative clustering as different thresholds yield different concept-granularity, similar to the flexibility of concepts in humans. Each cluster is assigned with a weight that corresponds to 1) the token with the maximal soft-max score, to avoid problems related to surface form competition, and 2) the token with the maximal number of repetitions across augmentations' top-100 completions, to increase robustness (a token that repeated frequently is probably very relevant).[1]

4 Evaluation

To evaluate our concept-aware method, we focus on *fill-mask* task (completing a masked sequence).

Dataset. We use the ProtoQA dataset, consisting of questions regarding prototypical situations (e.g., "Name something you are likely making if you buy milk, eggs, sugar and cream.") (Boratko et al., 2020). We believe this setting is relevant for our use case, as there are usually several related answers. To make the input similar to the language LMs

[1] See details in appendix.
[2] https://www.wordtune.com/

Input sentence	Completion	BERT	Concept-BERT
I bought a fake [MASK] from a street vendor.	jersey	0.08	0.79
When I retired I started [MASK].	cycling	0.06	0.77
Whenever I suffer from cold I always [MASK].	shudder	0.04	1
	rise	0.93	0.24
When I go to the beach I use [MASK] to protect myself from the sun.	sticks	0.91	0.28
	soap	0.74	0.03
I always take my [MASK] with me to the gym.	laptop	0.76	0.29
I squeezed myself into the [MASK].	sand	0.71	0.23

Table 1: Examples of completions for which the weight BERT and concept-BERT assign are notably different. Our manipulation increases the score of appropriate completions and decreases the weight of inappropriate ones. Weight calculation: $(1 - < completion\ rank >)/k$ where k=100 for BERT and k=number of outputted clusters for concept-BERT. Color coding: red=low weight, orange=intermediate, green=high.

are usually trained on, we manually changed the questions to first-person statements ("I bought milk, eggs, sugar and cream to make a [MASK]."). We used 63 sentences to fine-tune our parameters and an additional 100 sentences for evaluation.[1]

Experiments. We used BERT, the most popular fill-mask LM (Devlin et al., 2018);[3] with and without our method, outputting BERT's top 100 completions and concept-BERT's ranked *clusters*.

In the following, we verify the clusters are coherent and distinct (§4.1) and the ranking is meaningful (§4.2). In §4.3 we explore disagreements between BERT and concept-BERT.

4.1 Cluster quality

As a sanity check, we measured clusters' semantic coherence using the cosine similarity of word2vec's token embedding (first ten clusters for all sentences). The mean **within cluster** similarity is 0.41, whereas the mean **inter cluster** is 0.12. For reference, BERT's similarity (top ten completions) is 0.22. Hence, our clusters are coherent and distinct.

A closer examination of the clusters highlights the distinction between the next-token-prediction approach and ours. Consider the sentence "I can't get home for the holidays because of the [MASK]." and its cluster: {blizzard, cold, snow, snowfall, temperature, weather}. While this is a coherent concept (cold weather conditions), some specific *tokens* are less-natural completions without their cluster-context (e.g., temperature).

We note that our clustering approach is rather simple, and sometimes fails to capture nuances. Consider "I forgot to take off my [MASK] before going to bed." and its cluster {clothes, clothing, pajamas}. While pajamas is a type of clothing, it is the type people usually *put on* before going to bed.

4.2 Ranking quality

We evaluate the **precision** of concept-BERT by annotating all completions in the top ten clusters for all 100 input sentences. Three Amazon Mechanical Turk workers received the masked sentence and a possible completion, and were asked to classify the completion as either: likely /possible but unlikely /does not make sense (see qualifications, compensation, and instructions in Appendix). Note this evaluation cannot be automated, as we wish to see if our concept-aware modification aligns the LM's output with *humans*. Completion's aggregated score is its maximal score (mean-variance across annotations=0.17). Our **precision at k=1 is 72%** and 67% at k=10 (see Appendix Figure 3).

Since our clusters sometimes contain completions that make less sense than others (although belong to the same concept), we also treat the clusters as concept-indicators (e.g., cold weather conditions), and test the *percentage* of reasonable completions within each concept. If we search for clusters with at least *one* good completion, our mean precision $= 90\%$ for k={1, 10}. If we restrict to at least *two* good completions (mean cluster size=2.9), our mean precision is: k=1 (77%) and k=10 (75%).

4.3 Completions in dispute

We now focus on completions for which BERT and concept-BERT **disagree** – one predicts it is likely, while the other predicts it is much less likely (Table 1). These are the most interesting regions for evaluating the effect of our manipulation, specifically, how it corresponds to human judgment.

