8 Cost for Running the Experiments

For each problem, we ran the proposer for 10 times on average; assuming each prompt to be at most 4000 tokens, we spent around $2.4 for each problem on `OpenAI` APIs if we use `gpt-4` and `text-davinci-003`, and the cost would decrease to $0.8 if we use `gpt-3.5-turbo`. Notice that these estimates are computed based on the prices as of 05/14/2023, and we expect the price to further decrease in the future. We ran the Flan-T5 based validator for $\tilde{2}$ hours on 1 80G A100 GPUs.

The total amount of computational resources spent in this research paper is around $2,500 in terms of `OpenAI` API and 3,000 hours of compute on A100 GPU with 80G memory.

9 Generation Process of SYND5

The high-level description is in Section 2.2. Here we discuss the procedure that generated SYND5.

We consider three dimensions of differences: topic, genre, and style. For each, we generated 14/9/7 values, e.g., "*celebrity love stories*" and "*sports team recruiting athletes* for the topic attribute, "*rap lyrics*" and "*screen play*" for the style attribute, and "*French*" and "*Spanish*" for the language attribute. We then used GPT-4 and the Claude API to synthesize 54K text samples, where for each text sample we sampled a topic, genre, and style randomly, e.g. "*Write a rap about a sports team recruiting athletes in French*". To synthesize a random SYND5 problem, we randomly sampled a distractor dimension (e.g. language) and a target dimension (e.g. topic), and for each dimension we sampled two random values (e.g. English and French for language, sports and art for topic).

For each problem, we sampled 10 texts for corpus A such that all of them satisfy one sampled value for the distractor dimension (e.g. corpus A is entirely in English), and 10 texts for corpus B for to satsify the other distractor dimension (e.g. corpus B is entirely in French). Then we set V fraction of corpus A to satisfy the reference target attribute, e.g. "is sports-related", and f fraction of corpus B to satisfy the other value for the target dimension (e.g. "is art-related"). We chose V uniformly at random from [0.6, 0.8, 1]. Finally, we provide k example hypotheses from the target dimension other than the target dimension values for Corpus A and Corpus B, and we chose k from [0, 2] uniformly at random. We then sampled 300 D5 problems in total from this distribution.

10 Robustness Checks for Results on SYND5

Table 4 shows the accuracy of different systems using `text-davinci-003` as the judge for semantic equivalence. Table 5 shows the accuracy of different systems if we consider outputs semantically similar to the reference to be correct. Across all setups, we found that the conclusion reached in Section 5 still holds under these robustness checks.

`text-davinci-003`	w/ goal	wo/ goal	`gpt-4`	w/ goal	wo/ goal
w/ validator	6%	1%	w/ validator	23%	9%
wo/ validator	3%	0%	wo/ validator	6%	2%

Table 4: Same Table as 1, except that we use `text-davinc-003` instead `Claude-v1.3` to judge similarity. Using the validator, the goals, and `gpt-4` leads to better results.

`text-davinci-003`	w/ goal	wo/ goal	`gpt-4`	w/ goal	wo/ goal
w/ validator	46%	23%	w/ validator	53%	43%
wo/ validator	24%	16%	wo/ validator	24%	24%

Table 5: Same Table as 1, except that we calculate how often the output is similar, rather than equivalent, to the reference. Using the validator, the goals, and `gpt-4` leads to better results.

11 Computing Turker Judgement

Scoring. To estimate $T(h, x)$ with Turker's rating, where h is a truth predicate of a text sample x, the Turker needs to read h and x and then choose among six options: "Certainly Yes", "Likely Yes",

"Neutral", "Likely No", "Certainly No", and "Confusing/Cannot be decided." For each (h, x) pair, we collect responses from three Turkers. To compute the average across them, we collect a list of scores using the following rule: each "Certainly Yes" would receive a score of 1.00, "Likely Yes" 0.75, "Neutral" 0.50, "Likely No" 0.25, "Certainly No" 0.00, and "Confusing/Cannot be decided." receive two scores of 0.50. We then take the average over all the scores we collected from the Turkers for one h and x. "Confusing/Cannot be decided." receives two scores of 0.50 because we want such a response to drag the average rating towards neutral and it has a larger effect than choosing "Neutral".

Payment. We adjust the payment for each HIT task based on the number of words they need to read. We pay them approximately 0.001 cent per word, and using the conservative estimate that adults read about 200 words per minute, we pay them around $12 per hour. We spent in total around $5K on this HIT task.

Qualification. We only recruited Turkers who are located in the U.S. Additionally, we designed qualification test with 8 questions; the questions are designed to be easy to answer as long as they have read our instructions below, and we only accepted turkers who made mistakes on at most one questions.

Annotation Instruction. We show our annotation instruction below. We only show examples of choosing "Certainly Yes", "Certainly No", and "Confusing" to encourage the Turkers not to choose neutral ratings. Additionally, we explicitly tried to address Halo effect – where the text does not satisfy a predicate h but satisfies a predicate h' that is highly correlated with h. For example, for the text sample $x =$ "*Really love the flight!!*" does not satisfy the predicate $h =$ "*mentions that the breakfeast is good on the plane*", even though it satisfies a highly correlated predicate $h' =$ "*likes the flight.*"

11.1 Instructions

Below are the same instructions we have shown you during the qualification. Thanks for visiting this page and refresh your memory about the instruction!

Instruction: In this task, you will check whether a TEXT satisfies a PROPERTY

Example 1
Property: mentions a natural scene.
Text: I love the way the sun sets in the evening.

- A) Certainly Yes.
- B) Likely Yes.
- C) Neutral.
- D) Likely No.
- E) Certainly No.
- F) Confusing/Cannot be decided.

Answer. A. sun set is nature-related; if you feel a bit ambivalent, B is also acceptable.

Example 2
Property: writes in a 1st person perspective.
Text: Makima is cute.

- A) Certainly Yes.
- B) Likely Yes.
- C) Neutral.
- D) Likely No.
- E) Certainly No.
- F) Confusing/Cannot be decided.

Answer. E. This text is undoubtedly written in the 3rd person perspetive, so E.

Example 3
Property: is better than group B.
Text: I also need to buy a chair.

- A) Certainly Yes.
- B) Likely Yes.
- C) Neutral.
- D) Likely No.
- E) Certainly No.
- F) Confusing/Cannot be decided.

Answer. F. It is unclear what the hypothesis mean (e.g., what does group B mean?) and doesn't seem related to the text. So F.

Example 4
Property: mentions that the breakfast is good on the airline.
Text: The airline staff was really nice! Enjoyable flight.

- A) Certainly Yes.
- B) Likely Yes.
- C) Neutral.
- D) Likely No.
- E) Certainly No.
- F) Confusing/Cannot be decided.

Answer. E. Although the text appreciates the flight experience, it DOES NOT mention about the breakfast. So the answer is E.

Example 5
Property: appreciates the writing style of the author.
Text: The paper absolutely sucks because its underlying logic is wrong. However, the presentation of the paper is clear and the use of language is really impressive.

- A) Certainly Yes.
- B) Likely Yes.
- C) Neutral.
- D) Likely No.
- E) Certainly No.
- F) Confusing/Cannot be decided.

