Interpretable-by-Design Text Classification with Iteratively Generated Concept Bottleneck

Josh Magnus Ludan **Qing Lyu** **Yue Yang** **Liam Dugan**
Mark Yatskar **Chris Callison-Burch**
University of Pennsylvania
{jludan, lyuqing, yueyang1, ldugan, myatskar, ccb}@seas.upenn.edu

Abstract

Deep neural networks excel in text classification tasks, yet their application in high-stakes domains is hindered by their lack of interpretability. To address this, we propose Text Bottleneck Models (TBMs), an intrinsically interpretable text classification framework that offers both global and local explanations. Rather than directly predicting the output label, TBMs predict categorical values for a sparse set of salient concepts and use a linear layer over those concept values to produce the final prediction. These concepts can be automatically discovered and measured by a Large Language Model (LLM), without the need for human curation. On 12 diverse datasets, using GPT-4 for both concept generation and measurement, we show that TBMs can rival the performance of established black-box baselines such as GPT-4 fewshot and finetuned DeBERTa, while falling short against finetuned GPT-3.5. Overall, our findings suggest that TBMs are a promising new framework that enhances interpretability, with minimal performance tradeoffs, particularly for general-domain text.[1]

1 Introduction

Interpretability has become a critical aspect of deep learning systems, especially in high-stakes domains such as law, finance, and medicine, where understanding and analyzing model behavior is crucial (Bhatt et al., 2020; Dwivedi et al., 2023). A promising line of work focuses on "self-interpretable" models, which provide built-in explanations along with their predictions (Du et al., 2019; Linardatos et al., 2020). These model-provided explanations can come in various forms: token-level importance scores, influential training examples, or even free text. However, these types of explanations oftentimes provide only *local* justification for individual predictions and fail to offer *global* insights into the overarching principles that guide model behavior (Bhatt et al., 2020).

[1]Code is available at github.com/JMRLudan/TBM.

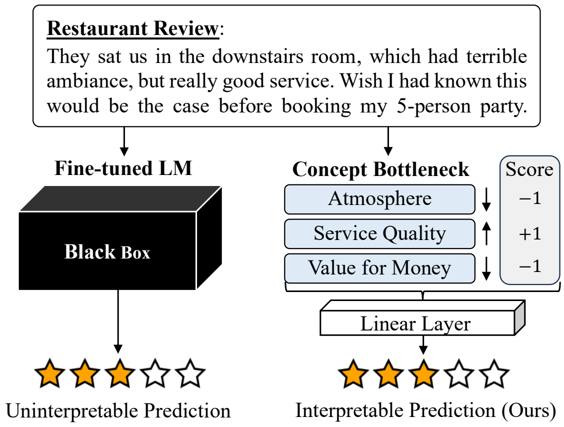

Figure 1: Unlike end-to-end black-box models (left), Text Bottleneck Models (right) first discover and measure a set of human-interpretable concepts and then predict the label with a linear layer.

An alternative form of explanation that addresses this issue is *concept*-based explanations (Madsen et al., 2022). A concept is an abstract feature representing some aspect of the input text, such as "food quality" for a restaurant review. Concept-based explanations can provide both global and local insights by identifying important concepts across the dataset and localizing how these concepts relate to each individual prediction. However, concept-based approaches typically involve extensive human labor to implement, since they require experts to curate a set of concepts for each new task, and the concept values need to be further annotated on each training example (Abraham et al., 2022). Additionally, current approaches often lack *sparsity*, including hundreds or even thousands of concepts in their explanations (Rajagopal et al., 2021). With such large concept spaces, it remains difficult to draw useful takeaways on the global behavior of the model (Ramaswamy et al., 2022).

In this work, we propose **T**extual **B**ottleneck **M**odels (TBMs), an extension of Concept Bottle-

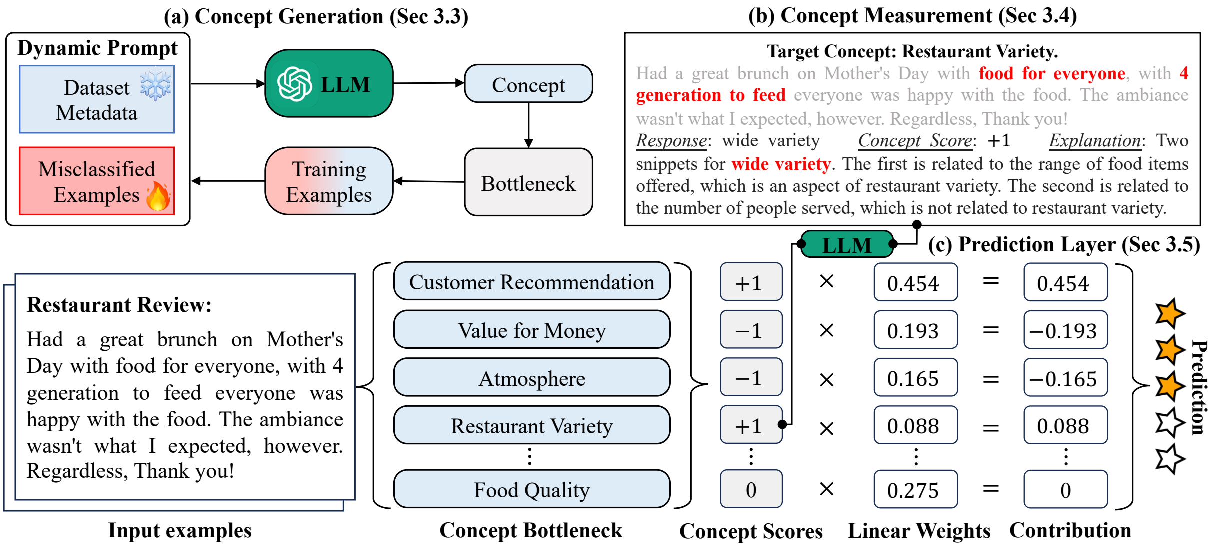

Figure 2: Demonstration of the system with an example from the CEBaB (Abraham et al., 2022) dataset. Given an input example (restaurant review), during Concept Generation (a), it iteratively discovers new concepts (e.g., "Restaurant Variety"). Concept Measurement (b) measures the value of concepts by identifying relevant snippets (e.g., "food for everyone") and providing a numerical concept score (e.g., +1). Finally, the Prediction Layer (c) aggregates all concept scores for the input and learns their relative weights to make final prediction of the task label.

neck Models (CBMs) from the vision domain (Koh et al., 2020) to text classification and regression tasks. Our system has three modules, all fully automated: Concept Generation, Concept Measurement, and Prediction Layer, as shown in Figure 2. Given a dataset of input texts (e.g., restaurant reviews), the Concept Generation module iteratively discovers a sparse set of concepts (e.g., "Restaurant Variety") that help discriminate between texts with different output labels. The Concept Measurement module then determines the value of each concept (e.g., "wide variety") for a text as a numerical score (e.g., +1). Finally, these concept scores are aggregated into the final prediction by a white-box Prediction Layer (e.g., a linear layer).

Using GPT-4 to generate and measure concepts, we evaluate our system on 12 diverse datasets, spanning from fake news detection to sentiment classification. TBMs perform competitively with strong black-box baselines including few-shot GPT-4 and finetuned BERT, but lag behind state-of-the-art models like finetuned GPT 3.5. In particular, TBMs are highly competitive for sentiment comprehension and natural language inference tasks, though there is room for improvement in specialized domains like news and science.

To understand where the error comes from, we perform a manual evaluation of each module. We find that the Concept Generation module can consistently generate relevant and unambiguous concepts, but can occasionally struggle with redundancy and leakage. The Concept Measurement module is found to score the majority of concepts in sentiment analysis with high accuracy, whereas those in fake news detection are harder to measure, which might be a reason behind the performance difference in these domains. Finally, the concept learning curves make it transparent what concepts are learned over time and their relative impact, which can offer valuable insights for model understanding and debugging.

Our contributions are as follows:
(1) We introduce TBMs, a text classification framework that provides both global and local interpretability, by automatically constructing sparse concept bottlenecks using LLMs without any human effort.
(2) We demonstrate that, on average, TBMs perform competitively with strong, but not state-of-the-art, black-box baselines across 12 diverse datasets.
(3) We provide an in-depth human evaluation and analysis of each module in the TBM and show how the system allows for easier model interaction and debugging.

2 Related Work

Self-interpretable NLP models aim to provide a built-in explanation along with the prediction, without relying on post-hoc explanation methods. They offer diverse forms of explanation. Token-based explanations, such as rationales (Lei et al., 2016; Bastings et al., 2019), provide a span of important tokens that are minimally sufficient for the prediction. Example-based explanations (Han et al., 2020; Das et al., 2022) identify the most similar examples within the training set relative to the examples for inference. Free-text explanations, such as those in (Camburu et al., 2018; Nye et al., 2021; Wei et al., 2022), generate a free-form justification in Natural Language for the prediction. We note that these only provide local interpretability, and our approach differs in that it provides both local and global insights into model behavior owing to the use of concept-based explanations.

Concept Bottleneck Models were first introduced by Koh et al. (2020) for vision tasks such as image classification. In their work, they tasked experts with manually crafting a set of human-interpretable concepts that then became the only input for a classifier model. Stakeholders could then intervene in these concepts and correct them, allowing easier model behavior analysis. Collins et al. (2023) describe several problems with CBMs, such as information leakage (Mahinpei et al., 2021) and having too many concepts (Ramaswamy et al., 2022). Information leakage causes the concept bottleneck to be *unfaithful* (Jacovi and Goldberg, 2020; Lyu et al., 2022) by having the labeling task as a concept. Having too many concepts causes information overload for the user, preventing them from developing a general understanding of model behavior. We note that these problems can also exist in the text domain, so we carefully evaluate them in our manual analysis.[2] To reduce the cost of concept generation, previous work in computer vision has also used LLMs to automate this process for image classification (Yang et al., 2023; Pratt et al., 2023). Our work extends this method to the text domain, with additionally introduced benefits such as sparsity.

Concept-based explanations in NLP can be broadly categorized into two lines of work. The first focuses on mechanistic interpretability, analyzing *what* latent concepts are represented by different neurons in pre-trained LMs (Sheng and Uthus, 2020; Bills et al., 2023; Vig et al., 2020). The second focuses on explaining *why* models make certain decisions, providing explicit concepts as supporting evidence for predictions (Rajagopal et al., 2021; Wu et al., 2023). Our work belongs to the second category.

Within this category, SELF-EXPLAIN (Rajagopal et al., 2021) is an explainable framework that jointly predicts the final label and identifies both globally similar concepts from the training set and locally relevant concepts from the current example. Notably, there is no bottleneck structure in their approach, which makes information leakage easier. Also, they define each *phrase* (e.g., "for days", "the lack of", etc.) in each example as a concept, resulting in an enormous concept space of hundreds of thousands of phrases. By contrast, our concepts are high-level, categorical *features*, resulting in a sparse space of ≤ 30 concepts for each dataset, from which it is easier to draw useful takeaways. Another representative work (Wu et al., 2023) trains a Causal Proxy Model that mimics the behavior of a black-box model using human-annotated counterfactual data. Our definition of concepts is consistent with theirs, but our method does not require expert data curation.

3 Method

Figure 2 provides an overview of our system. It consists of three components: **Concept Generation**, which iteratively discovers new concepts using misclassified examples; **Concept Measurement**, which measures the concept scores for each example; and **Prediction Layer**, which predicts the output label with only the concept scores as input. The first two modules are implemented by prompting an LLM,[3] and the last module is implemented as training a linear layer.

3.1 Method Formulation

We describe the structure of TBMs as follows:[4] Given a text classification or regression dataset with a training set $\mathcal{D}_{\text{train}}$ and a test set $\mathcal{D}_{\text{test}}$, each instance can be denoted as a text-label pair (t, y). During training, we generate a set of N concepts $C = \{c_1, c_2, \dots c_N\}$ using $\mathcal{D}_{\text{train}}$, where each concept c_i is a categorical feature (e.g., "restaurant variety") with multiple possible values (e.g., `high`, `low`, `mixed` or `unmentioned`). For each text t_{train}, we measure the values of all concepts as a list of numerical scores, $[s(t_{\text{train}}, c_i)|c_i \in C]$ (e.g., $+1$, -1, 0). The sign of the score represents the polarity of a concept in the text, i.e. a positive/negative score

[2] See Sec 5.2 for "Redundancy" and "Leakage" evaluation.