To do so, we treat the middle 15% of the ranked lists as buffer and output tokens that are above the buffer according to one model and below according to the other. We identified 282 disputed tokens.

[3]Most common according to Hugging Face: https://huggingface.co/models?pipeline_tag=fill-mask.

Scenario	Mean score	Norm. score
Concept-BERT ↑ BERT ↓	0.84	0.304
Buffer	0.74	-
Concept-BERT ↓ BERT ↑	0.66	-0.142

Table 2: Mean scores and mean normalized (using the buffer) scores of the three scenarios in the dispute evaluation. Tokens concept-BERT ranked as probable while BERT ranked as improbable (first row) are significantly higher than both the buffer (middle row) and the tokens BERT ranked high and concept-BERT low (bottom).

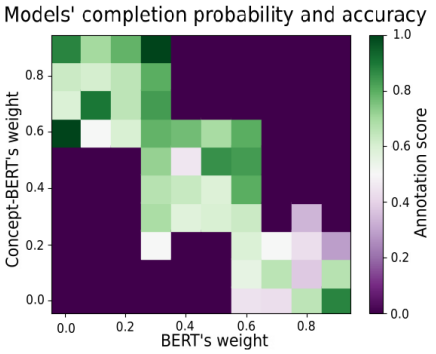

Figure 2: Heat-map of the disputed completions (higher means better). The y-axis represents concept-BERT's completion weight. The x-axis represents BERT's weight. The top-left part of the map received higher scores compared to the middle (buffer) and the bottom-right part. Meaning, our manipulation ranked high appropriate completions and low inappropriate ones.

In addition, we annotated completions that both models ranked in the middle 15% (buffer).

Volunteer computer science graduate students annotated 585 completions using the same setup as in §4.2. Each completion was annotated by two students and aggregated to its maximal score (mean variance across annotations=0.1).[1]

We divide the annotated completions to 3 groups and compute their mean score. As some sentences have more good completions than others, we also computed a normalized mean score (normalizing per sentence using the sentence's buffer scores). Results (Table 2) show that whenever the models disagree, concept-BERT is more often correct.

Figure 2 depicts a score-heat-map of the disputed completions. Y-axis represents concept-BERT's token weight, x-axis shows BERT's weight. The top-left part (concept-BERT=probable, BERT=improbable) scores are higher compared to the middle (buffer) and the bottom-right (the opposite scenario). Meaning, we rank appropriate completions high, and inappropriate ones low.

Next, we compute completions' accumulated mean accuracy as a function of rank given by each of the models. We expect a negative correlation as the quality should decrease when going down the ranked list) While concept-BERT does have a negative correlation, BERT's correlation is actually *positive* (meaning, its top-ranked completions are on average worse than the bottom-ranked ones). Both curves have significant correlation (p-values<0.05), whereas BERT's is weaker (coefficient 0.54 versus 0.91). We stress this is not a random sample, but rather the disputed completions (and buffer). Thus, we reveal appropriate completions and remove inappropriate ones.

Lastly, we also analyze the mean accuracy of the disputed completions as a function of how strict the threshold for "in dispute" is. BERT's accuracy decreases much sharper compared to concept-BERT ($> 10\%$ versus $< 2\%$), hinting our manipulation increases robustness (see Appendix Figure 5).

To conclude, even though we presented a simple implementation, it led to overall coherent concept-clusters with meaningful ranking. We see promising indications that this may enhance the LM's robustness and help identify good completions.

5 Conclusions & Future Work

We are inspired by the importance of *concepts* in human cognition, and specifically for language. In particular, LMs work at the level of *words*, rather than concepts. This is problematic as even if they represent the same underlying concept in a given context, different surface forms compete for probability mass, distorting the ranking.

To better align LMs with humans, we present a model-agnostic method to shift any off-the-shelf pretrained LM from the token- to the concept-level, without fine-tuning or adding any external information. We evaluate our concept-aware approach using BERT, outputting a ranked list of *concepts*, which are coherent and diverse. We show that our method improves the LM's ranking and robustness.

While this is only preliminary work, we believe concept-aware LMs can benefit many downstream tasks. Thus, in the future, we believe it would be beneficial to shift LMs' training towards *concepts*, either leveraging external data sources (similar in spirit to SenseBERT (Levine et al., 2020) and KID (Liu et al., 2022)), or through a more flexible system allowing ad-hoc concepts.

6 Limitations & Ethical Considerations

Our method heavily relays on the input LM, and thus might preserve some of the LM's biases. One might try to overcome these biases, e.g., by injecting external knowledge.

