Polling Latent Opinions: A Method for Computational Sociolinguistics Using Transformer Language Models

Anonymous ACL submission

Abstract

Text analysis of social media for sentiment, topic analysis, and other analysis depends initially on the selection of keywords and phrases that will be used to create the research corpora. However, keywords that researchers choose may occur infrequently, leading to errors that arise from using small samples. In this paper, we use the capacity for memorization, interpolation, and extrapolation of Transformer Language Models such as the GPT series to learn the linguistic behaviors of a subgroup within larger corpora of Yelp reviews. We then use prompt-based queries to generate synthetic text that can be analyzed to produce insights into specific opinions held by the populations that the models were trained on. Once learned, more specific sentiment queries can be made of the model with high levels of accuracy when compared to traditional keyword searches. We show that even in cases where a specific keyphrase is limited or not present at all in the training corpora, the GPT is able to accurately generate large volumes of text that have the correct sentiment.

1 Introduction

Large-scale research involving humans is difficult, and often relies on labor-intensive mechanisms such as polling, where statistically representative populations will be surveyed using landline and cellphone interviews, web surveys, and mixed-mode techniques that combine modes. Often, participants in a survey may need to be recontacted to update responses as a result of changing events (Fowler Jr, 2013).

As social media has developed, many attempts have been made to determine public opinion by mining data that is available from online providers such as Twitter and Reddit, e.g. (Colleoni et al., 2014; Sloan et al., 2015). However, though social data can be analyzed in a variety of ways, it cannot replace the pollster asking about items that do not explicitly exist in the data.

Due to the the emergence of Transformer-based Language Models (TLMs) this may be ready to change. These models, such as the Generative Pre-trained Transformer (GPT) series developed by OpenAI, have been trained on millions of high-quality web pages. The model generates text (with words represented as high-dimensional vectors) as a function of an input sentence and its previous hidden states. These hidden states are in turn functions of the input sentence and the previous hidden states, etc. Since the model is not trained using any hand-crafted rules about language, it effectively learns its own set of rules for generating natural language. In doing so, it captures semantic and syntactic regularities in language that were previously hidden to traditional statistical methods.

As such, these models contain tremendous amounts of sociolinguistic information such as the patterns of speech, word choice, and dialects (Meyerhoff, 2018) that can be accessed *interactively*. By using carefully constructed probes, the model can be "polled", allowing researchers to access latent knowledge contained in the model in ways that are analogous to accessing the knowledge in a population by calling a random sample.

Large transformer-based language models such as the GPT encode short-range and long-range statistical relationships between tokens. Tokens represent words, and clusters of tokens represent more nebulous structures that incorporate *concepts* and *beliefs*. A key point in understanding how these models behave are the concepts of *interpolation* and *extrapolation* (Rytting and Wingate, 2021). Consider the following example using the GPT-3, where a prompt (in bold face) describing a fictional bird, a "Floober", is used to create a contextually consistent description of this imaginary beast:

> ***The Floober is a large flightless bird that lives in semi-arid areas in Africa. Like its cousin the Ostrich****, the Floober is a member of the Struthionidae family.*

The Floober was discovered in the early 19th century by explorers and naturalists like William Cornwallis Harris, who named them after the sound they made: "Floober". They were first described in 1821 by Thomas Burchell, the discoverer of Burchell's zebra.

In this example, the GPT-3 is able to *interpolate* by arranging tokens in the model's information space related to African zoology based on the attention relationship relationship to the given prompt. The GPT also *extrapolates* from the prompt cue of "Floober" and "Ostrich" by accessing the concept of *Struthionidae*, which include ostriches. These relationships are encoded as statistical dependencies among tokens, which means that when a token is missing from a query, the model can use its contextual knowledge to predict which other tokens should be included. This does not mean that the GPT-3 is foolproof. In this case, it makes a factual error by accessing tokens related Thomas Burchell (1799–1846) [1] rather than William John Burchel (1781 – 1863)[2], who was the first Westerner to describe the zebra for science.