[3] See Appendix E for all relevant prompts.

[4] Note that this is a generic structure of TBMs independent of implementation.

Key	Value
Concept Name	Build Quality
Concept Description	Build quality refers to the craftsmanship, durability, and overall construction of a product. This concept encompasses various aspects such as the materials used, design, manufacturing techniques, and attention to detail.
Concept Question	What does the review say about the build quality of the product?
Possible Responses	Positive, Negative, Uncertain, Not applicable
Response Guide	**Positive:** The review mentions positive aspects such as being well-made, sturdy, durable, use of high-quality materials, excellent craftsmanship, etc. **Negative:** The review mentions negative aspects such as poor construction, flimsiness, use of cheap materials, bad design, being easily breakable, etc. **Uncertain:** The review does not clearly mention the build quality, provides ambiguous or vague information, or mentions both positive and negative aspects. **Not applicable:** The review does not mention the build quality of the product at all.
Response Mapping	Positive: $+1$, Negative: -1, Uncertain: 0, Not applicable: 0

Table 1: JSON Representation for the concept "Build Quality" for a hypothetical product review dataset included in the Concept Generation prompt as an in-context example.

indicates that the concept is positively/negatively reflected, and a zero score represents uncertainty or absence of the concept. The magnitude of the score represents the intensity of a concept, with larger magnitude indicating higher intensity. These concept scores are then used as the only input to train a white-box prediction layer to predict the label y_{train}. During inference, given a new input text $t_{\text{test}} \in \mathcal{D}_{\text{test}}$, we measure the score of each concept in the generated concept set $[s(t_{\text{test}}, c_i) | c_i \in C]$, and use the trained prediction layer to predict the final label y_{test}.

In the following sections, using Figure 2 as a running example, we describe our specific implementation of each TBM module in terms of how concepts are represented, generated and measured, and how these concept measurements are turned into predictions.

3.2 Concept Representations

Each concept consists of the following components, represented as a JSON object in our prompts:

- **Concept Name**: The name of the concept.
- **Concept Description**: A description of the concept and the factors relevant to measuring it.
- **Concept Question**: The question we use to measure the concept value.
- **Possible Responses**: The set of possible responses to the concept question.
- **Response Guide**: A list of criteria for possible responses, to guide the process of answering the concept question.
- **Response Mapping**: A dictionary mapping each possible response to a numerical score.

Table 1 shows an example representation of the concept "Build Quality" for a product review dataset. The concept question and response guide are important during Concept Measurement stage.

3.3 Concept Generation

At a high level, we generate concepts by prompting an LLM to iteratively discover new concepts that help discriminate between misclassified examples. As outlined by Algorithm 1, given the training set (e.g., restaurant reviews), we initialize the TBM with an empty concept set C. In each iteration, to generate a new concept c, we first identify training examples that have similar representations in the existing concept space but have a high prediction error under the current Prediction Layer. For example, if the current concept space C contains only "Atmosphere" (c_1) and "Food Quality" (c_2), then the two reviews *Great food and ambiance, but quite limited choices on the menu* (3-star) and *Food, atmosphere, variety of choices... everything was excellent!* (5-star) will both be represented as $[+1, +1]$. However, a new concept "Restaurant Variety" can help differentiate between them. Therefore, we construct the concept generation prompt (GeneratePrompt) using the dataset metadata (description and labelling scheme) and these hard examples as in-context exemplars, in order to encourage the generation of a new discriminative concept. To reduce concept duplication, we also include the list of previously generated concepts in this prompt.

Taking GeneratePrompt as input, the LM generates a new candidate concept c, which will then be refined through RefinePrompt. RefinePrompt

Algorithm 1 Iterative Concept Generation

$\mathcal{D}_{\text{train}} \leftarrow$ training samples
$C \leftarrow$ [], list of concepts
initialize TBM **with** $\mathcal{D}_{\text{train}}$, $C = []$.
for i = 1 to N **do**
  / Identify misclassified examples. */*
  $\mathcal{D}_{\text{mis}} \leftarrow \{(t, y) \in \mathcal{D}_{\text{train}} \mid \text{TBM}(t) \neq y\}$
  / Prompt with misclassified examples to generate concept */*
  $c \leftarrow$ **GeneratePrompt**($\mathcal{D}_{\text{mis}}$.sample(), C)
  / Refine the generated concept. */*
  $c' \leftarrow$ **RefinePrompt**(c)
  / Train TBM with updated concepts. */*
  TBM$'$ $\leftarrow$ **train** TBM **with** $\mathcal{D}_{\text{train}}$.sample(), $C + \{c'\}$
  / Admit concept if model is improved. */*
  if TBM$'$.score - TBM.score > γ **then**
    $C \leftarrow C$.append(c')
    TBM $\leftarrow$ TBM$'$
  end if
end for

contains a few examples of problematic concepts, such as those with ambiguous questions and invalid JSON formatting, and how they are fixed. The resulting refined concept, c', along with the existing C, is used to train a new Prediction Layer to create a new candidate TBM$'$. If TBM$'$ outperforms existing TBM on a random subset of $\mathcal{D}_{\text{train}}$ by some threshold γ, it is retained, otherwise, it is omitted. The above procedure is iteratively executed for N cycles, resulting in a final concept set C.[5]

3.4 Concept Measurement

With the generated concept set C, the Concept Measurement module determines the scores $[s(t, c_i)|c_i \in C]$ for any given text t. To measure a concept, we prompt an LLM in a zero-shot fashion to answer the concept question associated with that concept, using the concept description and response guide as context (see Sec 3.2). For instance, consider the concept "Restaurant Variety" in Figure 1. Given a restaurant review, the concept question prompts, "How does the review describe the variety and originality of the restaurant?" The possible answers could be "wide variety", "low variety", "uncertain", or "not applicable". The response given by the LLM is then converted into a numerical concept score using the concept's response mapping (+1 for Positive, -1 for Negative, 0 otherwise). In addition to the categorical answer, the prompt also instructs the LLM to provide relevant snippets in the input text as supporting evidence, for example, "food for everyone, with 4 generations to feed" as supporting snippets for "wide variety".

3.5 Prediction Layer

To combine the concept scores $[s(t, c_i)|c_i \in C]$ into a final prediction y, we train a Prediction Layer on $\mathcal{D}_{\text{train}}$, using linear regression for regression tasks and logistic regression for classification tasks.[6] It learns a weight associated with each concept using y as the supervision signal. For a new input example at inference time, its measured concept scores are multiplied by their weights and summed into the final prediction logit. For example, in Figure 2, across the dataset, "Customer Recommendation" and "Food Quality" are the most important concepts, while "Restaurant Variety" is less crucial. On the given review, "Customer Recommendation" and "Restaurant Variety" are positively scored, but "Atmosphere" and "Value for Money" are negatively scored. Their weighted sum results in a final prediction of 3 stars.

Finally, the concept weights provide a *global* explanation for their relative importance across the dataset, and the concept scores and supporting snippets provide a *local* explanation for the decision on each individual example.

4 Experimental Setup

Implementation Details. We use GPT-4 (`GPT-4-0613`) (OpenAI, 2023) as the underlying LLM for both Concept Generation and Concept Measurement and use Scikit-learn (Pedregosa et al., 2011) to implement linear and logistic regression (with default parameters). See Appendix D for implementation details and prompts.

Datasets. We evaluate on a total of 12 datasets, which we split into 7 "general domain" and 5 "specialized domain" categories. Specialized domain tasks include Fake News Detection (Zhong et al., 2023), News Partisanship Classification (Kiesel et al., 2019), Citation Intent Detection (Cohan et al.,

[5]We set $\gamma =$ and $N = 30$ for all experiments in this paper and the size of the $\mathcal{D}_{\text{train}}$ subset to 100.

[6]We note that at this stage, any interpretable classifier that operates on numerical concept scores, such as decision trees, can be used for the final classification.

Dataset	General domain (classification)				General domain (regression)			Specific domain (classification)				
	Rotten	Amazon	Poem	SNLI	Hate	CEBaB	Yelp	AG	Fake	SciCite	Partisan	Patent
Train Size	250	250	250	250	500	250	250	250	250	250	250	500
Metric	Acc↑	Acc↑	Acc↑	Acc↑	MSE↓	MSE↓	MSE↓	Acc↑	Acc↑	Acc↑	Acc↑	Acc↑
Model (Interptable)												
BERT-base (✗)	0.788	0.872	0.712	0.480	1.868	0.567	0.935	0.904	0.705	0.692	0.784	0.462
DeBERTa-base (✗)	0.824	0.924	0.728	0.512	1.712	0.346	0.539	**0.912**	0.846	0.776	0.776	0.488
GPT-3.5-finetune (✗)	0.916	0.964	**0.820**	0.864	**1.079**	0.300	**0.400**	0.904	**0.950**	**0.764**	**0.852**	**0.604**
GPT-4-10shot (✗)	0.912	**0.980**	0.628	**0.868**	1.666	**0.272**	0.412	0.908	0.842	0.688	0.808	0.474
Naive Bayes (✓)	0.640	0.684	0.604	0.368	2.789	1.173	1.380	0.716	0.419	0.512	0.668	0.282
TBM (Ours) (✓)	**0.924**	0.976	0.796	0.864	1.246	0.431	0.461	0.832	0.776	0.740	0.772	0.450

Table 2: Model performance on 12 datasets. ✗ and ✓denote whether the model is interpretable or not. For each dataset, the highest performance is **bold**, and the second highest is underlined.

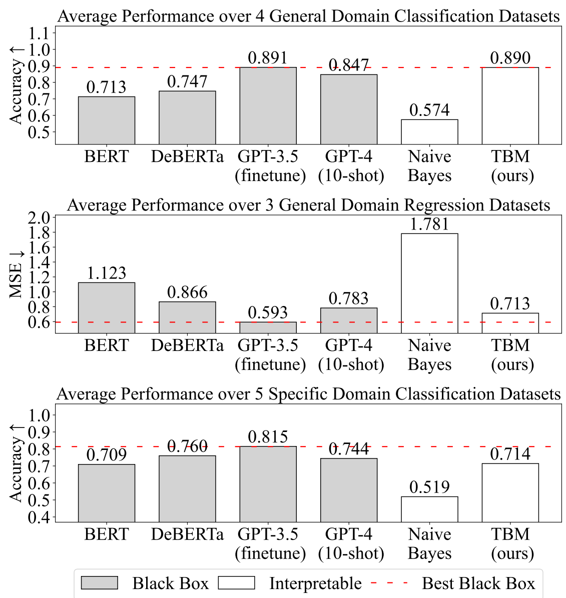

Figure 3: Average performance on 4 General-domain classification datasets (left), 3 General-domain regression datasets (middle) and 5 Specific-domain classification datasets (right).

2019), AG News (Gulli, 2004), Patent Classification (Sharma et al., 2019). General domain tasks include Stanford Natural Language Inference (SNLI) (Bowman et al., 2015), Hate Speech Detection (Kennedy et al., 2020), and five sentiment analysis datasets (Rotten Tomatoes (Pang and Lee, 2005), Amazon reviews (McAuley and Leskovec, 2013), Yelp reviews (Zhang et al., 2015), CEBaB (Abraham et al., 2022), and Poem Sentiment (Sheng and Uthus, 2020)). They differ in that the former requires domain-specific knowledge (mainly in the news and science domain) to solve, whereas the latter can be solved mostly based on common sense and world knowledge. More details of these datasets can be found in Appendix B. Three of these datasets involve a regression task (CEBaB, Yelp, and Hate Speech), while the rest involve classification. With few noted exceptions, we train TBM using 250 examples and test on 250 examples for each dataset considering the expense from API queries.