Another limitation of our method is the usage of not just the LM's completions, but also their embeddings. This does not let us apply our method to any LM that is accessible through API that exposes only its completion output.

Lastly, computation is somewhat slower than the LM's computation time, as we run it several times, plus postprocessing (paraphrasing extraction, dimensionality reduction, clustering, etc.). We note, however, that many of those operations are easily parallelizable.

References

Michael Boratko, Xiang Li, Tim O'Gorman, Rajarshi Das, Dan Le, and Andrew McCallum. 2020. ProtoQA: A question answering dataset for prototypical common-sense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1122–1136, Online. Association for Computational Linguistics.

Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Communications of the ACM*, 58(9):92–103.

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*.

Luciano Floridi and Massimo Chiriatti. 2020. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694.

Mattia Fumagalli and Roberta Ferrario. 2019. Representation of concepts in ai: Towards a teleological explanation. In *JOWO*.

Liane Gabora, Eleanor Rosch, and Diederik Aerts. 2008. Toward an ecological theory of concepts. *Ecological Psychology*, 20(1):84–116.

Peter Gardenfors. 2014. *The geometry of meaning: Semantics based on conceptual spaces*. MIT press.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051.

Yoav Levine, Barak Lenz, Or Dagan, Ori Ram, Dan Padnos, Or Sharir, Shai Shalev-Shwartz, Amnon Shashua, and Yoav Shoham. 2020. Sensebert: Driving some sense into bert. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4656–4667.

Ruibo Liu, Guoqing Zheng, Shashank Gupta, Radhika Gaonkar, Chongyang Gao, Soroush Vosoughi, Milad Shokouhi, and Ahmed Hassan Awadallah. 2022. Knowledge infused decoding. *arXiv preprint arXiv:2204.03084*.

Gregory Murphy. 2004. *The big book of concepts*. MIT press.

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. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Implementation details

LM. We used BERT-base-uncased with the default parameters. We performed no training.

Augmentation. For this phase we first replaced the missing token with the LM's most probable completion that contains more than three letters and is not a stop-word (using the $stopwords$ list from $nltk.corpus$ package). We inserted $\mathcal{S}_0$ to AI21's Wordtune paraphrasing model using the default parameters:

```
requests.post(
"https://api.ai21.com/studio/
v1/experimental/rewrite",
headers={"Authentication":
<ai21-private-token>}
json={"text": S_0,
"intent": "general"})
```

And extracted the text suggestions from the output JSON file. We then searched for the original completion and masked all sentences. Sentences in which we were unable to automatically find the word were dropped.

Top-k completions retrieval. We used $k = 100$ for each masked sentence. We drop each completion that did not appear in at least half of our augmentations. Note: another possible implementation would be a function of *unique* completions.

Clustering & Ranking. As a latent space representation of contextual token, we extract the LM's token embedding for this token from the last hidden layer with the input sentence $\mathcal{S}_0$. We reduce the dimensionality of the embeddings from 768 to 100 using PCA (scikit learn implementation, n_components=100, svd_solver='full') and from 100 to 10 using t-SNE (scikit learn implementation, n_components=10, init='pca', perplexity=10, method='exact').

We cluster the embeddings after the dimensionality reduction using agglomerative clustering using the distance metric cosine similarity, linkage='linkage', distance threshold=0.45, n_clusters=None, and compute_distances=True (scikit learn implementation)

To rank the clusters, we used the following formula:

$$weight(C_i) = \alpha \cdot max_{weight}(weight(t)\ \forall t \in C_i) + (1-\alpha) \cdot max_{rep}(rep(t)\ \forall t \in C_i)$$

where $\alpha = 0.7$.

Human annotators

For both annotation tasks (computer science graduate students and Amazon Mechanical Turk), annotators were presented with a sentence and a possible completion and were asked "Do you think this completion makes sense?". Possible responses are: {likely, possible but unlikely, does not make sense}.

Precision at k. We used Amazon Mechanical Turk with the following qualifications: {HIT Approval Rate > 98, Number of HITs Approved > 5000, Location is one of CA, GB, US (for English speakers)}. We also used a custom qualification using five example sentences and completions. Annotators were allowed to make one error in order to qualify. We paid annotators $0.02 per completion. Overall, we had 39 unique annotators.

Full instructions:

You will be presented with a sentence containing a missing word and a candidate word to fill-in the blank. Your role is to determine for each completion whether it is likely / possible but not likely / does not make sense at all. Note! If a completion is not grammatically correct ("I enjoy *raining*" instead of *rain*) that is fine, we do not care about grammar here. But if the sentence + completion is not a full sentence ("I enjoy *doing*") that is NOT fine, as the sentence is meaningless.