Because of this ability to synthesize responses, language models such as GPTs can provide capabilities for capturing the human opinions and beliefs encoded in the training text that more resemble the traditional polling model. Rather than performing training data analysis (e.g., supervised classification), we can *poll* the model's responses to probes. But to do this effectively requires that we develop methods to systematically reveal the relevant information captured in these models.

In this paper, we finetune (Sun et al., 2019) a set of GPT-2 models on a Yelp corpora that reflect populations of users with distinctive views. We then use prompt-based queries to probe these models to reveal insights into the biases and opinions of the users. We demonstrate how this approach can be used to produce results more accurately than traditional keyword or keyphrase searches, particularly when data is sparse or missing.

In addition to the concepts of interpolation and extrapolation, we introduce the concept of language model *memorization*, where models can be trained to incorporate repeating patterns. We incorporate this concept by introducing the technique of *meta-wrapping*, which adds information to the training corpora that aids in the automated identifying of particular parts of the generated text. We further find a correlation of when the model is trained sufficiently to accurately reproduce these wrappings and the overall accuracy of the model in representing the explicit and latent information that it has been trained on.

Lastly, we provide methods for validating transformer language models in each of these contexts. We extensively study our methodology on Yelp data, where we have ground truth in the form of user-submitted stars, and discuss applications in other domains.

2 Related Work

Since the introduction of the transformer model in 2017, TLMs have become a field of study in themselves. The transformer uses self attention, where the model computes its own representation of its input and output (Vaswani et al., 2017). So far, significant research has been in increasing the performance of these models, particularly as these systems scale into the billions of parameters, e.g. (Radford et al., 2019). Among them, BERT (Devlin et al., 2018) and GPT (Radford et al., 2018) are two of the most well known TLMs used widely in boosting the performance of diverse NLP applications.

Understanding how and what kind of knowledge is stored in all those parameters is becoming a sub-field in the study of TLMs. Among them, (Petroni et al., 2019) used probes that present a query to the mode as a cloze statement, where the model fills in a blank (e.g. "Twinkle twinkle _____ star"). Research is also being done on the creation of effective prompts. Published results show that mining-based and paraphrasing approaches can increase effectiveness in masked BERT prompts over manually created prompts (Jiang et al., 2020). For example, mined prompts can be produced by mining phrases in the Wikipedia corpus that can be generalized as template questions such as *x was born in y* and *capital of x is y*. These can then be filled in using sets of subject-object pairs. Improvements using this technique can be substantial, with improvements of 60% over manual prompts. Paraphrasing, or the simplification of a prompt using techniques such as back-translation can enhance these results further (Jiang et al., 2020).

Using TLMs to evaluate social data is still nascent. A study by (Palakodety et al., 2020) used

[1] en.wikipedia.org/wiki/Thomas_Burchell
[2] en.wikipedia.org/wiki/William_John_Burchell

BERT fine tuned on YouTube comments to gain insight into community perception of the 2019 Indian election. They created weekly corpora of comments and constructed a tracking poll based on the prompts "Vote for MASK" and "MASK will win" and then compared the probabilities for the tokens for the parties BJP/CONGRESS and candidates MODI/RAHUL. The results substantially tracked traditional polling.

Lastly, we cannot ignore the potential dangers of TLMs. OpenAI has shown that the GPT-3 can be "primed" using "few-shot learning" (Brown et al., 2020). In their paper *The radicalization risks of GPT-3 and advanced neural language models* (McGuffie and Newhouse, 2020),the GPT-3 was primed using mass-shooter manifestos with chilling results. We will discuss these and other related issues in the ethics section.

3 Methods

For all the research involving finetuning, we used the Huggingface (Wolf et al., 2019) 117M parameter GPT-2 model. This was done for two reasons:

1. Increased speed: During the course of this study, we finetuned 48 models. We were able to finetune a model in 2-3 hours using one NVidia TITAN RTX.
2. Reduced carbon footprint: It is clearly possible to train larger models using more hardware in the same amount of time, but since this was a *comparative* study, there was no need to add the cost and energy of spinning up a multi-GPU cloud instance.