Answer. A. Although the text dislikes the paper, it DOES like the writing style. So the answer is A.

12 Prompt to Judge Predicate Similarity

We prompt Claude v1.3 (Bai et al., 2022b) to judge whether the predicated predicate is similar to the reference. We consider a response that leads to a"yes" to be correct when we require the discovery to be semantically equivalent to the reference, and consider a response that leads to a "yes" or "related" to be correct when we require the discovery to be semantically similar to the reference.

> " *Is text_a and text_b similar in meaning? respond with yes, related, or no.*
>
> *Here are a few examples.*
> *Example 1:*
> *text_a: has a topic of protecting the environment*
> *text_b: has a topic of environmental protection*

and sustainability
output: yes

Example 2:
text_a: has a language of German
text_b: has a language of Deutsch
output: yes

Example 3:
text_a: has a topic of the relation between political figures
text_b: has a topic of international diplomacy
output: related

Example 4:
text_a: has a topic of the sports
text_b: has a topic of sports team recruiting new members
output: related

Example 5:
text_a: has a named language of Korean
text_b: uses archaic and poetic diction
output: no

Example 6:
text_a: has a named language of Korean
text_b: has a named language of Japanese
output: no

Target:
*text_a: {**predicate**}*
*text_b: {**reference**}*
output:"

13 Meaningfulness: Relevance, Novelty, and Significance

Not every valid discovery is meaningful. For example, if the goal is to understand the topical differences between news from 2008 (Corpus A) and news from 2007 (Corpus B), the discovery that Corpus A "*contains news from 2008*" is completely valid by definition but meaningless, since it provides only trivial information and is irrelevant to the goal of understanding topical differences.

McGarry (2005) surveyed a list of desirable properties for discovery, and we condensed them into three submetrics to rate how meaningful a discovery is based on the exploration goal: 1) relevance, 2) novelty, and 3) significance. We evaluate these independently of validity and assume that the discovery is already valid. For example, the discovery that "something can travel faster than light" is meaningful if true, even though it is highly implausible.

We rate each submetric with ⓪, ①, or ②, where higher is better. We show the evaluation instructions below and present our rating on `text-davinci-003` proposed hypotheses.

13.1 Evaluation Instructions

Relevance. How relevant the discovery is to the goal. For example, suppose we were a student comparing essays rated as convincing vs. not convincing to figure out what writing style is convincing. Then:

- The discovery "*write in first person*" is directly related to the writing style, so we rate it ②.
- The discovery "*use the word "I"*", is not exactly a writing style, but can still inform the relevant underlying principle of "*write in first person*", so we rate it ①.

- The discovery "*argue for abortion*" does not tell us about the underlying writing style, so we rate it ⓪.

Novelty. The difficulty of generating the discovery, e.g. can we think of the discovery in 5 minutes with the goal but without looking at the corpora? For example, suppose we were an airline manager trying to find improvements to the flight experience, and we were comparing negative reviews vs. positive reviews. Then:

- The discovery "*contain more negative language*" is almost certain for negative reviews, so we rate it ⓪.
- The discovery "*complain about the crew members*" is not entirely novel, but is not tautologically true and hence requires confirmation, so we rate it ①.
- The discovery "*mention a language barrier with the crew members*" is specific and hard to think of without looking at the data, so we rate it ②.

Note that our evaluation is "blinded to the samples": we still consider a discovery novel as long as it is hard to think of before looking at the corpora, even if it might be easy to think of after looking at the corpora. For example, the physical law that $F = ma$ is easy to observe if we have collected and plotted the data on acceleration, mass, and force; however, it might be difficult to think of before we see any such data, so we consider it novel.

Significance. Given the exploration goal, how beneficial is it to learn the discovery for the first time? For example, suppose we were an Amazon retailer trying to figure out what customers like and dislike about my product based on negative reviews and positive reviews. Then:

- The discovery "*accuses the team pushing out a bad product*" is not significant since it cannot direct the retailer to improve the product, so we rate it ⓪.
- The discovery "*asks for a more durable product*" gives some hints about how to improve the product, but isn't sufficiently helpful on its own, so we rate it ①.
- The discovery "*says the wrench is missing*" can lead to concrete actions for improvement, so we rate it ②.

13.2 Goal Leads to More Meaningful Hypotheses

	with-goal	no-goal	kappa	spearmanr	p of avg	worst p of ind
Relevance	1.68	1.20	0.56	0.71	1×10^{-10}	1×10^{-8}
Novelty	1.24	0.97	0.37	0.50	5×10^{-6}	4×10^{-2}
Significance	1.56	1.05	0.46	0.64	2×10^{-10}	2×10^{-7}

Table 6: **Left.** For each metric, we report the average rating on hypotheses generated with or without using the exploration goal, and find that the former performs better. **Middle.** The inter-annotator agreement rate averaged across pairs of author evaluators, measured by Kappa and Spearman rank coefficient; we find substantial correlations between evaluators across all these subjective metrics, with relevance $>$ significance $>$ novelty. **Right.** We compute the p-values for the null hypothesis that "with-goal and no-goal result in the same performance". The p of avg column reports the p-values after we average the ratings from all evaluators, while the "worst p of ind" column takes the max of all p-values based on ratings of individual evaluators. Overall, the conclusions are statistically significant and they can be robustly reproduced across individual evaluators.

Compared to Zhong et al. (2022), we added the exploration goal to our prompt when generating hypotheses. Does this improve the quality of the proposed hypotheses? To investigate this, we sampled 100 problems from OpenD5 with distinct exploration goals and randomly sampled 2 hypotheses from GPT-3 with and without using exploration goal (see Figure 3), resulting in 400 hypotheses to evaluate. Three authors then rated their meaningfulness based on the three metrics defined in Section 3, while being blinded about which hypotheses were generated with the exploration goal.

The results are shown in Table 6. We found that, when prompted with the exploration goal, GPT-3 on average proposes more relevant, novel, and significant hypotheses; additionally, it proposes hypotheses with ratings higher than ⓪ 31%/21%/28% more often in terms of relevance/novelty/significance.

Since this is a subjective evaluation, the Kappa inter-annotator agreement is only moderate, ranging from 0.37 to 0.56. However, we can still robustly conclude that the model can propose more meaningful hypotheses when conditioned on the goal: we calculate the p-values for the null hypothesis that with-goal and no-goal have equal performance, and we find p-values to be highly significant and robust across evaluators, for all three submetrics.

14 Full Pipeline of the Proposer

We present the full details of how we generated the hypotheses with the language model. The process roughly contains four stages: 1) obtaining representative samples for each corpus, 2) sampling hypotheses from GPT-3, 3) rewriting hypotheses, and 4) optionally plugging in example hypotheses.

Obtaining representative samples. This step is the same as Zhong et al. (2022), and we borrow the related text from that paper for the reader's convenience. Since $\mathcal{D}_A^{\text{res}}$ and $\mathcal{D}_B^{\text{res}}$ might overlap significantly, random samples from $\mathcal{D}_A^{\text{res}}$ and $\mathcal{D}_B^{\text{res}}$ might not be representative and informative enough for GPT-3 to notice the differences between the two distributions. Therefore, we choose samples that are representative of their differences. To find those samples, we fine-tune RoBERTa-Large Liu et al. (2019) to predict whether each sample comes from Corpus A or Corpus B and keep the top-p percentile samples with the highest confidence. Next, we take samples from the top-p percentile to prompt GPT-3.