Baselines. For comparison, we choose the following four baselines:

- **DeBERTa** (He et al., 2021): We finetune a DeBERTa-base[7] classifier for three epochs.
- **BERT** (Devlin et al., 2018): We finetune a bert-base-uncased[8] classifier for three epochs.
- **Naive Bayes** (McCallum et al., 1998): We fit a Naive Bayes classifier on top of the TF-IDF matrix of the texts as an interpretable baseline.
- **GPT-4 (10-shot)** (OpenAI, 2023): We use up to 10 examples[9] with labels to prompt GPT-4 (`gpt-4-0613`).
- **GPT-3.5-turbo** (Peng et al., 2023): We finetune a GPT-3.5-turbo model (`gpt-3.5-turbo-0613`) for three epochs.

Evaluation Metrics. We evaluate TBMs in three ways. First, we compute the end-to-end performance (Mean Squared Error (MSE) for regression and accuracy for classification) compared to the above baselines. Next, we evaluate the Concept Generation and Concept Measurement modules using human annotation (see metrics in Sec 5).

5 Results

5.1 End-to-End Performance

TBMs perform competitively with black-box baselines except for finetuned GPT-3.5. As shown in Figure 3, TBMs achieve the second-highest average accuracy across all sentiment classification datasets (0.89) and the second-lowest average MSE (0.713) across all regression datasets, surpassing all the baselines except for finetuned GPT-3.5. Compared to black-box baselines such as

[7]https://huggingface.co/microsoft/deberta-base

[8]https://huggingface.co/bert-base-uncased

[9]We reduce the number of examples if the maximum context length is reached.

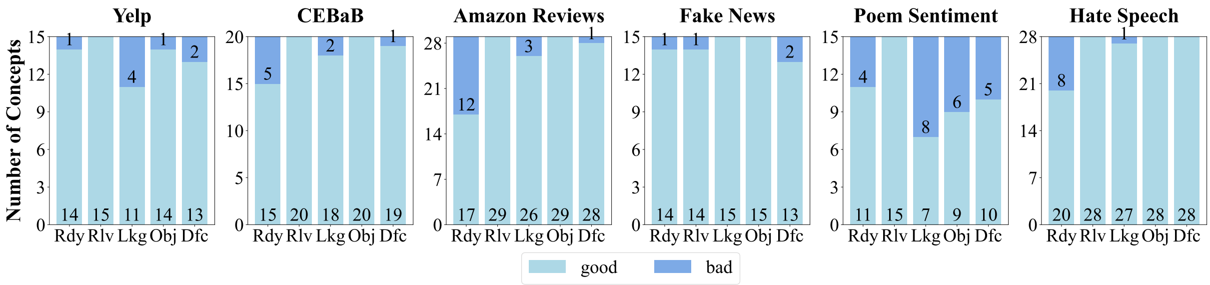

Figure 4: Expert concept annotations for concept generation quality on five aspects: **Redundancy (Rdy)** is concept duplication, "bad" indicates repetition; **Relevance (Rlv)** is pertinence to the task, "bad" identifies spurious concepts; **Leakage (Lkg)** checks if the concept directly performs the task, "bad" indicates leakage; **Objectivity (Obj)** is measurability clarity, with "bad" indicates subjectivity; and **Difficulty (Dfc)** checks the complexity of measuring the concept, "bad" means the concept measurement is harder than dataset task.

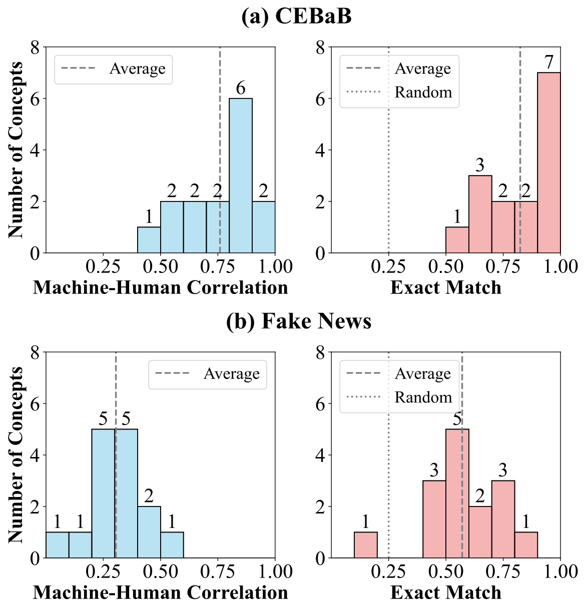

Figure 5: Human evaluation on concept measurement. **Machine-human correlation** measures the Pearson correlation between the concept scores measured by the LLM vs. human annotators. **Exact Match** refers to the performance of the LLM in predicting the exact string label for a concept when using human annotation as gold-standard.

GPT-4 fewshot and finetuned DeBERTa, TBMs exhibit competitive and consistent performance. These results are particularly surprising given that, compared to black-box models, TBMs have access to much less information due to the concept bottleneck while still maintaining the performance. By contrast, the interpretable baseline, Naive Bayes, falls far behind.

Zooming into individual datasets in Table 2, TBMs achieve the highest or second highest performance on 5 datasets. On the remaining 7 datasets, TBMs are visibly outperformed by black-box baselines, but the performance gap tends to be small. On datasets where the TBMs underperform the best model, the average performance difference between TBMs and the best model is 14% (9.6% when excluding finetuned GPT-3.5). This gap shrinks for classification tasks on sentiment (Rotten Tomatoes, Amazon Reviews, Poem Sentiment), where the average performance gap is 1.4%, indicating minimal interpretability-performance tradeoffs for this domain.

TBMs excel in general-domain texts but struggle for domain-specific texts. After further examining the results in different domains, we observe that TBMs perform well on the general domain tasks, including sentiment comprehension and Natural Language Inference. However, it falls behind in specialized domain tasks, including those in the news and science domain. Below, we compare the performance of TBMs against all other baselines excluding finetuned GPT-3.5 to allow for a cleaner comparison with these established baselines.

Among all 7 general domain datasets, compared to all baselines except finetuned GPT-3.5, TBMs achieve the best performance on 3 of them (Rotten Tomatoes, Poem Sentiment, and Hate Speech). On Amazon Reviews and SNLI, TBMs closely match the best baseline, with an accuracy difference $\leq$ 0.004. The only exception is CEBaB and Yelp, where TBMs are outperformed by a large margin (relative difference of 37% and 10%), which we have not understood well. Overall, we hypothesize that the encouraging performance in these tasks is potentially because they do not require domain-specific knowledge, making it easier for LMs to discover concepts by relying on knowledge learned during pretraining.

In specialized domains such as news (Fake News, AG News, News Partisanship) and science (Patent, SciCite), TBMs are consistently outperformed by either GPT-4 fewshot or finetuned DeBERTa. We postulate that this can be attributed to the fact that it is more challenging for LMs to discover relevant concepts for these tasks in a zero-shot fashion, without domain-specific knowledge. Another factor can be that certain generated concepts in specialized domains, such as "Fact Checking" in detecting fake news, are as difficult to measure as the target label. Therefore, it is challenging for the Concept Measurement module to assess the concept value in a zero-shot manner accurately. This hypothesis is further investigated in Sec 5.2. Other potential factors remain to be explored in the future.

Overall, all the above results demonstrate that TBMs are competitive with GPT-4 fewshot and finetuned DeBERTa on average, with exceptional performance on sentiment classification and NLI tasks, but still have room for improvement in domains such as news and science. To further understand where the error comes from, we manually evaluate each module in the TBM pipeline in the next two subsections.

5.2 Concept Generation Module Evaluation

To assess the Concept Generation module, we manually evaluate the generated concepts in 6 aspects: *Redundancy*, *Relevance*, *Leakage*, *Objectivity*, and *Difficulty*, each explained in the caption of Figure 4. Three annotators, who are all authors of this paper, perform this evaluation for each concept on six datasets with conflicts resolved by a simple majority vote.

According to Figure 4, across all datasets, the overwhelming majority of concepts are of high quality, except Poem Sentiment. On average, Redundancy emerges as the most common issue (25%), followed by Leakage (15%), while the other issues, including difficulty (9%), objectivity (6%) and relevance (1%), are less frequent. This suggests that the module has almost no problem discovering concepts that are relevant to the task label and can mostly ensure that the concepts are unambiguous and easy to measure. However, the Concept Generation Module occasionally accepts unnecessary concepts that are too similar to previously generated ones or concepts that directly leak the task label. The prevalence of these issues varies across datasets. For instance, Poem Sentiment shows high concept error rates in almost all aspects except relevance, while Hate Speech concepts have mostly redundancy issues.

Redundant concepts unnecessarily increase the size of the concept space, which can increase the cognitive load of users trying to interpret the model behavior. Leaky concepts can undermine the faithfulness of provided explanations, making the "self-explanatory" claim invalid. To mitigate these issues, we are exploring other heuristics to filter problematic concepts during generation, in addition to the performance improvement threshold.

5.3 Concept Measurement Module Evaluation

To determine whether the Concept Measurement Module measures concepts correctly, we compare the concept scores rated by the LLM with those rated by humans on the CEBaB and Fake News datasets. We asked a group of crowdworkers[10] to answer the questions generated by the model for each concept, with the concept description and response guide as additional context. This is the same information that the LLM receives when performing Concept Measurement. We compute the exact match and correlation between the human and LLM judgments. If annotators do not have a clear majority decision for an instance, it is labeled as "uncertain".

Figure 5 (a) shows the histogram of the correlations and accuracies for all the concepts in the CEBaB dataset. We see that the TBM can measure a majority of the concepts it generates accurately: the median correlation and accuracy are high at 0.814 and 0.893 respectively, with the average being 0.759 for correlation and 0.824 for accuracy. This level of agreement is remarkable since concept measurement is done in a zero-shot manner, with no training data about the specific concept being measured.

In contrast, the performance for the Fake News dataset, as shown in Figure 5 (b), is modest: the median correlation and accuracy are 0.317 and 0.549, respectively, while the average scores are 0.305 for correlation and 0.571 for accuracy. Most of this reduced performance comes from hard-to-measure concepts such as "Fact Checking", where the LLM asserts that a text can be fact-checked despite no access to external resources.

Meanwhile, this stark difference in performance between the two datasets reflects the transparency and auditability of TBMs. The exemplary performance on the CEBaB dataset validates the potential

[10]More details in Sec C.2

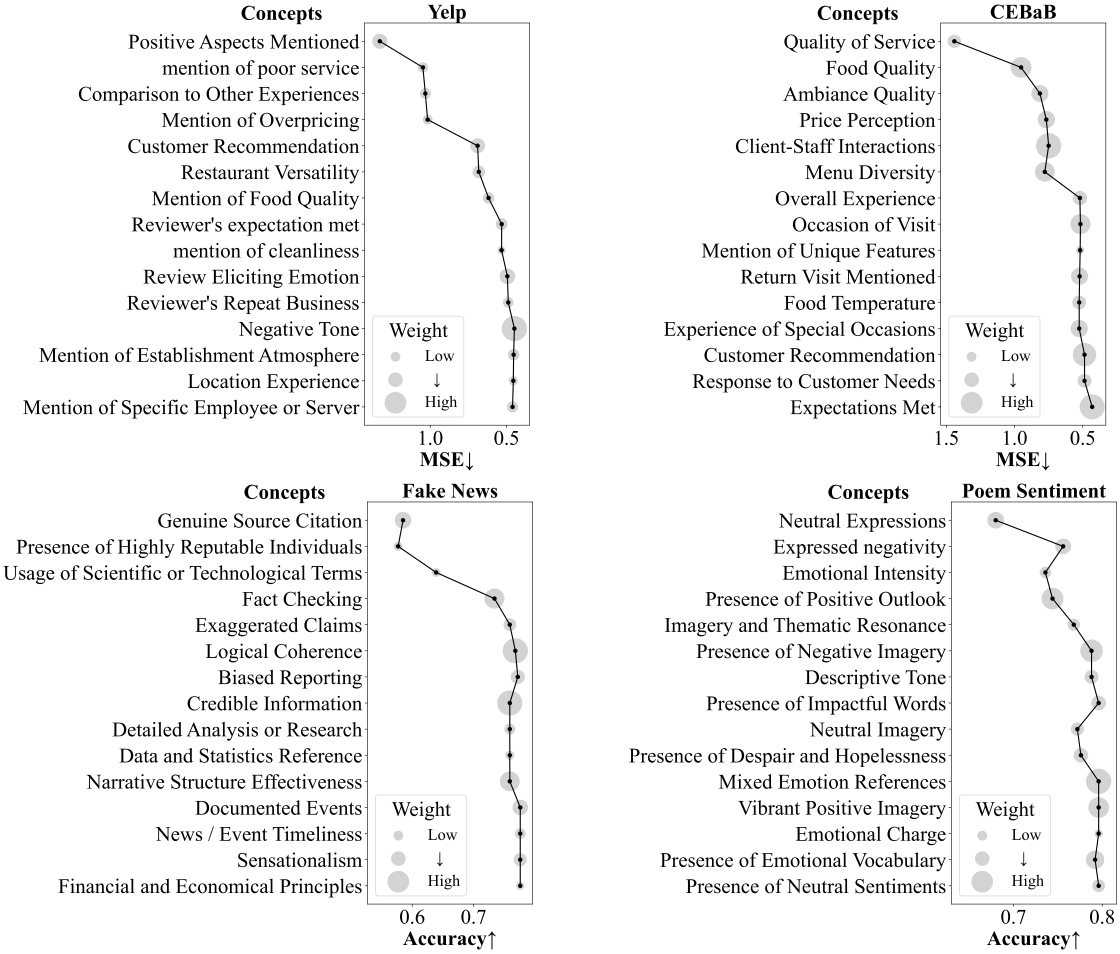

Figure 6: Concept learning curves of TBMs on 6 datasets. The x-axis represents the TBM's performance (MSE for regression task and Accuracy for classification tasks) at each iteration, and the y-axis indicates the specific concept added to the bottleneck during that iteration. The size of each node is determined by the magnitude of the weight of the corresponding concept in the prediction layer.

and effectiveness of this module. Simultaneously, the suboptimal results on Fake News provide clear signals for potential pitfalls that require human intervention and debugging.