Our methods focus on understanding the *memorization*, *interpolation*, and *extrapolation* behaviors of these language models. To do this, we made use of the Yelp Open Dataset[3]. The Yelp dataset contains reviews of different businesses by customers. It incorporates social-media-like text, locations, business names, and star reviews, which can serve as a form of ground truth for performing sentiment analysis on review text. More specifically, we created specific sets of corpora for these GPT behaviors:

- *Memorization – Ratings and votes*: This corpora includes numeric information only, including stars and votes. This data was used to evaluate the *Global* characteristics of the model.
- *Interpolation – Reviews with stars*: This corpora includes a review and the associated stars. We evaluate the star rating and its relationship to the review text in the ground truth and generated data. This is used to evaluate the *Local* characteristics of the model.
- *Extrapolation – Masked reviews*: This corpora is trained using the same set of reviews as the previous item, only without any review that contains the phrase "vegetarian options". It is used to compare the behavior of the model in zero-shot situations when compared to ground truth and the model trained using the masked data.

For the purposes of our research, we concentrate on reviews of *American* restaurants. At 1,795,036 reviews, this subset is more than three times larger than Italian, the next most common cuisine. This provided us with the widest spectrum of options with respect to sub-queries of ground truth.

The overall technique used to create models, then generate and evaluate results is as follows:

1. Download and store the Yelp dataset in a MySQL database.
2. Analyze number of reviews by category.
3. Create a corpora, wrapping with meta-information (e.g. Figure 1).
4. Fine-tune models, using the Huggingface API.
5. Evaluate the model on a set of prompts and store the results. Each experiment contains an id, date, description, model name, list of textual probes, seed, and model hyperparameters.
6. Calculate sentiment and parts-of-speech analysis on generated text[4]. We also ran the same sentiment evaluation on a subset of "ground truth" reviews taken from the Yelp dataset.
7. Generate charts by running queries on the database and performing analytics.

We trained and evaluated three sets of models. The first sets were trained exclusively on stars and votes (See training corpora example in Figure 1). This was used to evaluate the statistical properties of the GPT against well-characterized numeric data.

[3] www.yelp.com/dataset

[4] github.com/flairNLP/flair

```
stars = 4.0, useful_votes = 0, funny_votes = 0, cool_votes = 0
stars = 5.0, useful_votes = 1, funny_votes = 0, cool_votes = 1
stars = 4.0, useful_votes = 0, funny_votes = 0, cool_votes = 0
stars = 2.0, useful_votes = 0, funny_votes = 0, cool_votes = 0
stars = 4.0, useful_votes = 1, funny_votes = 1, cool_votes = 1
stars = 4.0, useful_votes = 1, funny_votes = 0, cool_votes = 0
stars = 5.0, useful_votes = 2, funny_votes = 3, cool_votes = 3
stars = 5.0, useful_votes = 2, funny_votes = 0, cool_votes = 1
stars = 3.0, useful_votes = 0, funny_votes = 0, cool_votes = 0
stars = 3.0, useful_votes = 0, funny_votes = 0, cool_votes = 1
```

Figure 1: Corpus section with meta-wrapping

```
review: This place used to be a cool, chill place. Now its a bunch of neanderthal
bouncers hopped up on steroids acting like the can do whatever they want. There are
so many better places in davis square where they are glad you are visiting their
business. Sad that the burren is now the worst place in davis., stars: 1.0 --

review: Probably one of the better breakfast sandwiches I've ever had. I had the
EGGMEATMUFFIN, the bread was toasted perfectly and the bacon was a real thick cut.
Not that lame bacon we are more familiar with at your conventional breakfast diner.
In addition, the place was clean and the staff was very helpful. The butcher had
several different cuts available and was knowledgable as well as friendly. I left
with some cuts of pork and beef and am excited to come back!, stars: 5.0 --
```

Figure 2: yelp_review-stars_test_American_6.txt

The second sets were trained using corpora of reviews followed by stars (Figure 2). These models were used to evaluate how effectively the models learned the relationship of the generated text to the star review. In these corpora, the training and test text were wrapped in meta information consisting of the text "review: ", ", stars: ", and terminated by a "`-- <CR>`". The use of this wrapping allowed a rapid evaluation of the level of training of the model (i.e. did it learn the wrapping pattern effectively), and once learned, the meta-wrapping supported easy extraction of the synthetic data using regular expressions.