Selecting samples to prompt GPT-3. We randomly select S =25 samples from the top-5 percentile from Corpus A and Corpus B to prompt GPT-3 to propose the hypotheses, using the template shown in Figure 3 left. We require the length of the prompt to be at most 3,200 GPT-3 tokens (the max window size for GPT-3 text-davinci-003 is 4096) and gradually decrease the number of samples S in the prompt until the prompt length is less than 3,200; additionally, we truncate each text samples to at most 256 GPT-3 tokens. Finally, to prevent GPT-3 from proposing hypotheses that reflect simple lexical correlations that can be detected with unigram models, e.g., "*uses the word "hey" more often.*", we incrementally construct the subset of samples for Corpus A and Corpus B such that at any time of the construction, no single word can appear $0.25S$ times more often in one corpus than the other. We repeat the same process for the top-20 and top-100 percentile until we obtain 60 hypotheses.

Rewriting hypotheses with GPT-3. As mentioned in Section 6.2, the hypotheses generated by GPT-3 are frequently statements about the corpus, while the validator requires the hypothesis to be a predicate on individual text samples. For example, when comparing definitions that people like from `UrbanDictionary.com` to other definitions, the hypothesis that the former "*is more likely to include slang or colloquial terms.*" is a statement about a collection of text samples, rather than a predicate on an individual sample. $T(h, x)$ is undefined in this case, since it does not make sense to check whether a single text sample is more likely to include slang. Ideally, we want to detect these comparison statements and automatically remove the comparatives, e.g., rewrite it to "*includes slang or colloquial terms.*".

To detect and remove the comparatives from the hypotheses, we tag the part of speech for each word in the hypotheses using the NLTK package (Bird et al., 2009) and check whether any tag is `JJR` or `RBR`. If a hypothesis indeed contain theses tags, we prompt GPT-3 to rewrite the hypothesis. We show an example prompt in Figure 4.

Plugging in example hypotheses (optionally). We can also add a few problem-specific example hypotheses to the prompt to elicit more relevant hypotheses, and we do so by adding them to the "formatting instruction" part in the prompt used to propose hypotheses Figure 3. In OpenD5, we provided example hypotheses for each problem to steer our system to generate more meaningful discoveries; we produced the example hypotheses by prompting GPT-3 to generate a few hypotheses and selecting the meaningful ones from them.

For the reported discoveries in Section 6.1, we confirmed that they are unambiguously different from our provided hypotheses; otherwise, the system might have produced the discoveries by copying the provided hypotheses. We did not use the example hypotheses in Section 5 to test GPT-3's zero-shot understanding of the goal.

```
Remove the comparatives. Remove mention of Group A or B if they appear.

Input: contain longer sentences than those from group B
Output: contain long sentences

Input: supports DACA more strongly
Output: supports DACA strongly

Input: uses the hashtag #AllLivesMatter more often
Output: uses the hashtag #AllLivesMatter

Input: is more likely to contain grammatical errors
Output: contain grammatical errors

Input: sounds happier than those from Group B
Output: sounds happy

Input:  is more likely to include slang or colloquial terms
Output: includes slang or colloquial terms
```

Figure 4: The prompt to remove comparatives from a hypotheses.

15 Collecting Data to Fine-tune the Validator

Here we provide a high-level description of how the data was collected. For each problem in OpenD5, we used our proposer to produce a list of hypotheses. We automatically judged each hypothesis on a subset of samples from the research split using GPT-3 text-davinci-002 (Ouyang et al., 2022), Flan-T5 (Chung et al., 2022), and a model trained with RLHF from Bai et al. (2022a). We created the input distribution for training by combining and equally weighting the following $3 \times 2 = 4$ distributions: the subset of (h, x) pairs that GPT-3/Flan-T5/"RLHF" considers Yes or No to be the most likely answer. We then collected averaged turker ratings for in total 3138 (h, x) pairs and used them to fine-tune Flan-T5 to create the validator (Chung et al., 2022).

To test cross problem generalization capability of our D5 system, whenever we applied our D5 system to a problem in OpenD5 in Section 6.1, we used a validator that is NOT fine-tuned on the (h, x) pairs from this problem. We achieved this by keeping track of which problem each (h, x) pair comes from and split all the (h, x) pairs into three folds based on the problems; whenever we applied our D5 system to a problem, we used the validator trained on the two folds that do not contain this problem.

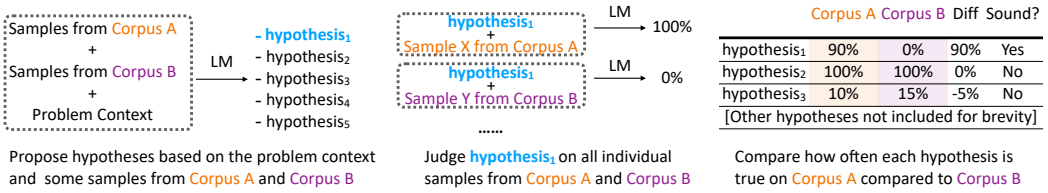

Figure 5: A sketch of the baseline method. The description can be seen in Section 4 and the actual prompts can be seen in Figure 3.

16 What Discoveries Did we Choose to Present

Our system in total produces 3296 discoveries on OpenD5. However, we do not have enough budget to validate every finding, since estimating V is expensive (Section **??**). Therefore, from the remaining 3296 discoveries, we manually selected 21 discoveries that 1) the authors think are relevant enough, 2) are representative of potential use cases, 3) do not require expert knowledge for Turkers to judge, and 4) are likely to achieve a small p-value with fewer than 200 samples from $\mathcal{D}_A^{\text{val}}$ and $\mathcal{D}_B^{\text{val}}$. We then estimated their validity based on the procedure described in Section **??** by using fewer than 200 samples from the validation split and calculated the p-values.[2] Since we are testing

[2]We determined the number of samples s.t. V' can achieve a p-value of 0.005. Estimating V for these discoveries costs ∼$1500.

discovery	V	p	V'	p'
argues for a path forward to promote the fair ...	0.16	1.26e-04	0.35	2.01e-73
refers to illegal immigrants as criminals	0.09	6.17e-03	0.19	3.17e-38
has an informal tone, such as slang or colloqu...	0.08	2.35e-03	0.24	1.46e-35
mentions lack of legroom	0.16	1.15e-03	0.38	1.34e-45
mentions children or family	0.08	1.00e-05	0.11	8.05e-09
Uses language that is positive or uplifting	0.12	2.12e-03	0.24	4.18e-59
references violence or aggression	0.06	9.87e-03	0.17	4.25e-26
involves physical activity, such as walking, p...	0.13	4.92e-03	0.37	7.07e-101
contains keywords related to business, finance...	0.08	2.89e-02	0.35	1.45e-95
mention disasters and crimes, such as plane ac...	0.03	7.03e-02	0.09	4.61e-06
discusses coronavirus-related topics	0.21	1.01e-04	0.27	9.19e-78
references pop culture, such as movies, books,...	0.21	2.67e-04	0.58	2.09e-30
uses vivid imagery and metaphors to convey a f...	0.09	2.47e-02	0.45	5.04e-64

Table 7: The full table of discoveries, along with their V, V', p, and p' scores.

multiple discoveries and each of them can be statistically significant merely due to chance, we keep 13 discoveries with V that are significantly non-zero with p-value below 7%, a threshold determined by the Benjamini Hochberg's procedure with a false discovery rate of 10%. In other words, fewer than 10% of the discoveries presented are false discoveries in expectation.