6 Analysis of Learning Curves

One unique advantage of TBMs is that their interpretable structure allows users to analyze model behavior in a more granular way compared to black-box models. We demonstrate this by plotting the *concept learning curves* of TBMs on four of our datasets in Figure 6.[11] These learning curves show how the TBM's performance changes on the test data as it iteratively generates new concepts. The figure also shows the importance of each concept, which is calculated as the absolute value of its weight learned by the final prediction layer.[12]

These curves make it easy to see the system's performance change as it progressively adds new concepts to the bottleneck allowing users to directly identify the most helpful concepts. For example, in the Yelp dataset, the introduction of the "Customer Recommendation" concept led to a significant drop in MSE. However, the concept "Emotional Intensity" appears less informative in the poem sentiment task, as the accuracy decreases after it is added to the bottleneck.

Interestingly, we also observe that the most important concepts in terms of weight are not always discovered immediately. Instead, they can still show up at later stages of iteration. For example, in CEBaB, "Expectations Met" has one of the highest importance weights but is discovered last.

[11] We only show 4 datasets here due to space limits. See Appendix A.3 for the learning curves on other datasets.

[12] For linear regression, this is the magnitude of regression weight associated with the concept. For logistic regression, this is the average absolute value of the concept weight across all classes.

These learning curves can be contrasted with the learning curves in black-box models, where sudden increases in model performance require in-depth investigation to identify the cause of improvement. We include two additional examples of how this added interpretability can be useful in our Appendix. Appendix A.1 shows an analysis of how the performance of different training runs on the same dataset can be explained using the discovered concepts, and Appendix A.2 shows how we can explain the overfitting of our TBM on a small dataset based on the discovery of spurious concepts.

7 Discussion

In this section, we discuss some additional benefits of using TBMs.

TBMs allow users to intuitively interact with the concept bottleneck Since concepts are fully represented in natural language, practitioners can easily add or delete concepts in the concept space without relying on an LLM. This allows experts to directly inject domain-specific inductive bias at a high level of abstraction. Additionally, they can tweak how concepts are measured by simply rewriting the instructions in the Concept Measurement prompt. For example, if a practitioner wants to increase the granularity of the concept "Noise" which currently has two options "noisy" or "not noisy", they can edit the responses to add options such as "moderately noisy" and "unbearably noisy". These interactions make it easier to steer the behavior of TBMs compared to black box models.

TBMs can be used to characterize domain shifts. In addition to greater applicability in high-stakes domains, TBMs provide a more grounded handle that we can use to characterize domain shifts. For example, a hypothetical TBM fit on predicting popular movies on dataset A can be re-fit on dataset B using the same concepts. If dataset A contains reviews from expert critics and dataset B contains reviews from casual fans, the difference in reviewer tendencies can be described based on the shifts in the concept weights. For instance, "character quality" may have a higher weight in dataset A compared to dataset B, indicating that expert critics might have placed a greater emphasis on well-written characters.

8 Conclusion

In this paper, we present Text Bottleneck Models (TBMs)—an innovative text classification framework that is interpretable by construction. TBMs provide both global and local interpretability with sparse concept-level explanations, allowing users to understand the general principles being used for inference as well as the specific reasoning for individual examples. TBMs can be fully automated, requiring no human-curated concept set. In our evaluation, we show that TBMs achieve competitive performance against strong black-box baselines such as GPT-4 fewshot and finetuned DeBERTA across 12 diverse text regression and classification datasets despite being constrained by an information bottleneck. Human evaluations reveal that the concepts generated by the system are mostly relevant and objective, but there still exist issues in redundancy and leakage. Overall, we demonstrate that TBMs are a promising general architecture to construct a highly interpretable predictor with minimal performance trade-offs for general-domain text.

9 Limitations

Scalability. Given the heavy reliance of the model on large language models for concept measurement, the current implementation is not very scalable. This is because, for every text we want to measure, the number of times we have to run inference on an LLM is equal to the number of concepts in the bottleneck. To improve the scalability of the system, it may be possible to finetune smaller language models that perform the concept measurement after the TBM has scored enough texts. This will reduce the number of LLM calls to only scale with regard to the number of concepts, rather than scaling in proportion with both the number of texts and the number of concepts.

Redundant and Leaky Concepts. The analysis of generated concepts reveals the existence of duplicate concepts and concepts that leak classification labels in some datasets. To mitigate these issues, future work can include steps in concept generation to filter problematic concepts since currently we only filter concepts that do not improve model performance.

10 Ethics Statement

Potential risks We note that despite being designed to be more interpretable, the system we present in this paper still relies fundamentally on LLMs whose outputs may be unpredictable or unsafe. Accordingly, a proper deployment of our system would require safeguards to reduce the risk of harm if present.

Acknowledgements

This research is based upon work supported in part by the Air Force Research Laboratory (contract FA8750-23-C-0507), the DARPA KAIROS Program (contract FA8750-19-2-1004), the IARPA HIATUS Program (contract 2022-22072200005), and the NSF (Award 1928631). Approved for Public Release, Distribution Unlimited. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of AFRL, DARPA, IARPA, NSF, or the U.S. Government.

We would also like to thank Matthew Pressimone and Saurabh Shah for their help in assessing the initial feasibility of the project.

References

Eldar David Abraham, Karel D'Oosterlinck, Amir Feder, Yair Ori Gat, Atticus Geiger, Christopher Potts, Roi Reichart, and Zhengxuan Wu. 2022. Cebab: Estimating the causal effects of real-world concepts on nlp model behavior.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. *arXiv preprint arXiv:1905.08160*.

Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José MF Moura, and Peter Eckersley. 2020. Explainable machine learning in deployment. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 648–657.

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. Language models can explain neurons in language models. *URL https://openaipublic. blob. core. windows. net/neuron-explainer/paper/index. html.(Date accessed: 14.05. 2023)*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

Zhoujun Cheng, Jungo Kasai, and Tao Yu. 2023. Batch prompting: Efficient inference with large language model apis. *arXiv preprint arXiv:2301.08721*.

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. 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 3586–3596, Minneapolis, Minnesota. Association for Computational Linguistics.

Katherine Maeve Collins, Matthew Barker, Mateo Espinosa Zarlenga, Naveen Raman, Umang Bhatt, Mateja Jamnik, Ilia Sucholutsky, Adrian Weller, and Krishnamurthy Dvijotham. 2023. Human uncertainty in concept-based ai systems. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*, pages 869–889.

Anubrata Das, Chitrank Gupta, Venelin Kovatchev, Matthew Lease, and Junyi Jessy Li. 2022. Prototex: Explaining model decisions with prototype tensors. *arXiv preprint arXiv:2204.05426*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Mengnan Du, Ninghao Liu, and Xia Hu. 2019. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1):68–77.

Rudresh Dwivedi, Devam Dave, Het Naik, Smiti Singhal, Rana Omer, Pankesh Patel, Bin Qian, Zhenyu Wen, Tejal Shah, Graham Morgan, et al. 2023. Explainable ai (xai): Core ideas, techniques, and solutions. *ACM Computing Surveys*, 55(9):1–33.

Antonio Gulli. 2004. Ag's corpus of news articles.

Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5553–5563, Online. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

Chris J Kennedy, Geoff Bacon, Alexander Sahn, and Claudia von Vacano. 2020. Constructing interval variables via faceted rasch measurement and multitask deep learning: a hate speech application. *arXiv preprint arXiv:2009.10277*.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. SemEval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 829–839, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *Proceedings of the 37th International Conference on Machine Learning*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*.

Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. 2020. Explainable ai: A review of machine learning interpretability methods. *Entropy*, 23(1):18.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2022. Towards faithful model explanation in nlp: A survey. *arXiv preprint arXiv:2209.11326*.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2022. Post-hoc interpretability for neural nlp: A survey. *ACM Computing Surveys*, 55(8):1–42.

Anita Mahinpei, Justin Clark, Isaac Lage, Finale Doshi-Velez, and Weiwei Pan. 2021. Promises and pitfalls of black-box concept learning models. *arXiv preprint arXiv:2106.13314*.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172.

Andrew McCallum, Kamal Nigam, et al. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. Madison, WI.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

OpenAI. 2023. Gpt-4 technical report.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Andrew Peng, Michael Wu, John Allard, Logan Kilpatrick, and Steven Heidel. 2023. Gpt-3.5 turbo fine-tuning and api updates.

Sarah Pratt, Ian Covert, Rosanne Liu, and Ali Farhadi. 2023. What does a platypus look like? generating customized prompts for zero-shot image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15691–15701.

Dheeraj Rajagopal, Vidhisha Balachandran, Eduard H Hovy, and Yulia Tsvetkov. 2021. SELFEXPLAIN: A self-explaining architecture for neural text classifiers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 836–850, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Vikram V Ramaswamy, Sunnie SY Kim, Ruth Fong, and Olga Russakovsky. 2022. Overlooked factors in concept-based explanations: Dataset choice, concept salience, and human capability. *arXiv preprint arXiv:2207.09615*.

Eva Sharma, Chen Li, and Lu Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. *arXiv preprint arXiv:1906.03741*.

Emily Sheng and David Uthus. 2020. Investigating societal biases in a poetry composition system.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Zhengxuan Wu, Karel D'Oosterlinck, Atticus Geiger, Amir Zur, and Christopher Potts. 2023. Causal proxy models for concept-based model explanations. In *International Conference on Machine Learning*, pages 37313–37334. PMLR.

Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. 2023. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19187–19197.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Ruiqi Zhong, Peter Zhang, Steve Li, Jinwoo Ahn, Dan Klein, and Jacob Steinhardt. 2023. Goal driven discovery of distributional differences via language descriptions. *arXiv preprint arXiv:2302.14233*.

A Further Analysis

A.1 Does the model generate similar concepts across repeated runs?

To evaluate the variance in concept generation, we compare the concepts generated by a TBM on five runs on the CeBaB dataset. Figure 9 visualizes the concepts generated across model runs and Figure 10 shows the learning curves for the TBMs. We can see that concepts such as “Menu variety”, “Food Quality”, “Reviewer Expectations”, “Ambiance Quality”, “Service Quality” are generated by the majority of the TBMs. Inspecting the concept learning curves also reveals that these concepts tend to be highly important in the model. Overall, these results indicate that TBMs can consistently discover the important concepts across replicated model runs.

The MSE for the final models are 0.43, 0.48, 0.29, 0.36, and 0.48 respectively. We note that both models that achieve the best performance contain concepts that leak the task label, such as “Dining Experience” and “Overall Restaurant Quality”.