The third set was trained using a masked corpora that did not include the phrase "vegetarian options" to compare against the other model and ground truth.

4 Results

In this section, we describe how the GPT is able to incorporate memorization, interpolation, and extrapolation into its behavior. We find that each one of these contexts provides useful mechanisms for determining the performance of such models.

4.1 Memorization

In this section, we focus on the ability of the GPT to memorize repeating patterns while also reproducing statistically similar data with respect to ground truth. To do this, we generated *meta-wrappers* from the ground truth. In this case, the number of stars, useful votes, funny votes and cool votes contained in the Yelp data. Examples of this are shown in Figure 1. When given an insufficiently large corpora, the model would fail to learn the pattern correctly

model (lines)	error %	correlation %
6k	0.24%	0.36%
12k	0.22%	0.62%
25k	0.14%	0.86%
50k	0.00%	0.96%
100k	0.00%	0.99%
200k	0.00%	0.98%

Table 1: Memorization Error & Correlation

resulting in generated strings like:

```
stars_votes =
0stars_stars_stars_min = 2.0,
useful_votes = 0,
```

However, once the corpus contained more then 50,000 lines, the model learned the pattern perfectly, and there were no more errors (Column 'error %' in Table 1).

We also tested the effects of corpus size on the ability of the model to reproduce the statistical properties of the ground truth numeric data[5]. We found that increasing the number of lines improved the learning of the statistical information by the models using Pearson's correlation. However, as can be seen in the 'correlation %' column of Table 1, it appears that the best training occurs at 50k-100k lines, with the 200k line model overfitting and no longer generalizing (Dietterich, 1995).

These results indicate that the TLMs can both memorize the structure of data and reproduce arbitrary amounts of information using that structure that are substantially similar to ground truth. These memorization properties allow us to evaluate the quality of models by injecting known ground truth into the data using meta-wrapping and evaluating the statistical properties of the results.

4.2 Interpolation

In this section, we explore how finetuned GPT models are able to generate data that appropriately represents the behaviors of the group that provided the corpora. In this case, we trained models on 50k and 100k review corpora using the "American" cuisine. The arrangement of the corpus used for training is shown in Figure 2.

We then trained a model using 50k corpora and 6 epochs to compare to ground truth data. We then generated 10,000 reviews using the prompt "review:" and parsed and stored the results. Any review that ran too long to generate a star value

[5]The vote data is mostly zeros and not as useful as the star information

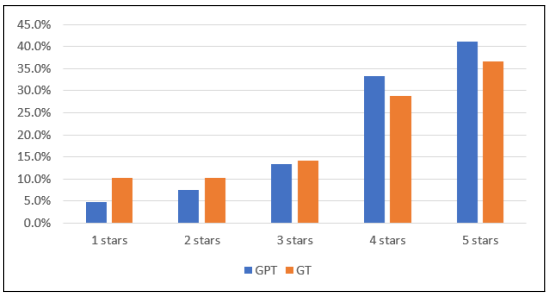

Figure 3: American Ground Truth and GPT star distribution

Avg star rating	GPT	GT	% difference
NEGATIVE	2.56	2.29	5.45%
POSITIVE	4.45	4.44	0.25%

Table 2: Star ratings for Sentiment

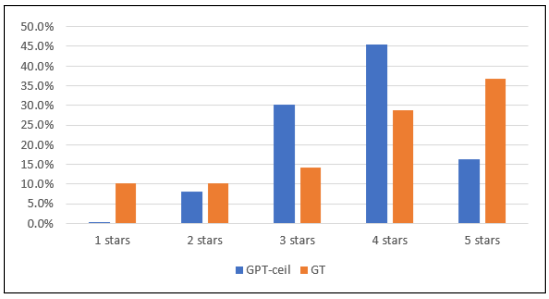

Figure 4: Pre-trained GPT vs Ground Truth

Avg star rating	GPT-pre	GT	% difference
NEGATIVE	3.25	2.29	29.46%
POSITIVE	4.04	4.44	9.9%

Table 3: Star ratings for Sentiment (pretrained GPT-2)

was rejected resulting in a total of 9,228 usable review/star pairs. This model accurately reflected the distribution of stars in the ground truth with a Pearson's correlation of 99.6% (Figure 3).