17 More Example Discoveries on OpenD5

Analyzing errors in NLP systems. We considered the task of perspectrum classification (Chen et al., 2019), which has the following instruction: "given a perspective and a claim, classify whether the given perspective supports or undermines the claim. If the perspective could possibly convince someone with different view, it is supporting, otherwise it is undermining." We considered two few-shot learning systems: GPT-3 Instruct Curie (Ouyang et al., 2022) and Tk-Instruct-11B (Wang et al., 2022). We focused on the perspectives where the ground truth label is undermining, and compare the following two corpora: Corpus A – the set of perspectives where Curie correctly classifies the input as undermining but Tk-11B is wrong, and Corpus B – the set where TK-11B is correct while Curie is wrong. We found that Corpus B more often "*Uses language that is positive or uplifting*" ($V \approx 0.12$, AUCROC ≈0.67). One possible explanation is that Curie made many mistakes by misinterpreting undermining as a label for negative sentiment rather than a logical relation between the claim and the perspective.

Comparing lyrics from different eras. Compared to lyrics from the 70s, those from the 80s more often "*references violence or aggression*" ($V \approx 0.06$, AUCROC ≈ 0.58).

Describing distribution shift. We compared the premises from the SNLI dataset and MNLI dataset, and the former "*involves physical activity, such as walking, playing, climbing, or biking*" ($V \approx 0.13$, AUC-ROC ≈0.64). One possible explanation is that SNLI is based on image captions.

Comparing discussion topics between bots and human users. We compared the topical differences between tweets identified as written by bots vs. human users on Twitter, and our system finds that the bots more often "*contains keywords related to business, finance or trading*" ($V \approx 0.08$, AUC-ROC ≈ 0.61). One possible explanation is that bots are frequently used to generate finance-related scams.

Identifying temporal differences in news headlines. We compared headlines published by ABC news across different years. Compared to 2014, headlines from 2010 "*mention disasters and crimes, such as plane accidents and assaults*" more often ($V \approx 0.03$, AUCROC ≈ 0.53). Compared to year 2019, year 2020 more often "*discusses coronavirus-related topics*" ($V \approx 0.21$, AUCROC ≈ 0.65).

Describing text clusters. We present two example descriptions for text clusters. One from Wikipedia: "*references pop culture, such as movies, books, and television shows*." ($V \approx 0.21$, AUC-ROC ≈ 0.73); one from PoetryFoundation.com: "*uses vivid imagery and metaphors to convey a feeling*" ($V \approx 0.09$, AUC-ROC ≈0.65).

18 Limitations and Future Work

We still face many challenges in building a broadly useful system. We describe technical challenges that machine learning researchers can tackle in Appendix 18.1 and organizational challenges that require domain experts in Appendix 18.2.

18.1 Engineering Challenges

Hypotheses about the corpora might not be appropriate predicates on individual samples. When comparing highly rated definitions from `UrbanDictionary.com` to others, our system generates the hypothesis that the former "*is more likely to include slang or colloquial terms.*" This is a statement about a collection of text samples, but the validator requires the hypothesis h to be a predicate on individual text samples x. To address this, we used GPT-3 to automatically remove comparatives from the hypotheses, e.g. rewriting the hypothesis above to "*include slang or colloquial terms*."

However, some versions of this problem were harder to remove. For example, when comparing reviews from American Airlines (AA) flights and Delta Airlines to understand which aspects of each airline are doing better/worse, the proposer generated the hypothesis "*mentions **American Airlines'** staff being unfriendly and unhelpful*". Interpreted literally, this hypothesis can only be true on the corpus of AA reviews, since it presupposes the review to be about AA. The correct predicate for use on individual samples should instead be "*mentions staff being unfriendly and unhelpful*" (without the words "*American Airlines*"'). Therefore, future systems should explicitly convert corpus-level statements to their corresponding correct predicates, and the metrics should evaluate whether the validity of the predicates implies the corpus-level statements.

Beyond truth predicates. Our work requires the discovery to be a truth predicate that maps a text sample to a truth value. However, scientific discoveries can be arbitrary natural language expressions; extending to more flexible expressions requires a significant redesign of our system and evaluation framework. Some more feasible near-term extensions include 1) allowing natural language expressions that map from text samples to real values, e.g., "how polite the sentence is compared to other samples from the corpora" or 2) using additional logical forms to combine individual truth predicates; e.g., learn a shallow and interpretable decision tree where each split point is a natural language predicate.

Beyond corpus-level differences. Our work focuses on describing corpus-level differences and validates a discovery by comparing how often it is true on each corpus. Future work can consider other ways to validate a discovery: for example, suppose each text sample is associated with a continuous target variable, we can validate whether a discovery is more likely true if the target variable is large.

Clarifying a discovery. Some discoveries seem to have clear meanings on the surface, but they become ambiguous when we judge them on individual text samples. For example, judging whether a text sample $h =$ "*mentions people*" seems like an unambiguous task a priori; however, it is unclear whether it is true on the sample $x =$ "*I woke up this morning.*", since the "*people*" in h is a plural form, while x only mentions one person "*I*". Future work can use a language model to automatically clarify the meaning of a hypothesis and make it more specific, e.g., rewrite h as "*mentions one or more humans.*"

Correlation $\neq$ causation. Like other tools that rely on correlations to analyze patterns in data (e.g., linear regression), our system cannot establish causal relations either. For example, when comparing self-reported happy moments from females and males, even if the former corpus has more samples that "*mention children and family*", it does not necessarily imply family plays a more important role in inter-personal relations for females; an alternative hypothesis is that females might mention any other people more often than males, hence leading to the observation that they mention family more often. Future work can use language models to propose what control hypothesis to test.

Decreasing the cost of validation. As alluded to in Section **??**, estimating V is extremely expensive as it requires a lot of human labor. Future work can consider an importance sampling procedure that uses $\hat{T}$ as a proposer to improve the sample efficiency of estimating V.

Training a better proposer. We developed a self-supervised learning algorithm to propose more valid hypotheses. However, it does not take into account the meaningfulness metric, and it is unclear

how to manage its trade-offs with validity if they exist. We look forward to future works that can train a better proposer with as minimal supervision as possible.

Combining Meaningfulness and Validity Metrics. To simplify evaluation, we assumed meaningfulness to be independent of the magnitude validity V. Such an assumption allows us to directly evaluate hypotheses that are not necessarily valid but is also limiting for evaluating the final discoveries: for example, for that 2008 "*discuss economy*" more often than 2007, it would be way more significant if $V = 0.99$ compared to $V = 0.0000001$. Future works can propose better metrics that do not assume that validity and meaningfulness are independent.