A.2 Can TBMs work on small training sets?

To evaluate the effect of training on a small dataset, we train three TBMs on the CEBaB dataset after limiting the size of the training set to 50 examples. The learning curves for these TBMs can be see in figure 8. In this figure, we see that it is possible for TBMs to overfit after generating too many concepts. Across all three runs, we see that performance increases until we reach around 5 concepts, and afterwards, it starts to drop. This drop can be explained by the fact that the TBM starts to admit concepts that are too specific, for instance, in the third model we see a "Menu Misrepresentation" concept which does not exist in any of the five full-size TBM replications. Another explanation for this drop is that there is not enough information to determine the importance of concepts relative to one another. Thus, even if the correct concepts are generated, the weights assigned to them under low training size samples can be unstable and generalize poorly outside of the training distribution.

A.3 Learning Curves on all datasets

Figure 7 shows the concept learning curves on all datasets, in addition to the four reported in Section 6.

B Dataset Details

Table 3 shows the dataset description, possible labels, and an example from the dataset. Among all datasets, Yelp Reviews, CEBaB, and Hate Speech Detection involve a regression task, while others involve a classification task.

C Human Evaluation for Concept Generation and Measurement

C.1 Concept Generation

The authors used the following guide to annotate each concept’s quality in Sec E.3. Quality scores equal to 1 indicate no problems while quality scores greater than 1 indicate issues.

Evaluation Metrics. For evaluation, we use the following metrics:
Redundancy (Rdy): 1 - No issues; 2 - Given the rest of the concepts already generated, this concept is redundant.
Relevance (Rlv): 1 - This concept is related to the task; 2 - This concept is unrelated to the task.
Leakage (Lkg): 1 - This concept does not leak the labelling task; 2 - This concept leaks the labelling task.
Objectivity (Obj): 1 - This concept can be measured objectively; 2 - This concept is subjective.
Difficulty (Dfc): 1 - Answering this concept question is easier than the labelling task; 2 - Answering this question is around the same difficulty as the labelling task; 3 - This question is harder than the labelling task.

C.2 Concept Measurement

To evaluate the TBM’s performance on concept measurement, we generate a questionnaire for each dataset and ask human crowdworkers to measure the scores of TBM-generated concepts. To avoid cases where the questions do not have an answer, we insert a “None of the Above” response at the end. Our annotators are students from a graduate-level AI class at the University of Pennsylvania, with good English proficiency. Both tasks are given as optional extra credit assignments in the class. Participation is solely voluntary. Before participation, students can preview the tasks, and are given a clear description of how the data will be used at the beginning of the instructions. The population size is 98.

Dataset (Citation)	Description with Possible Labels	Example with Label
Fake News Detection (Zhong et al., 2023)	A dataset containing real and fake news from different publishers. **Possible labels**: Fake, Real	Brazil qualify for 2018 World Cup after Coutinho and Neymar down Paraguay Brazil... **Label**: Real
Yelp Reviews (Zhang et al., 2015)	This dataset contains user-written Yelp reviews. The goal is to predict the review rating (1 to 5 stars) based on the text of the review. **Possible labels**: 1 Star, 2 Stars, 3 Stars, 4 Stars, 5 Stars	OMG. The best authentic Mexican food. Spicy - yes. **Label**: 4 Stars
Poem Sentiment (Sheng and Uthus, 2020)	This dataset contains verses of poems with their sentiment labels. The goal is to predict the sentiment of a verse based on its text. **Possible labels**: Negative, Positive, No Impact, Mixed	and say, 'fie, pale-face! are you english girls **Label**: No Impact
Rotten Tomatoes (Pang and Lee, 2005)	This dataset contains movie reviews from Rotten Tomatoes. The goal is to predict the binary sentiment of a review based on its text. **Possible labels**: Negative, Positive	the performers are so spot on, it is hard to conceive anyone else in their roles. **Label**: Positive
Stanford Natural Language Inference (SNLI) (Bowman et al., 2015)	This dataset contains pairs of sentences (premise and hypothesis). The goal is to predict the relationship between the premise and hypothesis. **Possible labels**: Entailment, Neutral, Contradiction	Premise: A man giving a speech for the student financial administrators. Hypothesis: There is a man in this picture **Label**: Neutral
AG News (Gulli, 2004)	A dataset containing news of various categories. The goal is to predict the category based on the news text. **Possible labels**: Business, Science/Technology, Sports, World/Political	4 studios back Toshiba HD DVD TOKYO... **Label**: Business
Amazon Reviews (McAuley and Leskovec, 2013)	This dataset contains product reviews from Amazon. The goal is to predict the binary sentiment of a review based on its text. **Possible labels**: Negative, Positive	Someone recommended this product to me - it keeps my floors cleaner longer... **Label**: Positive
CEBaB (Abraham et al., 2022)	Restaurant reviews from Opentable. **Possible labels**: 1 Star, 2 Stars, 3 Stars, 4 Stars, 5 Stars	Very poor service and food, this is a second try for this restaurant... **Label**: 1 Star
News Partisanship (Kiesel et al., 2019)	Hyperpartisan News Detection for PAN @ SemEval 2019 Task 4. Given a news article text, decide whether it follows a hyperpartisan argumentation. **Possible labels**: Not Hyperpartisan, Hyperpartisan	title: Trump Must Now Be Compelled to Withdraw text:This is now bigger than who becomes the next president. Trump is a threat to our democracy. **Label**: Hyperpartisan
Citation Intent (Cohan et al., 2019)	A dataset for classifying citation intents in academic papers into method, background, or result. **Possible labels**: Method, Background, Result	However, the k-safeness of the hypercube does not guarantee the connectivity of the network unless we also bound the number of faulty nodes by 2(n - k) - 1 [17, 35]. **Label**: Background
Patent Classification (Sharma et al., 2019)	A Patent Classification Dataset classifying patents into various categories. **Possible labels**: Human Necessities, Performing Operations; Transporting, Chemistry; Metallurgy, Textiles; Paper, Fixed Constructions, Mechanical Engineering; Lightning; Heating; Weapons; Blasting, Physics, Electricity, General tagging of new or cross-sectional technology	a display device has a measuring circuit to detect flicker due to the presence of a dc voltage by monitoring the pixel voltage and , if necessary , modifying driving signals. **Label**: Physics
Hate Speech Detection (Kennedy et al., 2020)	Online comments with an associated hate speech score. The measure is continuous for hate speech, where higher values indicate more hateful content, and lower values indicate less hateful or supportive speech. A score greater than 0.5 is approximately hate speech, a score less than -1 is counter or supportive speech, and scores between -1 and +0.5 are neutral or ambiguous. **Possible labels**: Continuous score	I never saw the privilege of being gay until i just saw a guy grab 8 boxes of pregnancy tests at the dollar store **Label**: -1.48

Table 3: Summary of Datasets.

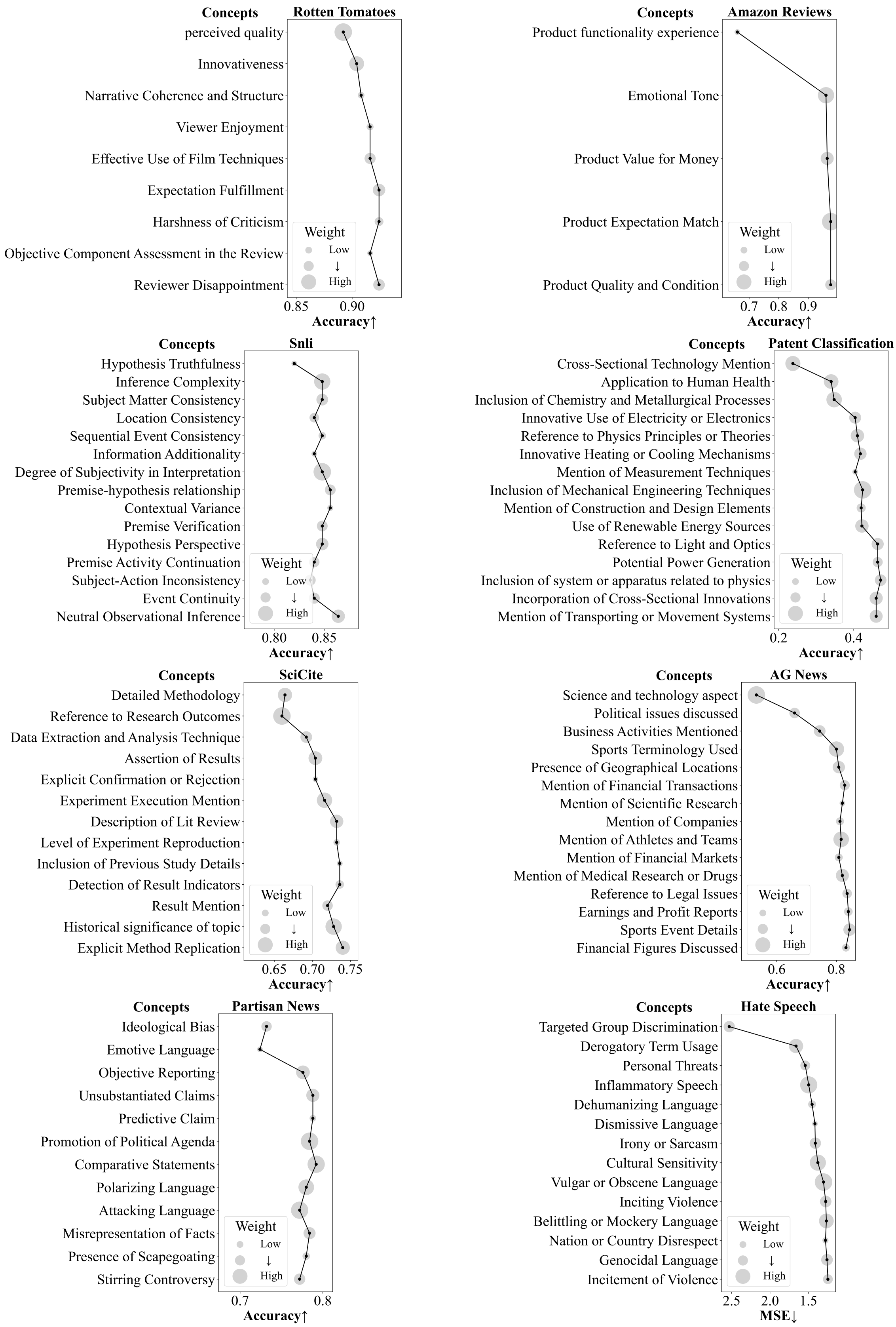

Figure 7: Concept learning curves on all datasets.

We design an Amazon Mechanical Turk interface for the task, which can be found in the Supplementary Materials (Fig 11). With an hour of work, students can earn 1% in extra credit of the overall

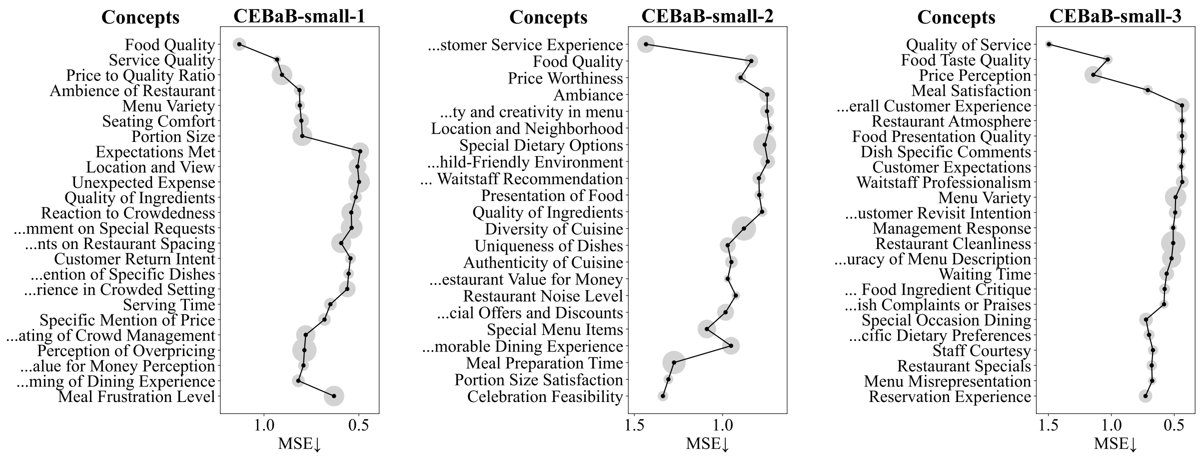

Figure 8: Concept learning curves on small CEBaB datasets with different runs.

course grade.