Example 1
Sentence: I went to the parent teacher conference with my _____.
Completion: parent
Desired response: Likely
Example 2
Sentence: I went to the parent teacher conference with my _____.
Completion: schedule
Desired response: Does not make sense
Example 3
Sentence: I went to the parent teacher conference with my _____.
Completion: grandfather
Desired response: Possible but unlikely
Explanation: While this is not the common scenario, it is still possible.
Example 4
Sentence: I went to the parent teacher conference with my _____.
Completion: mothers
Desired response: Likely OR Possible but unlikely

Completions in dispute. We recruited 8 volunteers, all are graduate students from the computer science department (same instructions as the Amazon Mechanical Turk experiment, see instructions above). Each student annotated about 150 completions (cutoff at the end of the sentence). Each completion was annotated by two students. Mean-variance across annotation=0.1, showing a fairly good quality of annotations (possible responses=$\{0, 0.5, 1\}$). Students reported the task to take about 15 minutes to complete.

Figures & Tables

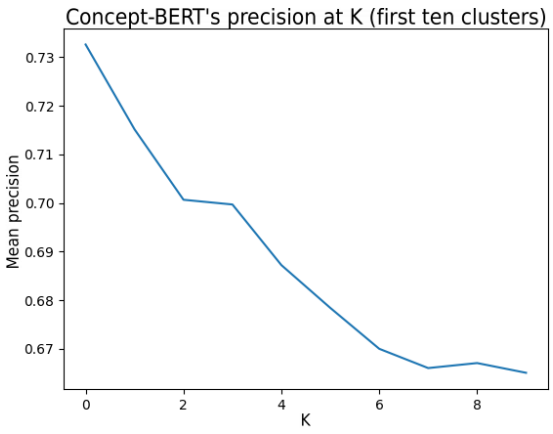

Figure 3: Concept-BERT's precision at k=10 (number of clusters annotated by three Amazon Mechanical Turk annotators). Our method's mean precision at k=1 is 72% and 67% at k=10.

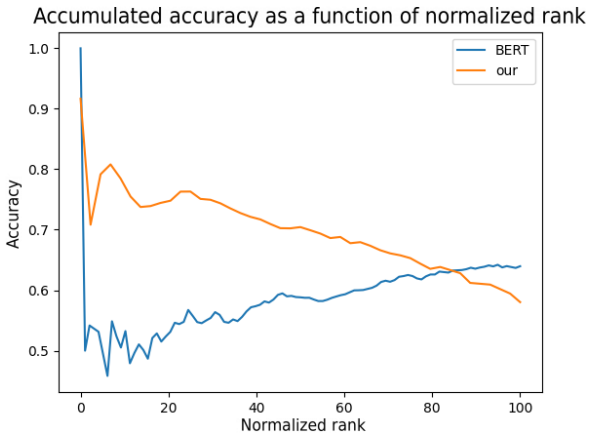

Figure 4: Completions' accumulated mean accuracy as a function of rank given by each of the models. Both curves have significant correlation (p-value < 0.05), whereas BERT's correlation is weaker (correlation coefficient 0.54 versus 0.91). Interestingly, while concept-BERT has a negative correlation, as expected since the rank's quality should decrease, BERT's correlation is positive. We stress this is not a random sample, but rather the disputed completions (and the buffer). Thus, this again strengthens our claim, that our manipulation helps to reveal appropriate completions and remove inappropriate ones, with respect to the original LM.

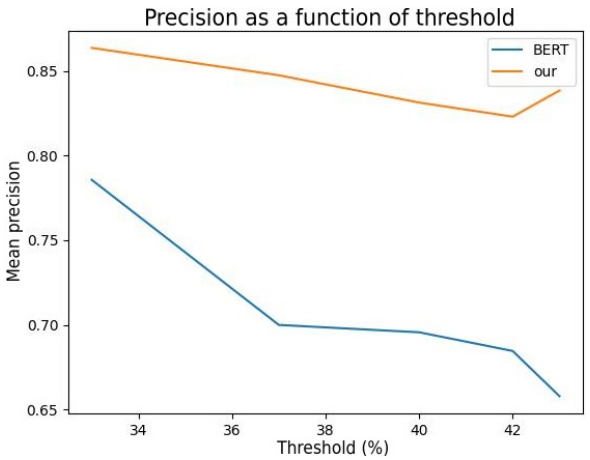

Figure 5: Mean accuracy of the disputed completions for both models as a function of how strict the threshold for disagreement is. BERT's accuracy decreases sharply compared to concept-BERT, suggesting our manipulation increases robustness.