18.2 Organizational Challenges

As discussed in Polanyi et al. (2000), it requires implicit community norms rather than explicit deductive logic to decide what counts as good research results; to guide our system to produce truly important discoveries, our system needs feedback from researchers who work in the domain of interest. However, except for machine learning, the authors do not have research expertise in most of the domains listed in Figure 2. We look forward to future contributions from other domains and list concrete directions below.

What problems to solve? We generated the problems in OpenD5 by reading relevant papers and guessing what domain experts might care about. However, our guesses can be inaccurate. Future works can directly gather problems from domain experts to reflect the actual usage of our system.

How to interpret a discovery? We asked for Turker's judgment to compute $T(h, x)$. However, many hypotheses require expert knowledge to interpret properly. For example, only law experts can reliably judge whether a contract x satisfies the predicate h "*contains a license grant that is irrevocable*." Domain experts are needed to evaluate the validity of a discovery and supervise the validator.

What discoveries are meaningful? Our work developed the evaluation instructions to approximately evaluate what hypotheses are meaningful. However, just as no one can become an outstanding peer reviewer simply by reading the review guideline, we do not consider it feasible to provide a gold evaluation simply by reading our instructions. Whether a discovery is meaningful depends heavily on implicit community norms, and we hope domain experts can provide better evaluation and training signals for our system.

19 Self-Supervised Learning with Open-Ended Problems: A Proof of Concept

Since the problems in OpenD5 are open-ended, our system could potentially produce discoveries with higher validity scores than our current system. Therefore, we design a self-supervised learning algorithm to improve an LM's ability to propose more valid hypotheses, using the principle that **it is easier to validate a discovery than to generate one**.

Algorithm. Suppose we are given a set of problems for training and an initial language model m_{init}. Our goal is to automatically generate a set of *prompt-completion* pairs to fine-tune m_{init} so that it can propose hypotheses that are more valid. To generate a *prompt*, we randomly sample a problem and create a proposer prompt following the procedure in Section 4.1. To generate the desired *completion* given a prompt, we sample multiple hypotheses from m_{init}, approximate their V' score on the samples in the proposer prompt with the same language model m_{init} (Section 4.2), and select the highest scoring hypothesis. Finally, we use the prompt-completion pairs to fine-tune m_{init}.

However, since we cannot fine-tune instruction-tuned GPT-3, we can only experiment with Flan-T5 (Chung et al., 2022), an open-sourced instruction-tuned model that might only work well for easier "mini-problems". As a proof of concept, we tested our algorithms for describing groups of four samples, where each group comes from a text cluster. As an overly simplified example, we will give the LM the prompt "*Group A: 1. dog 2. cat 3. pig 4. cow. Group B: 1. phone 2. laptop 3. desk 4. cup*" as an input and the LM can output "*mentions an animal*" as a hypothesis.

Data. We created 33 corpora by merging all corpora in OpenD5 with the same domain, and automatically generated 4503 text clusters using RoBERTa embeddings (Aharoni & Goldberg, 2020). We focused on clustering because it can automatically generate a large amount of semantically coherent groups of samples. To create a pair of four samples, we randomly sampled a corpus, sampled two clusters within that corpus, and took four random samples from each cluster. To test

cross-corpus generalization, we reserved 28 of the 33 corpora to create mini-problems for evaluation, using the rest for training. We used Flan-T5 (Chung et al., 2022) as m_{init} and sampled hypotheses with a temperature of 0.8. For training, we sampled 30,000 mini-problems and selected the best of eight hypotheses generated by m_{init} as the target completion; for evaluation, we sampled 200 mini-problems to calculate V with Turkers and 1500 mini-problems to calculate V' automatically.

Results. We evaluated randomly sampled hypotheses from the language model before and after self-supervised training. The automated "self-evaluation" validity score V' improves substantially from 0.22 to 0.37, and the "true" validity score V according to Turker evaluation improves from 0.07 to 0.10, with a p-value of 0.02. This result provides preliminary evidence that our algorithm (or similar variants) could be applied to a large set of problems to improve the validity of the hypotheses; we expect future validators to simulate human judgments better, hence decreasing the approximated gap of improvement between V and V'.

20 Comparing D5 to Naïve Bayes

We qualitatively compare the discovery generated by our D5 system to the top-5 unigram features extracted by Naive Bayes, a traditional exploratory analysis method. The Naive Bayes method is effective when the target difference can be saliently reflected by individual words. For example, "*yo*" implies a rap genre, "*die*" implies a language of Deutsch, and ["*rank*", "*higher*", "*univeristy*"] hints at the topic of "*college ranking changes*". Additionally, compared to black-box neural networks, such a method is fully interpretable.

In comparison, D5 can directly generate a semantically coherent description for the target difference, saving users' time to guess the underlying correlation by inspecting the top unigram features. Additionally, it can capture differences that are hard to detect at a word level; for example, "*the genre of biblical scripture*" is mainly reflected in its sentence structure rather than individual words. Finally, D5 only describes goal-related differences, while Naïve Bayes picks up on any discriminative feature; for example, when identifying the topical differences between a English and a Deutsch corpus, Naïve Bayes fails catastrophically and only picks up common determiners such as "*the*" or "*die*" instead of topic words, since they are the most useful feature at telling which sample comes from which corpus. Given the respective strength of D5 and traditional exploratory methods, we envision D5 to serve as a complementary method to traditional methods.

21 Annotation Interface to Collect Human-Generated Hypotheses

(This section describes an interesting research direction we did not have time to fully pursue.)

Task. To fine-tune the language model to propose better hypotheses and perform validation more accurately, we also designed an interface to collect human annotations earlier in the project. In this annotation task, the annotators see five text samples from each of the two corpora; they then write one or many natural language predicate(s) that describe how samples from the two groups are different and choose which text samples satisfy each predicate the annotator has written. Since it is challenging for humans to identify systematic differences between even groups of five sentences, we made the task easier for them by

- we chose the representative samples from each corpus to form the two groups of samples, similar to the process in Section 14, and
- we highlighted subspan of the text samples that are informative for how the two corpora differ. For example, if Corpus A is sports related while Corpus B is entertainment related, we hope to highlight sports-related words like "basketball". To automatically identify the text spans to highlight, we fine-tuned RoBERTa to classify whether a sample comes from Corpus A and Corpus B, used the SHAP library to calculate how much each text span influences the classifier's decision, and highlighted the text spans based on the influence.

A screenshot of the annotation interface can be seen in Figure 6.

Preliminary Results We performed initial experiments on text clusters formed on the wikitext-2 dataset (Merity et al., 2016). We asked the authors to write hypotheses for 30-50 samples and then compare the results with GPT-3 generated hypotheses. We found that human annotators were able to

Figure 6: A detailed screenshot of our annotation interface.

write 2-4 valid hypotheses per pair of text groups, while GPT-3 text-davinci-003 was able to generate 4-6. Out of the valid generated hypotheses, approximately a third were variations on another valid hypothesis. The number of times humans were able to write a hypothesis that GPT-3 was unable to generate was around a third of the samples, while GPT-3 was able to generate a novel hypothesis humans have not thought about before in nearly every single text corpora. Given that GPT-3 is close to our author's ability to write hypotheses, we estimated that we would not be able to fine-tune T5 to propose better hypotheses with human annotations, and hence gave up on this research direction.