An extract from a generated 4-star review is shown below:

> *"Service is good, staff is very friendly and helpful. Prices are reasonable and the restaurant is clean. The food was great. I had the veggie burger, which was great."*

To determine sentiment for reviews like this, we used the Flair sentiment analysis API (Akbik et al., 2019) for each review and stored the results (6,926 positive, 2,302 negative). We also did this for 10,000 Yelp reviews selected from the "American" cuisine (6,624 positive, 3,376 negative). We then calculated the average number of stars for a POSITIVE review and a NEGATIVE review for the generated and ground truth data. The results of this comparison are shown in Table 2.

The generated results are nearly identical with the ground truth, and show how well the GPT is able to generate internally consistent reviews and stars.

To see how different this was from the pretrained model, we used the prompt "What follows is a typical example of a restaurant review of an American-style taken from Yelp's database:". This was more complex in that there was no meta wrapped output, so more complicated parsing had to be done. For instance, the GPT would sometimes rate on a 10-point rating and these scores had to be converted to the 5-point scale. Figure 4 shows a bias towards positive (4-star) reviews that is inherent in the pretrained model, while the ground truth is biased towards 5 stars. The correlation here is nowhere near the 99.6% of the finetuned model, though it is still significant at 47.86%. The match of sentiment to stars is also still apparent in this data (Table 3) even though it is less pronounced than in the finetuned GPT output. This may be partially accounted for by the ways that ratings had to be parsed and combined.

This bias towards positive reviews in the pretrained model may have led to some interesting behavior on the part of the finetuned models when we tried to elicit negative (e.g. 1-star, 2-star, etc.) reviews. Although it was possible to produce bad reviews given a sufficiently negative prompt, the effort required to produce a one-star review was perplexing.

Figure 5 shows a prompt "No vegetarian options" that produced substantially negative reviews in the original reviews but produces generally positive reviews when submitted to the GPT trained on American reviews (Pearson's correlation of -63%).

Figure 6 shows a similar behavior for a generally positive prompt, "Many vegetarian options". In the ground truth, there are more 5-star reviews than any other, while in synthetic reviews, the peak is again at 4 stars. This is roughly the same pattern that appears in the pretrained GPT (Figure 4) and the negative review (Figure 5).

To create an overwhelmingly one-star review with this model required the prompt "Everything about this place is terrible. The food is crap. The