D Additional Details of Implementation

Concept Generation

D.1 Prompt Structure

The prompt contains three main sections: The instruction set, dataset information, and TBM state information. The instruction set contains details about what the concept generation task is, what the format of a concept is, and three examples of valid concepts for toxicity detection, product sentiment analysis, and scam detection. This is followed by dataset information which is where we insert the dataset information, label descriptions, and examples from the dataset with different labels. Finally, to avoid making duplicate concepts, we load the list of previously generated concepts at the end in the TBM state information section.

D.2 Selecting in-context examples

We selectively load in highly misclassified examples during the concept generation stage to increase the chances that we generate concepts that are relevant for these misclassified examples. We get these examples by examining the 10 nearest neighbors for each training example under the current concept feature space and then obtaining 20 examples with the highest "neighborhood loss" which is obtained averaging each neighborhood's MSE (for regression) or accuracy (for classification). We then check to see if this set of examples exceeds the token limit. If it does, we iteratively remove an example with the most common label within the group to ensure diverse representation. If we end up with less than 4 examples, we restart the process but truncate the texts by a factor of 0.8. We note that when the TBM generates its first concept, this sampling is random.

D.3 Managing Token Limits

The number of iterations we can perform is bounded primarily by the token limit of the LLM that we are using. As the number of iterations increases, the length of the TBM state information grows and at some point concept generation fails due to exceeding token limits. It is possible to truncate this step but that can cause issues with redundant concepts. To help manage token limits in other parts of the prompt, we dynamically truncate the text examples loaded in to ensure that the prompt stays within the token budget of the LLM being used.

Concept Measurement

This module relies heavily on the concept question and response guide associated with each concept to function. This module is flexible, allowing various prompting methods, such as directly answering the question or chain-of-thought prompting. In this paper, we structure the prompt as a three-step process that involves extracting pertinent snippets from the text and reasoning over them before yielding a final answer. The prompt takes in as input the text we want to measure along with the JSON of the concept being evaluated. The prompt returns a JSON object representing the salient snippets in the text for each possible classification, the reasoning of the model over those snippets, and then the final classification. We perform batch inference (Cheng et al., 2023) to reduce LLM costs. In cases where the generated text fails to parse as valid JSON or does not contain text we can turn

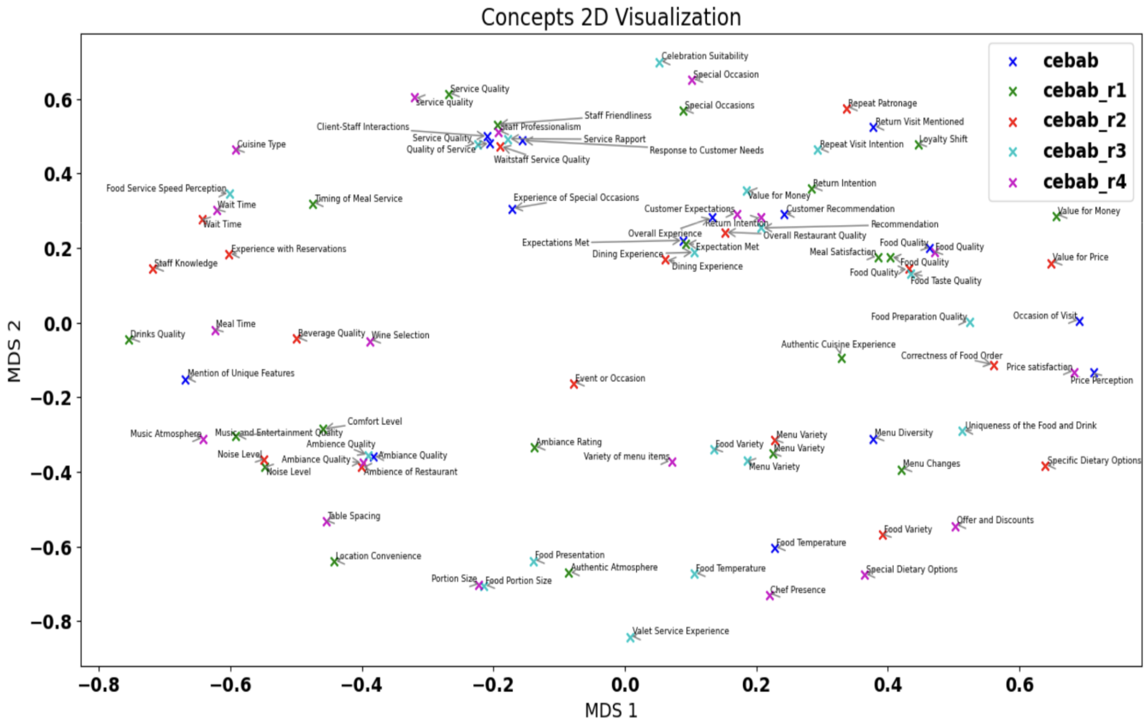

Figure 9: Concepts across multiple CEBaB training runs, as visualized in Multi-Dimensional Scaling (MDS) Plot

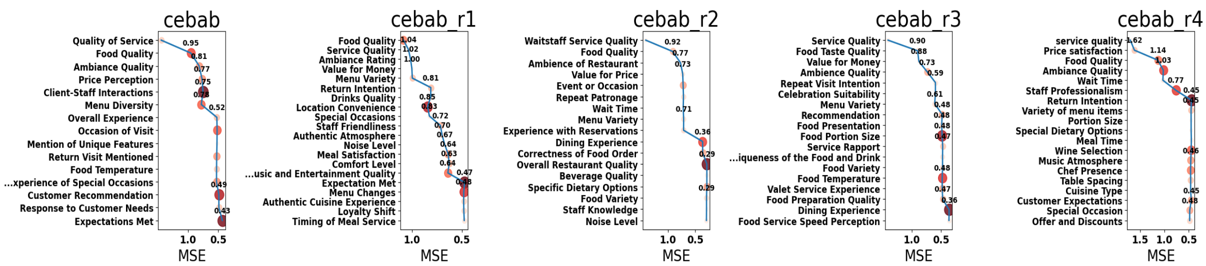

Figure 10: Concepts across multiple CEBaB training runs, as visualized in learning curves.

into a score using the response mapping, we return a concept score of 0.