22 Datasets

Many of our datasets come from the following sources: the Computational Models of Social Meaning class from Columbia University`http://www1.cs.columbia.edu/~smara/teaching/S18/`, the ACL Anthology `https://aclanthology.org`, , and Kaggle datasets with an NLP tag. `https://www.kaggle.com`

abc-headlines. We collect headlines published by ABC news, an American news company from Kulkarni (2018). ABC headlines are directly downloaded from Harvard Dataverse. The year is extracted from the publication date field. Samples are constructed from the headline text. The data is downloadable from `https://doi.org/10.7910/DVN/SYBGZL` with license CC0 1.0.

ad-transcripts. We collect ad scripts from a variety of industries from Hartman (2019). Ad transcripts are directly downloaded from Kaggle. The top eight industries by frequency are selected. Newlines are replaced with spaces. The dataset is downloadable from https://www.kaggle.com/datasets/kevinhartman0/advertisement-transcripts-from-various-industries with license CC0 Public Domain.

admin-statements. We collect statements of administration policy from American presidents from Progress (2022). Administration statements are extracted from a collection hosted on GitHub. Extraneous symbols are removed and samples are split by paragraph. The dataset is downloadable from https://github.com/unitedstates/statements-of-administration-policy#statements-of-administration-policy and origin files have a Creative Commons Attribution 3.0 License.

ai2-natural-instruction. We collect a learning-from-instructions dataset released by the Allen Institute for AI from Mishra et al. (2022). Natural instruction tasks are directly downloaded without modification. The dataset is released under an Apache-2.0 license.

airline-reviews. We collect reviews of airlines collected from the review website Skytrax. Airline reviews for airlines, airports, and seats are downloaded from a public GitHub repository. Names of aircraft, airlines, countries, and traveler types are standardized. Ratings of 1, 4, or 5 on a scale of 5, and 1, 5, 8, or 10 on a scale of 10 are kept. This dataset can be downloaded via https://github.com/quankiquanki/skytrax-reviews-dataset.

aita. We collect posts on the "Am I The Asshole" Subreddit, an online forum people ask others whether they were in the wrong from O'Brien (2020). Posts from r/AmITheAsshole are downloaded from a praw scrape of Reddit. Topic areas are chosen based on common themes in posts and coarsely defined based on manual keywords. Each post can belong to multiple topic areas. The dataset can be downloaded at https://doi.org/10.5281/zenodo.3677563.

all-the-news. We collect news articles collected from various outlets between 2015 and 2017 from Thompson (2019). News articles are downloaded directly from the Components website. The titles are used as text samples.The dataset can be downloaded at https://components.one/datasets/all-the-news-articles-dataset.

amazon-reviews. We collect Amazon reviews collected from various product categories from Ni et al. (2019). Amazon reviews are downloaded from a 2018 crawl of the website. The first 100,000 review texts are treated as the text sample. The dataset can be downloaded at https://nijianmo.github.io/amazon/index.html.

armenian-jobs. We collect job postings in Armenia from Udacity (2017). The Armenian job postings dataset is downloaded from a snapshot on GitHub. Different IT jobs are manually coded and time intervals are defined in order to balance sample availability. The dataset can be downloaded at https://www.kaggle.com/datasets/udacity/armenian-online-job-postings.

boolq. We collect a reading comprehension dataset of yes/no questions from Clark et al. (2019). Boolean questions are downloaded directly as is. The dataset can be downloaded at https://github.com/google-research-datasets/boolean-questions with license CC-SA-3.0.

clickbait-headlines. We collect headlines across time from the Examiner, a clickbait news site from Kulkarni (2020a). The Examiner headlines are directly downloaded from Kaggle. The year is extracted from the publication date field. Samples are constructed from the headline text. The dataset can be downloaded at https://www.kaggle.com/datasets/therohk/examine-the-examiner, with license CC0: public domain.

convincing-arguments. We collect arguments on a variety of topics annotated for convincingness from Habernal & Gurevych (2016). Annotated arguments are downloaded from the GitHub repository. Arguments are sorted by rank. The bottom 400 are treated as "unconvincing", the top 200 are treated as "convincing", and the next 200 are treated as "somewhat convincing." The dataset can be downloaded at https://github.com/UKPLab/acl2016-convincing-arguments, with license CC-BY 4.0.

craigslist-negotiations. We collect dialogue from Craigslist negotiations, an online seller platform from He et al. (2018). Craigslist negotiations are downloaded from Huggingface. Sequences which contained a "quit" intention or "reject" intention are categorized as failures; those which contained an "accept" intention are categorized as successes. The mid-price is defined as the mean

price of the items sold. Within each category, the items are sorted by mid-price. The top half is treated as high-price and the bottom half is treated as low-price. This dataset can be downloaded at https://huggingface.co/datasets/Hellisotherpeople/DebateSum with MIT license.

debate. We collect evidence compiled for American competitive policy debate, published online by debate camps from Roush & Balaji (2020). The train split is downloaded from Huggingface. For each sample, we use the abstract as the text. Arguments are categorized by type, debate camp of origin, and topic/specific argument. For topics, we use domain knowledge to list relevant keywords for each topic and include any sample with a file name that includes any keyword. A single sample can belong to multiple topics. This dataset can be downloaded at https://huggingface.co/datasets/Hellisotherpeople/DebateSum with MIT license.

dice-jobs. We collect American technology job postings on dice.com from PromptCloud (2017). Job postings are downloaded from Kaggle. Posts from the six most popular companies are categorized by company. We remove miscellaneous characters and blank descriptions. We additionally apply our splitting procedure to reduce description length. This dataset can be downloaded at https://www.kaggle.com/datasets/PromptCloudHQ/us-technology-jobs-on-dicecom under CC BY-SA 4.0 .

diplomacy-deception. We collect dialogue from games of Diplomacy, which involves deception from Peskov et al. (2020). Diplomacy dialogues are downloaded from GitHub (all splits). The data are ASCII encoded and newlines are removed. Each message and label is treated as a sample. This dataset can be downloaded at https://huggingface.co/datasets/diplomacy_detection under unknown license.

echr-decisions. We collect facts of cases heard before the European Court of Human Rights from Chalkidis et al. (2019). Decisions are downloaded from a public archive. A random sample of 500 decisions is selected from the files. The samples with any violated articles are categorized as "violation," while the rest are categorized as "no violation." This dataset can be downloaded at https://paperswithcode.com/dataset/echr under unknown license.

essay-scoring. We collect essays from students from ess (2012). Essays are downloaded from a GitHub repository. Only essays from set 5 are considered. Essays with a score of at least 3 are categorized as good essays, while essays with a score less than 3 are bad essays. This dataset can be downloaded at https://www.kaggle.com/c/asap-aes under unknown license.