E Prompts

E.1 Concept Generation Prompt

```
Concept Feature Engineering Task

Below we are given a text dataset with accompanying labels.
      Our task is to identify a concept in the text that
      could be associated with the label. This is because we
      want to find the main factors that can be used to
      explain the label.

To do this, we will examine a sample of texts that have
      different labels so that we can look at the different
      characteristics that exist for one label and compare it
       to another. Good concepts are those that separate
      texts with one label from another.

After looking at these texts and finding a difference, we
      will define a concept definition JSON
Each full concept definition comes with a concept name,
      description, question, response set, and response guide
      . The concept description provides an intuitive
      overview of the concept. The concept question is our
      tool for measuring the concept, this will be graded by
      a human annotator. The possible responses list the
      possible responses to the question and the response
      guide provides information on what each rating means.
      We also include a response mapping to help with data
      processing.

Below are some examples of concepts for different datasets.
      Note that while we can have any number of possible
      responses, the concept should be designed for a
      positive/negative/uncertain question format that maps
      to 1, -1, and 0 respectively.

1. A possible concept for a dataset assigning toxicity
      scores to social media texts
{"Concept Name": "explicit language",
"Concept Description": "'Explicit language' refers to the
      use of words, phrases, or expressions that are
      offensive, vulgar, or inappropriate for general
      audiences. This may include profanity, obscenities,
      slurs, sexually explicit or lewd language, and
      derogatory or discriminatory terms targeted at specific
       groups or individuals.",
"Concept Question": "What is the nature of the language used
       in the text?",
"Possible Responses": ["explicit", "non-explicit", "
      uncertain"],
"Response Guide": {
"explicit": "The text contains explicit language, such as
      profanity, obscenities, slurs, sexually explicit or
      lewd language, or derogatory terms targeted at specific
       groups or individuals.",
"non-explicit": "The text is free from explicit language and
       is appropriate for general audiences.",
"uncertain": "It is difficult to determine the nature of the
       language used in the text or if any explicit terms are
       used."
},
"Response Mapping": {
"explicit": 1,
"non-explicit": -1,
"uncertain": 0
}
}###

2. A possible concept for evaluating the sentiment of
      product reviews on ecommerce site
{"Concept Name": "good build quality",
"Concept Description": "Build quality refers to the
      craftsmanship, durability, and overall construction of
      a product. It encompasses aspects such as materials
      used, design, manufacturing techniques, and attention
      to detail. A product with good build quality is
      typically considered to be well-made, sturdy, and long-
      lasting, while a product with poor build quality may be
       prone to defects or wear out quickly.",
"Concept Question": "What does the review say about the
      build quality of the product?",
"Possible Responses": ["positive", "negative", "uncertain",
      "not applicable"],
"Response Guide": {
"positive": "Review mentions aspects such as well-made,
      sturdy, durable, high-quality materials, excellent
      craftsmanship, etc.",
"negative": "Review mentions aspects such as poor
      construction, flimsy, cheap materials, bad design,
      easily breakable, etc.",
"uncertain": "Review does not mention build quality, the
      information is ambiguous or vague, or it has both
      positive and negative aspects mentioned like 'the
      product is sturdy but uses cheap materials'.",
"not applicable": "The review does not mention the build
      quality of the product at all."
},
"Response Mapping": {
"positive": 1,
"negative": -1,
"uncertain": 0,
"not applicable": 0
}
}###

3. A useful concept for scam detection for emails
{
"Concept Name": "Extremely generous offer",
"Concept Description": "The concept 'Extremely generous
      offer' refers to situations where the text describes an
       offer that seems too good to be true, such as promises
       of large financial gains, disproportionate rewards, or
       substantial benefits with seemingly little to no risk
      or effort required. These can often be indicative of
      scams or deceptive practices.",
"Concept Question": "What type of offer is described in the
      text?",
"Possible Responses": ["extremely generous offer", "ordinary
       offer", "no offer", "uncertain"],
"Response Guide": {
"extremely generous offer": "The text describes an offer
      that is disproportionately rewarding or beneficial with
       seemingly little to no risk or effort. This could
      include promises of large financial returns with
      minimal investment, 'free' gifts that require payment
      information, or rewards that are disproportionate to
      the effort required.",
"ordinary offer": "The text describes a typical or ordinary
      offer. For instance, normal sales or discounts,
      standard business offerings, or fair trades.",
"no offer": "The text does not describe any offer.",
"uncertain": "It is difficult to determine the type of offer
       described in the text. The text might be vague,
      ambiguous, or lack sufficient context."
},
"Response Mapping": {
"extremely generous offer": 1,
"ordinary offer": -1,
"no offer": -1,
"uncertain": 0
}
}

---

In the task, we will generate concepts for the fake_news
      dataset

Below is an explanation of the dataset and the labels
      therein:

A dataset containing real and fake news from different
      publishers

{'0': 'fake', '1': 'real'}


Some additional pointers to keep in mind are the following:
1. In this exercise, we will restrict ourselves to making
      positive/negative/uncertain questions
2. Design the response guide so that the responses are both
      mutually exclusive and collectively exhaustive.

Below are some example texts along with their labels.
---
text:Wells Fargo profits spike despite legal costs "Despite
      the looming court costs of its recent scandal, Wells
      Fargo bank has reported an increase in quarterly
      profits. Third quarter profit rose 2% to $6bn, up from
      $5.8bn last year. In response, the bank is hiring for
      positions which were previously cut when employees
      involved in the scandal were fired. The scandal
      involved employees opening up fake accounts in
      customers' names without the customers' knowledge. The
      latest report of profit increases has surprised many in
       and outside of the bank."
rating: fake
text:Shailene Woodley to lead hunger strike in jail over
      pipeline US actress Shailene Woodley has announced to
      fellow inmates and guards of a North Dakota jail that
      she will lead a hunger strike against the Dakota Access
       oil pipeline. Woodley is the start of the Divergent
      Series. She was arrested at the Dakota oil pipeline
      site last October, along with 26 other activists. Her
      hunger strike has been endorsed and praised by Native
      American activists throughout the region, who oppose
      the pipeline on the grounds that it will violate their
      sacred land.
rating: fake
text:Toshiba's Westinghouse files for US bankruptcy
      Westinghouse Toshiba's US nuclear unit has filed for US
```

```
bankruptcy protection. The US firm has struggled with hefty losses that have thrown its Japanese parent into a crisis putting the conglomerate's future at risk. Westinghouse has suffered huge cost overruns at two US projects in Georgia and South Carolina. Toshiba said the bankruptcy would not affect Westinghouse's UK operation which employs more than 1 000 workers. However the firm warned that the writedown of its US nuclear business could see Toshiba's total losses last year exceed 1 trillion yen ($9.1bn; 7.3bn) almost triple its previous estimate. The Japanese government confirmed on Wednesday that it was aware of Toshiba's plans.
rating: real
text:George Michael portrait by Damien Hirst sells for $580 000 "A portrait of the late George Michael by artist Damien Hirst has sold for just under half a million pounds at a charity auction. The money raised from the sale of Beautiful Beautiful George Michael Love Painting will go to HIV/Aids charity The Goss-Michael Foundation. The charity was founded by Michael and his former partner Kenny Goss. Goss posted an image of the artwork on Instagram writing: "Amazing result of $580 000 (around 461 011)." He described Damien Hirst as a "superstar" adding: "Thank you Damien!" The canvas went under the hammer in Dallas Texas at the MTV Re:define charity gala. Michael who enjoyed a lucrative pop career as one half of duo Wham before embarking on a successful solo career died on Christmas Day last year at the age of 53.
rating: real
text:Solar-powered 'skin' could make prosthetics more real Many people try to stay out of the sun. But if a new type of solar-powered electronic skin makes its way onto prosthetics, wearers will definitely want those rays shining on their limbs. Researchers are already working to create smart skin that embeds sensors that mimic the tactile feedback of human skin, making it possible for amputees to feel pressure, temperature and even dampness. But how to power the futuristic material? A team from the University of Glasgow in the UK has come up with a version that harnesses the sun's rays. Because it produces its own energy from a natural source, the engineers say, the electronic skin would operate longer than similar materials powered by batteries or tethered to a power source that would also limit portability, clearly a key feature of any everyday prosthetic or touch-sensitive robot on the go.
rating: real
---
As a reminder we already have the following concepts which are useful:
1. Biased language:Biased language refers to the use of words, phrases, or expressions that have an underlying political or ideological agenda. This may include words or phrases that are used to promote a specific point of view, or language that is used to discredit or denigrate certain individuals or groups., possible responses: ['biased', 'non-biased', 'uncertain']
2. Non-credible Sources:Non-credible sources refer to sources of information that lack authority, accuracy, objectivity, or authenticity. This may include sources that are not verified, are not authoritative, or have a poor track record of accuracy., possible responses: ['non-credible source', 'credible source', 'uncertain']
3. Misleading Information:Misleading information refers to statements that are false or misleading, either intentionally or unintentionally. This may include factual errors, inaccurate comparisons, or claims that are not supported by evidence., possible responses: ['misleading', 'accurate', 'uncertain']
4. Exaggerated Claims in News Title:Exaggerated claims in news titles refer to claims made in the title of a news article that exaggerate the truth or lack evidence. This may include statements that are too good to be true, unrealistic promises, or claims that are not supported by evidence., possible responses: ['exaggerated', 'accurate', 'uncertain', 'not applicable']
5. Misleading Language:Misleading language refers to words, phrases, or expressions that are used to mislead or deceive. This may include statements that are false or inaccurate, are presented in a way to distort facts or reality, or are used to manipulate the reader's beliefs., possible responses: ['misleading', 'accurate', 'uncertain']
6. unverified sources:Unverified sources are sources of information which do not have reliable evidence or proof to back them up. This may include sources that are not properly fact checked, are not independently verified, or have a track record of inaccuracy., possible responses: ['unverified source', 'verified source', 'uncertain']

The following concepts have been rejected by the system, avoid making similar ones:
1. exaggerated claims:Exaggerated claims refer to statements that are intentionally exaggerated or hyperbolic in order to create a false sense of urgency or importance. This may include statements that are too good to be true, unrealistic promises, or claims that lack evidence or are not supported by facts., possible responses: ['exaggerated', 'accurate', 'uncertain', 'not applicable']
2. Misleading Headlines:Misleading headlines refer to headlines that are false or misleading, either intentionally or unintentionally. This may include factual errors, inaccurate comparisons, or claims that are not supported by evidence., possible responses: ['misleading', 'accurate', 'uncertain', 'not applicable']
3. Inaccurate Representation:Inaccurate representation refers to statements in the text that are false or misleading, either intentionally or unintentionally. This may include factual errors, inaccurate comparisons, or claims that are not supported by evidence., possible responses: ['accurate', 'inaccurate', 'uncertain']


Keeping in mind the pointers above, create a concept below that is distinct from the current set of concepts. Additionally, make sure that all possible responses can be mapped to either 1, 0, -1, or "na".
Make sure that the concept is as relevant to the fake_news as much as possible instead of being general. As the number of rejected concepts gets larger (3 or more), we can start being more specific, picking out particular details we notice in the examples above.
Note the strict adherence to the json format.
Definition: {
"Concept Name": "Misleading Information in the Title",
"Concept Description": "Misleading information in the title refers to statements in the title of a news article that are false or misleading, either intentionally or unintentionally. This may include factual errors, inaccurate comparisons, or claims that are not supported by evidence.",
"Concept Question": "What does the title of the article say about the accuracy of the information?",
"Possible Responses": ["misleading","accurate","uncertain","not applicable"],
"Response Guide": {
"misleading": "The title of the article contains misleading information, such as factual errors, inaccurate comparisons, or claims that are not supported by evidence.",
"accurate": "The title of the article contains accurate information.",
"uncertain": "It is difficult to determine the accuracy of the information in the title or if any misleading information is present.",
"not applicable": "The title does not contain any information."
},
"Response Mapping": {
"misleading": 1,
"accurate": -1,
"uncertain": 0,
"not applicable": "na"
}
}
```

E.2 Concept Improvement Prompt

```
Concept Improvement Task

We have a concept that needs to be improved. The goal of this task is to identify any issues with the current concept and suggest improvements to make it more valid, clear, well-phrased, and properly formatted in JSON. The concept should be designed for a positive/negative/uncertain question format that maps to 1, -1, and 0 respectively.
In this task, we will return information about any potential problems in the concept along with the improved concept.
Note it is also possible that the concept requires no further improvement (even minor ones), in which case, we will return the original concept with "None" for the other responses.

Consider the following error cases while improving the concept:

1. Lack of validity: Ensure that the responses are mutually exclusive and collectively exhaustive.
- Example of a concept that is not mutually exclusive:
Input JSON:
{{
"Concept Name": "review sentiment",
"Concept Description": "The sentiment expressed towards the product in the review. It could be positive, negative, or neutral.",
```

```
"Concept Question": "What is the overall feeling towards the product?",
"Possible Responses": ["positive", "somewhat positive", "negative"],
"Response Guide": {{
"positive": "The reviewer expresses a positive opinion on the product, such as praising its quality, performance, or value.",
"somewhat positive": "The reviewer expresses a somewhat positive opinion on the product, such as mentioning some good aspects but also pointing out some flaws.",
"negative": "The reviewer expresses a negative opinion on the product, such as criticizing its quality, performance, or value."
}},
"Response Mapping": {{
"positive": 1,
"somewhat positive": 0.5,
"negative": -1
}}
}}
Response: {{
"Confirmation":"1. Mutual Exclusivity: The concept above contains non-mutually exclusive responses 'positive' and 'somewhat positive'
2. Collectively Exhaustive: clear
3. No leading questions: clear
4. Rich and objective response guide: clear
5. Interference with other concepts: clear
6. Invalid response set: clear",
"Errors": "The concept above contains non-mutually exclusive responses 'positive' and 'somewhat positive'",
"Fix": "We can address this by either combining 'positive' and 'somewhat positive' into a single response or defining clearer distinctions between them.",
"New Concept": {{
"Concept Name": "review sentiment",
"Concept Description": "The sentiment expressed towards the product in the review. It could be positive, negative, or neutral.",
"Concept Question": "What is the overall sentiment expressed towards the product in the review?",
"Possible Responses": ["positive", "negative", "neutral"],
"Response Guide": {{
"positive": "The reviewer expresses a positive opinion on the product, such as praising its quality, performance, or value.",
"negative": "The reviewer expresses a negative opinion on the product, such as criticizing its quality, performance, or value.",
"neutral": "The reviewer does not express a clear positive or negative opinion on the product, or the review contains a mix of positive and negative aspects."
}},
"Response Mapping": {{
"positive": 1,
"negative": -1,
"neutral": 0
}}
}}
}}###

- Example of a concept that is not collectively exhaustive:
Input JSON:
{{
"Concept Name": "product availability",
"Concept Description": "The availability of the product as described in the review.",
"Concept Question": "Is the product available?",
"Possible Responses": ["available", "unavailable"],
"Response Guide": {{
"available": "The reviewer mentions that the product is available, in stock, or easy to find.",
"unavailable": "The reviewer mentions that the product is unavailable, out of stock, or hard to find."
}},
"Response Mapping": {{
"available": 1,
"unavailable": -1
}}
}}
Response: {{
"Confirmation": "Confirmation:
1. Mutual Exclusivity: clear
2. Collectively Exhaustive: The concept above contains a non-collectively exhaustive response set
3. No leading questions: clear
4. Rich and objective response guide: clear
5. Interference with other concepts: clear
6. Invalid response set: clear"
"Errors": "The concept above contains a non-collectively exhaustive response set because it may be possible that a piece of text does not strictly match some criteria in the response guide,",
"Fix": "We can address this by adding a 'uncertain' response to cover cases where the availability is not clearly mentioned, and a 'not applicable' response for cases where the text does not discuss a product.",
"New Concept": {{
"Concept Name": "product availability",
"Concept Description": "The availability of the product as described in the review.",
"Concept Question": "What does the review say about the product's availability?",
"Possible Responses": ["available", "unavailable", "uncertain", "not applicable"],
"Response Guide": {{
"available": "The reviewer mentions that the product is available, in stock, or easy to find.",
"unavailable": "The reviewer mentions that the product is unavailable, out of stock, or hard to find.",
"uncertain": "The review contains mixed information or no information that makes it difficult to determine the availability of the product.",
"not applicable": "The reviewer is not discussing a product or anything else that could be described by this concept."
}},
"Response Mapping": {{
"available": 1,
"unavailable": -1,
"uncertain": 0,
"not applicable": "na"
}}
}}
}}###

2. Poor phrasing: Avoid leading questions and provide rich examples in the response guide.
- Example of a leading question:
Input JSON:
{{
"Concept Name": "product cost",
"Concept Description": "The cost of the product as described in the review.",
"Concept Question": "Is the product expensive?",
"Possible Responses": ["expensive", "affordable", "uncertain", "not applicable"],
"Response Guide": {{
"expensive": "The reviewer thinks the product is expensive.",
"affordable": "The reviewer thinks the product is affordable.",
"uncertain": "The reviewer does not provide a clear opinion on the product's cost.",
"not applicable": "The reviewer is not discussing a product or anything else that could be described by this concept."
}},
"Response Mapping": {{
"expensive": 1,
"affordable": -1,
"uncertain": 0,
"not applicable": "na"
}}
}}
Response: {{
"Confirmation":"1. Mutual Exclusivity: clear
2. Collectively Exhaustive: clear
3. No leading questions: The question may potentially be leading, "Is the product expensive?" leads the responder to a positive response.
4. Rich and objective response guide: The response guide is not very rich or objective. The responses guides for each answer are very similar, only differing by a few words and without providing examples.
5. Interference with other concepts: clear
6. Invalid response set: clear",
"Errors": "The concept contains a leading question and a bad response guide",
"Fix": "We can address this by changing the question to be more neutral and asking about the reviewer's description of the product's cost.",
"New Concept": {{
"Concept Name": "product cost",
"Concept Description": "The cost of the product as described in the review, in terms of whether the product is perceived as expensive or affordable.",
"Concept Question": "How does the reviewer describe the cost of the product?",
"Possible Responses": ["expensive", "affordable", "uncertain", "not applicable"],
"Response Guide": {{
"expensive": "The reviewer describes the product as costly, high-priced, or not worth the money.",
"affordable": "The reviewer describes the product as reasonably priced, good value for money, or budget-friendly.",
"uncertain": "The reviewer does not provide a clear opinion or information on the product's cost.",
"not applicable": "The reviewer is not discussing a product or anything else that could be described by this concept."
}},
"Response Mapping": {{
"expensive": 1,
"affordable": -1,
"uncertain": 0,
```

```
"not applicable": "na"
}}
}}
}}###

In addition to the errors above, some other problems could
      be:

1. The concept contains responses not in {{1, 0, -1, "na"}}.
       This can be fixed by shrinking the possible set of
      responses.
2. Lack of detail in the response guide. As much as possible
      , the response guide should contain detailed examples.
      This issue can be fixed by making the response guide
      more specific to allow annotators to be more objective
      about answering the question.

---
Below is the concept for you to improve.
{}
Response:{{
```