fake-news. We collect fake and legitimate news from Pérez-Rosas et al. (2017). Fake news articles are downloaded from the author's website. Full articles are treated as text snippets. This dataset can be downloaded at http://web.eecs.umich.edu/~mihalcea/downloads.html#FakeNews under CC-BY-4.0.

fomc-speeches. We collect Federal Open Market Committee (FOMC) speeches from 1996-2020, which describe Federal Reserve policy from Mish (2020). Fed speeches are downloaded from Kaggle. The macro indicator data are merged in on the year and month. Full speech text is split by paragraph and categorized by speaker, year, and macroeconomic indicator. This dataset can be downloaded at https://www.kaggle.com/datasets/natanm/federal-reserve-governors-speeches-1996-2020 under unknown license.

genius-lyrics. We collect lyrics collected from Genius.com before 2020 from Lim & Benson (2021). Genius lyrics are downloaded from Google Drive. The lyrics are merged with song metadata and treated as samples. We categorize lyrics by hand-selecting popular artists, common genres, time periods, and view counts (over 1M views is high, 500k-1M is medium). This dataset can be downloaded at https://www.cs.cornell.edu/~arb/data/genius-expertise/ under unknown license.

happy-moments. We collect self-reported happy moments and demographic characteristics from Asai et al. (2018). The HappyDB dataset is downloaded from the official GitHub repository. Demographic data is cleaned and merged into happy moments. Happy moment descriptions are treated as samples and are categorized by type of happy moment, country of origin, and other demographic features. This dataset can be downloaded at https://github.com/megagonlabs/HappyDB under unknown license.

huff-post-headlines. We collect headlines from the news outlet Huffington Post from Misra & Arora (2019) and Misra & Grover (2021). Huffington Post headlines are downloaded from Kaggle. The

short description of each article is treated as a sample and tokenized at the sentence level. This dataset can be downloaded at https://rishabhmisra.github.io/publications/ under CC-BY-4.0.

immigration-speeches. We collect congressional and presidential speeches that mention immigration from 1880 to the present from Card et al. (2022). Immigration speeches are downloaded from the replication package. The speech text is preprocessed to remove extraneous spaces. We engineer features corresponding to time periods, well-known speakers, other significant time periods, the racial group under discussion, and the geographic area within the United States. This dataset can be downloaded at https://github.com/dallascard/us-immigration-speeches/releases.

kickstarter. We collect names of startups on kickstarter.com from Mouillé (2017). We download a 2018 crawl from Kickstarter from Kaggle. The project name is treated as the text sample. This dataset can be downloaded at https://www.kaggle.com/datasets/kemical/kickstarter-projects?select=ks-projects-201612.csv under CC BY-NC-SA 4.0.

microedit-humor. We collect funny sentences generated by making one-word edits to normal statements from Hossain et al. (2019). The Microedit dataset is downloaded from the author's website. We make the relevant edit to each text sample and treat the edited text sample as the data point. We bin the mean annotator grade into 4 and denote each as unfunny, neutral, funny, and very funny, respectively. This dataset can be downloaded at https://paperswithcode.com/dataset/humicroedit.

mnli. We collect a collection of sentence pairs annotated with textual entailment information from a range of genres from Williams et al. (2017). The MNLI corpus is downloaded from the official website. We treat the premise and hypothesis as text samples. This dataset can be downloaded from https://cims.nyu.edu/~sbowman/multinli/, most of which are under the OANC license.

monster-jobs. We collect American job postings on monster.com. Jobs on Monster.com are downloaded from Kaggle. Job descriptions are treated as samples and split at the paragraph and sentence level. We keep and categorize jobs from seventeen large cities. This dataset can be downloaded from https://www.kaggle.com/datasets/PromptCloudHQ/us-jobs-on-monstercom under CC BY-SA 4.0 .

movie-tmdb. We collect movie plot summaries from TMDB from Kaggle (2018). TMDB movie overviews are downloaded from Kaggle. We keep only English movies and bin popularity by deciles. The top decile is considered "hits," the 70-80th percentiles are considered "average," and the 30-40th percentiles are considered "bad." This dataset can be downloaded from https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata21.

movie-wiki. We collect movie plot summaries collected from Wikipedia from Robischon (2019). Wikipedia movie summaries are downloaded from Kaggle. This dataset can be downloaded from https://www.kaggle.com/datasets/jrobischon/wikipedia-movie-plots under CC BY-SA 4.0.

news-popularity. We collect news headlines posted on social media platforms from Moniz & Torgo (2018). Headlines are downloaded from a reproduction package. The headline and title text are cleaned, and the title is treated as the text sample. The 100 most positive and negative or popular and unpopular articles on each topic are used as distributions. This dataset can be downloaded from https://archive.ics.uci.edu/ml/datasets/News+Popularity+in+Multiple+Social+Media+Platforms.

nli-benchmarks. We collect training examples from various natural language inference (NLI) datasets from Liu et al. (2022). NLI benchmarks are downloaded from a public collection on Google Drive. We examine the premise and hypothesis separately as samples. This dataset can be downloaded from https://github.com/alisawuffles/wanli.

npt-conferences. We collect Non-Proliferation of Nuclear Weapons (NPT) conference transcripts from Barnum & Lo (2020). NPT conference notes are extracted from the accompanying replication package. Text is split by paragraph, and only paragraphs longer than 50 characters are preserved. Text is split into three time ranges: pre-2008, 2008-2012, and post-2012. This dataset can be downloaded from https://journals.sagepub.com/doi/full/10.1177/0022343320960523.

open-deception. We collect arbitrary lies and truths from any domain generated by crowdworkers from Pérez-Rosas & Mihalcea (2015). Open domain lies are downloaded from the public dataset

and lie texts are split into lies and truths. This dataset can be downloaded from https://web.eecs.umich.edu/~mihalcea/downloads.html#OpenDeception.

open-review. We collect submissions to ICLR, a machine learning conference from 2018 to 2021. Open review abstracts are accessed via the openreview API. We query for abstracts from the 2018-2021 ICLR blind submissions. Abstracts are classified based on rating: $>= 7$ ("great"), 5-6 ("good"), and $<= 4$ ("bad"). This dataset can be downloaded from https://openreview.net/.

parenting-subreddits. We collect posts from various parenting-related subreddits, which are text-based forums on the site Reddit from Gao et al. (2021). Posts from various subreddits are downloaded from the paper's GitHub repository. We clean the text and split the posts according to the topic(s) each post is tagged with. This dataset can be downloaded from https://github.com/SALT-NLP/Parenting_OnlineUsage.

poetry. We collect poems from PoetryFoundation.com from Bramhecha (2019). Poems are downloaded from a 2019 scrape of the PoetryFoundation website from Kaggle. The text is cleaned and split according to subject tags and authorship. This dataset can be downloaded from https://www.kaggle.com/datasets/tgdivy/poetry-foundation-poems under GNU Affero General Public License.

political-ads. We collect political ads observed by Facebook users from pol (2021). Ads are downloaded from the Ad Observer website, which maintains an aggregate of all collected ads. We extract targeting metadata from the targeting field and define splits according to age, gender, location, interests, time, and political lean. This dataset can be downloaded from https://adobserver.org/ad-database/.

qqp. We collect questions from Quora.com from Quora (2017).

rate-my-prof. We collect reviews of lecturers from RateMyProfessor.com from He (2020). We download a sample of RateMyProfessor.com reviews from an online repo. We clean the text and guess the gender of the reviewed lecturer from the first name using the gender-guesser package. Due to data availability, we consider only male and female names. To improve the quality of the classification, we remove any posts which use pronouns from the opposing sex (e.g. "him"). This dataset can be downloaded from https://data.mendeley.com/datasets/fvtfjyvw7d/2 under CC BY 4.0 .

radiology-diagnosis. We collect impressions and medical histories of radiology patients from Pestian et al. (2007). Radiology diagnoses are downloaded from a GitHub copy of the original task dataset. We parse the metadata to retrieve the diagnostic code, decision type, impression, and patient history. Referencing the associated ICD codes, we convert codes to colloquial diagnoses (e.g. 786.2 denotes cough). We treat the histories and impressions as samples and split them according to diagnosis and level of consensus.

reddit-humor. We collect jokes posted on the Reddit forum r/Jokes, a message board for sharing jokes from Weller & Seppi (2020). Jokes are downloaded from the dev and test splits of the dataset. We clean the text and split the dataset according to whether they are labeled as funny. This dataset can be downloaded from https://github.com/orionw/rJokesData under Reddit License and Terms of Service, and users must follow the Reddit User Agreement and Privacy Policy, as well as remove any posts if asked to by the original user.

reddit-stress. We collect stress-related posts on Reddit from Turcan & McKeown (2019). We split the post text based on which subreddit they are posted on (related to PTSD, anxiety, or stress generally). Reddit posts are downloaded from https://github.com/gillian850413/Insight_Stress_Analysis, and we recommend following the Reddit User Agreement and Privacy Policy, as well as remove any posts if asked to by the original user.

reuters-authorship. We collect articles from various Reuters authors from Liu (2011). The articles are split according to the author. Reuters articles are downloaded from the UCI repository https://archive.ics.uci.edu/ml/datasets/Reuter_50_50.

riddles. We generated several riddles. The 3000 most common English words are manually copied from a website. Words with between 5 and 8 characters are kept. We create two popular riddles. First, we split words based on whether they have a duplicate character. We exclude any words with multiple "doubles" or more than 2 of any character. Second, we split words based on whether they have the letter T.

scotus-cases. We collect facts from cases heard by the Supreme Court of the United States (SCOTUS) from Alali et al. (2021). Supreme Court cases are downloaded from a GitHub repository. We identify state/federal parties by manually defining keywords. We split based on the winning party, the identity of each party, and the type of decision. We then define several time periods and relevant political eras and split decisions accordingly. Finally, we split according to the ruling's policy area and how it changes over time. The dataset can be downloaded from https://paperswithcode.com/paper/justice-a-benchmark-dataset-for-supreme-court under CC-BY-SA.

short-answer-scoring. We collect short answers from students from sho (2013). Short answers are downloaded from a GitHub mirror of the dataset. We consider only responses to essay set 1. The two scores are averaged and binned into good (>= 2.5), medium (1.5-2.5), and bad (<1.5). The dataset can be downloaded from https://www.kaggle.com/c/asap-sas.

snli. We collect a collection of sentence pairs annotated with textual entailment information from images from Bowman et al. (2015). The dataset can be downloaded from https://nlp.stanford.edu/projects/snli/ under CC BY-SA 4.0.

squad-v2. We collect reading comprehension questions crowdsourced from Wikipedia articles from Rajpurkar et al. (2018). The dataset can be downloaded from https://rajpurkar.github.io/SQuAD-explorer/ under CC BY-SA 4.0.

stock-news. We collect top news headlines on Reddit, an online message board from Sun (2017). Headlines are downloaded from a GitHub mirror. We clean the text and divide the samples based on whether the DOW rose or fell that day. The dataset can be downloaded from https://github.com/ShravanChintha/Stock-Market-prediction-using-daily-news-headlines under Reddit License and Terms of Service, and users must follow the Reddit User Agreement and Privacy Policy, as well as remove any posts if asked to by the original user.

suicide-notes. We collect posts from r/SuicideWatch and r/depression, two forums on Reddit fromHe (2021). The post title and body are combined to form the text samples. Samples are split based on whether they were posted in a suicide-related Subreddit. The dataset can be downloaded from a github: https://github.com/hesamuel/goodbye_world, under Reddit License and Terms of Service, and users must follow the Reddit User Agreement and Privacy Policy, as well as remove any posts if asked to by the original user.

times-india-headlines. We collect headlines from Times of India news from Kulkarni (2022). Headlines are downloaded from a Dataverse mirror. We use the first 1000 headlines in each year as samples. The dataset can be downloaded from https://www.kaggle.com/datasets/therohk/india-headlines-news-dataset under CC0 Public Domain.

trial-deception. We collect testimonies from witnesses in real trials from Pérez-Rosas et al. (2015). Trial testimonies are downloaded from the author's website. The testimonies are divided based on whether they are considered truthful. The dataset can be downloaded from https://web.eecs.umich.edu/~mihalcea/downloads.html#RealLifeDeception.

un-debates. We collect speeches from debates at the United Nations from Baturo et al. (2017). Debate transcripts are downloaded from the Dataverse reproduction package. Samples are divided based on the country and year of the snippet. First, we isolate samples from Russia, China, and the United States and specify 3 time periods of interest. Next, we divide all samples by the decade. Finally, we create distributions for 19 countries of interest. The dataset can be downloaded from https://doi.org/10.7910/DVN/0TJX8Y under CC0 1.0 .

unhealthy-conversations. We collect expert-annotated unhealthy conversations from Price et al. (2020). Conversation transcripts are downloaded from the official GitHub repository. For each annotated attribute, we split the dataset based on whether that form of unhealthy conversation is present in the sample. The dataset can be downloaded from https://github.com/conversationai/unhealthy-conversations under CC BY-NC-SA 4.0.

urban-dictionary. We collect definitions from UrbanDictionary.com, a crowdsourced English dictionary from Kulkarni (2020b). Urban Dictionary entries are downloaded from Kaggle. Definitions are split into groups representing the top 1, 5, and 10 percent of definitions ranked by both upvotes and downvotes; we sample 10,000 from each and create a control distribution by randomly sampling 10,000 definitions from all entries. The dataset can be downloaded from https://www.kaggle.com/therohk/urban-dictionary-words-dataset under CC0 Public Domain.

wikitext. We collect text snippets from Wikipedia from Merity et al. (2016). The Wikipedia snippets are loaded from HuggingFace. We remove any samples that are empty or start with '=' (which represent headings); samples are tokenized at the sentence level and used for clustering. The dataset can be downloaded from `https://huggingface.co/datasets/wikitext` under CC BY-SA 3.0.

yc−startups. We collect descriptions of companies that were part of the Y Combinator startup incubator from Bhalotia (2022). YCombinator company descriptions are downloaded from a 2022 scrape on GitHub. Only companies with long descriptions are preserved. Companies are split according to founder characteristics, year, "top company" designation, operating status, and location. The dataset can be downloaded from `https://www.kaggle.com/datasets/benhamner/y-combinator-companies`.