E.3 Concept Measurement Prompt

```
Answer the following question about the texts below by
      selecting from the following choices. Before answering
      the question, extract any potentially relevant snippets
       of the text that can serve as evidence for each
      classification. After that, compare the snippets
      against the response guide to come up with a final
      decision.

Format your response as a list of JSON objects with string
      keys and string values. Below is an example of a valid
      JSON response. Each JSON object contains keys for
      snippets, thoughts, and answer. End your response with
      ###
---
Text 1: Text
Text 2: Text
Text 3: Text

Response JSON:[
{"text": "Text 1", "snippets": {
"classification 1" : ["Snippet 1", "Snippet 2", ...],
"classification 2" : ["Snippet 3", "Snippet 4", ...]
...
},
"thoughts": "In this section, you weigh evidence based on
      the text and the extracted snippets to come to a final
      decision with the response guide as a reference. Be as
      objective as possible and ignore irrelevant information
      . Focus only on the snippets and avoid making guesses
      .",
"answer": "An answer from the response guide goes here. In
      answering the question, ignore irrelevant information
      and avoid making assumptions."},
{"text": "Text 2", "snippets": {
"classification 1" : ["Snippet 1", "Snippet 2", ...],
"classification 2" : ["Snippet 3", "Snippet 4", ...]
...
},
"thoughts": "...",
"answer": "..."},
{"text": "Text 3", "snippets": {
"classification 1" : ["Snippet 1", "Snippet 2", ...],
"classification 2" : ["Snippet 3", "Snippet 4", ...]
...
}
]###
---
Below is an example of the task being performed with the
      concept "build quality":

Concept:
{
"Concept Name": "good build quality",
"Concept Description": "build quality refers to the
      craftsmanship, durability, and overall construction of
      a product. It encompasses aspects such as materials
      used, design, manufacturing techniques, and attention
      to detail. A product with good build quality is
      typically considered to be well-made, sturdy, and long-
      lasting, while a product with poor build quality may be
       prone to defects or wear out quickly.",
"Concept Question": "What does the review say about the
      build quality of the product?",
"Possible Responses": ["high", "low", "uncertain", "not
      applicable"],
"Response Guide": {
"high": "Review mentions aspects such as well-made, sturdy,
      durable, high-quality materials, excellent
      craftsmanship, etc.",
"low": "Review mentions aspects such as poor construction,
      flimsy, cheap materials, bad design, easily breakable,
      etc.",
"uncertain": "Review does not mention build quality, the
      information is ambiguous or vague, or it has both
      positive and negative aspects mentioned like 'the
      product is sturdy but uses cheap materials'.",
"not applicable": "The review does not mention the build
      quality of the product at all."
}
}

Text 1: "This product has a great design and is really easy
      to use. It is also very durable."
Text 2: "Was excited for it to finally arrive, got here in
      nice sturdy packaging. Opened it up though and it
      smelled kind of weird? goes away after a while but
      otherwise an ok product. Saw some print aberrations it
      didn't interfere much with use."
Text 3: "A big fan of the product. Serves me well during
      workouts but I go through them like hotcakes. Don't
      expect it to last long compared to other brands but you
       get what you pay for. It does the job though."
Text 4: "Very disappointing. I was excited to order this but
       when it arrived I was shocked at how poorly it worked.
       Deceptive advertising at its finest."

Response JSON:[
{"text": "Text 1", "snippets": {
"high": ["It is also very durable", "really easy to use"],
"low": [],
"uncertain": []
},
"thoughts": "Two snippets for high. The first is related to
      durability which is an aspect of good build quality.
      The second is related to ease of use which is not
      related to good build quality. Overall the text
      describes good build quality..",
"answer": "high"},

{"text": "Text 2", "snippets": {
"high": ["got here in nice sturdy packaging"],
"low": ["Saw some print aberrations"],
"uncertain": []
},
"thoughts": "One snippet for high, one snippet for low. The
      low snippet mentions defects in manufacturing. In the
      high snippet, 'Sturdy' only refers to the packaging,
      not the product. The balance of evidence leans towards
      a classification of low.",
"answer": "low"},

{"text": "Text 3", "snippets":
{
"high": ["Serves me well during workouts"],
"low": ["Don't expect it to last long compared to other
      brands", "I go through them like hotcakes"],
"uncertain": ["you get what you pay for"]
},
"thoughts": "One high snippet, two low snippets, one
      uncertain snippet. The high snippet is about utility
      which is not related to build quality. The low snippet
      relates to durability, an aspect of build quality.",
"answer": "low"},

{"text": "Text 4", "snippets":
{
"high": [],
"low": ["Very disappointing", "shocked at how poorly it
      worked", "Deceptive advertising"],
"uncertain": []
},
"thoughts": "Three low spans. The first is related to
      overall judgment which is irrelevant, the second is
      related to functionality which is irrelevant, and the
      third is related to marketing/advertising which is also
       irrelevant. None are related to build quality.",
"answer": "uncertain"
}
]###
---
Perform the task below, keeping in mind to limit snippets to
       10 words and ignoring irrelevant information. Return a
       valid list of JSON objects ending with ###
Concept: {'Concept Name': 'Price to Quality Ratio', 'Concept
       Description': "Price to Quality Ratio is a measure of
      a customer's perception of the value of the goods or
      services they receive relative to the price they paid.
      A high price to quality ratio indicates that the
      customer believes they have received good value for
      their money, while a low price to quality ratio
      suggests that the customer believes they did not
      receive enough in return for the money they paid.", '
      Concept Question': "What is the customer's perception
      of the price to quality ratio of the goods or services
      they received?", 'Response Guide': {'high': 'The
      customer believes they have received a good value for
      their money, such as feeling that the product is worth
      more than what they paid for it.', 'average': 'The
      customer believes they received an average value for
      their money, such as feeling that the product is worth
      the same as what they paid for it.', 'low': 'The
```

```
      customer believes they did not receive enough in return
       for the money they paid, such as feeling that the
      product is not worth what they paid for it.', '
      uncertain': "It is difficult to determine the customer'
      s perception of the price to quality ratio, such as
      when the customer does not provide a clear opinion or
      information on the product's cost."}}

Text 1: I'm very very torn about how many stars to give Plum
      . Very torn.\n\nHere are the positive things:\n+great
      sushi\n+great interior\n+very solid appetizers and a
      knock-out hot and sour soup\n+good drink menu\n\nBut
      the negatives, oh...\n-service is almost non-existent.
      I've had to get up after 15 minutes of being ignored
      after being seated to ask if we have a waiter. No one
      even brought us water (!!) Then, the woman I asked,
      instead of apologizing, made me feel guilty by telling
      me that she wasn't my waiter, but she supposed she'd
      get us water and take our order (I suspect she was our
      waiter). This is not the only service horror story I
      have\n-price. Look, I get that this place is nice, but
      you can't charge this much for food and have service
      this bad. It's just insulting to have a waitress in
      sneakers when you're paying for a $30 plate of sushi. \
      nOh, and more staff issues. They have some guy busing
      tables in a baseball hat who has certainly been stoned
      every time I've been there. It's amazing. \n\nFood:
      pretty great, if a little pricey. \nService, staff, etc
      : Horrifically bad. \n\nFire the staff, re-hire people
      who know how to dress appropriately for the atmosphere,
       politely attend to customers, etc. Also, if you're
      going to have non-wait staff bus tables, you need to
      have them dress as waitstaff and politely take items.
      The guy they have now is like a (stoned) bull in a
      china shop. No one wants to pay that amount of money
      for food and be ignored.
Text 2: The light rail has become a big part of my life.  I
      moved into a place specifically for it's proximity to
      the Thomas/Central station.  I go to ASU and have
      ridden it every day this week, here's why:\n\n1. It's
      free, because I'm an ASU student\n2. It makes the walk
      at 6:30am in the dark seem kinda cool\n3. I enjoy
      sitting and not driving\n4. I finished a book with just
       my metro reading time\n5. I get some much needed \"me
      \" time\n6. I never have to deal with ASU's parking
      EVER AGAIN\n\nThanks light rail for keeping me sane and
       helping save me money until my goddamn FAFSA check
      comes in the mail.  I'm taking off a star because you
      don't run late enough to take back from work.\n\nI know
       you're thinking \"jesus, it sounds like this girl's
      life sucks hard core\".  Yeah, it really does.

Response JSON:[
{{
"text": "Text 1", "snippets": {
"high": ["great sushi", "great interior", "very solid
      appetizers", "knock-out hot and sour soup", "good drink
       menu"],
"low": ["price", "service is almost non-existent", "no one
      even brought us water", "made me feel guilty", "can't
      charge this much for food and have service this bad", "
      non-wait staff bus tables"],
"average": ["Food: pretty great, if a little pricey"],
"uncertain": []
},
"thoughts": "The text contains five snippets for high, six
      snippets for low, and one snippet for average. The
      snippets for high are related to the quality of the
      goods or services, while the snippets for low are
      related to the price and service. The balance of
      evidence leans towards a classification of low.",
"answer": "low"
},

{"text": "Text 2", "snippets": {
"high": ["It's free", "makes the walk at 6:30am in the dark
      seem kinda cool", "I enjoy sitting and not driving", "I
       finished a book with just my metro reading time", "I
      get some much needed \"me\" time", "I never have to
      deal with ASU's parking EVER AGAIN"],
"low": ["doesn't run late enough to take back from work"],
"average": [],
"uncertain": []
},
"thoughts": "The text contains six snippets for high and one
       snippet for low. The snippets for high are related to
      the quality of the goods or services, while the snippet
       for low is related to the price. The balance of
      evidence leans towards a classification of high.",
"answer": "high"
}
]
```

F Supplementary Material

Figure 11 shows an example of a survey question used to measure the concept “Quality of Service”.

Quality of Service

This concept refers to the assessment of the service delivered by the restaurant's staff, it could be high, low, not mentioned, or uncertain. It includes aspects such as politeness, punctuality, attention to customer needs, and overall demeanor.

How does the review describe the quality of service at the restaurant?

[high quality] - The reviewer compliments the service, with phrases indicating that the staff were attentive (met all client needs promptly), polite (had a respectful, friendly demeanor), and efficient (served customers in a timely manner).

[low quality] - The reviewer criticizes the service, with phrases indicating that the staff were inattentive (neglected customer needs), rude (had a dismissive or disrespectful demeanor), slow (took too long to serve customers), or inefficient (were disorganized or made mistakes).

[not mentioned] - The review does not provide any clear information on the quality of service at the restaurant.

[uncertain] - The review is vague or ambiguous about the quality of service, or the reviewer has mixed feelings about it (mentions both positive and negative aspects).

[None of the above] - If none of the answers apply or are valid

Figure 11: Survey format