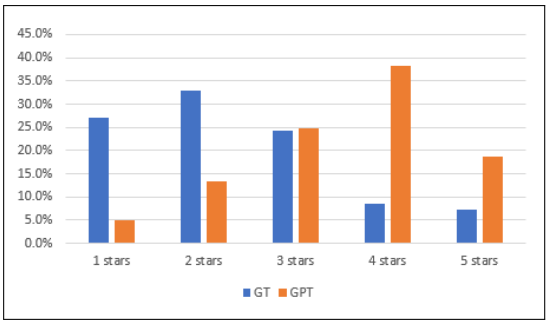

Figure 5: “No vegetarian options” Unbalanced

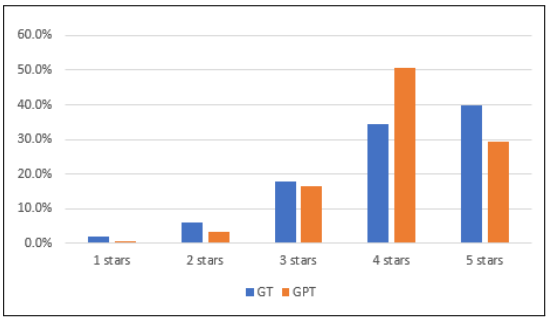

Figure 6: “Many vegetarian options” Unbalanced

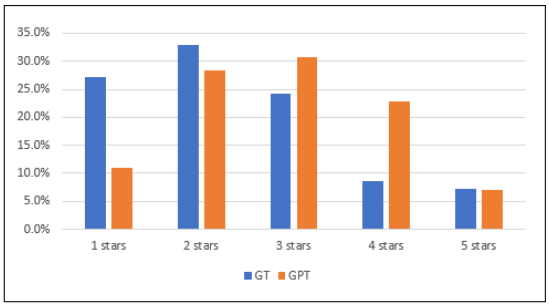

Figure 7: “No vegetarian options” Balanced

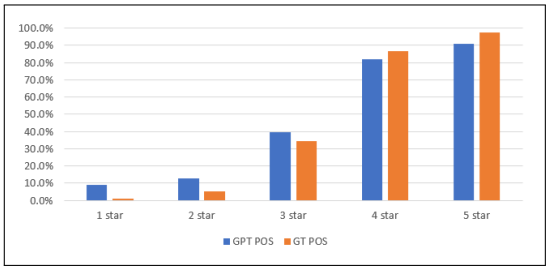

Figure 8: GPT/GT Isolated Star Positive

staff is terrible”. Clearly the model is capable of producing one-star reviews, but requires more extensive prompt tuning to do so.

It appears that although there are many pathways to produce 3, 4, and 5 star reviews, there is a smaller “prompt space” that produce a sequence of tokens that produce negative reviews. Remarkably, even when the model is trained on a corpus that is *balanced with respect to stars*, it still produces substantially more positive reviews for the “No vegetarian options” prompt (Figure 7) and less 5 star reviews than the ground truth for the positive prompt “Many vegetarian options”.

To generate the appropriate sentiment/star behavior, we had to train 5 models, one for each star rating for reviews with the “American” category. Each model was trained with a 50k review corpora created from the ground truth database as shown in Figure 2. Each model was prompted with the no/some/several/many vegetarian options described above.

The ratio of positive to negative sentiment for each model was compared to the sentiment ratio of 1, 2, 3, 4, and 5 star reviews in the ground truth data. As we can see in Figure 8 and Figure 9, these correlations are much stronger (Pearson’s correlation of 99.97%) than any of the previous approaches.

We believe that the reason that this works is because each star review represents a distinct linguistic population. On one end of the spectrum are the disgruntled, often using language that focuses on poor service such as in this extract:

> *“We were basically seated at a table by the host, then told quite rudely by the server that we couldn’t sit there. Then we proceeded to watch as the host and server fought over whether we could sit there or not.”*

At the other end of the spectrum is the 5-star group who have had a perfect meal with great service. These reviews are overwhelmingly classified as positive. We can show this relationship of these emotional terms to stars from a different perspective by using the Linguistic Inquiry and Word Count (LIWC) Dictionary (Pennebaker et al., 2001), which calculates the representation percentages of certain sets of words. One set of terms in the LIWC has to do with affect, ranging from positive (e.g. happy, pretty, good) to negative (e.g. hate, worthless, enemy). We can see in Table 4 how dissimilar the one and five star groups are:

These clusterings and patterns of usage allow the GPT to effectively learn the linguistic behaviors of the population so that it can accurately generate novel text that has the same sentiment patterns. And as we will see in the next section, these models

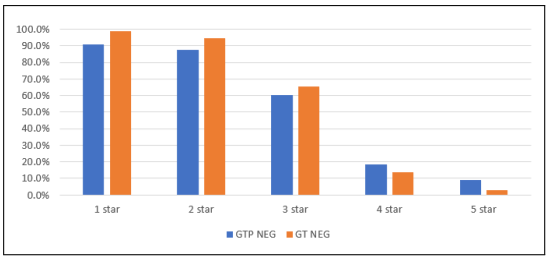

Figure 9: GPT/GT Isolated Star Negative

Affect	Pos Emo	Neg Emo
GPT 1 star	2.869%	1.461%
GT 1 star	2.710%	1.936%
GPT 5 star	7.241%	0.358%
GT 5 star	8.277%	0.572%

Table 4: LIWC Affect Terms for GT and GPT reviews

POSITIVE	no	some	several	many
1 star	**0**	**0**	**0**	**0**
2 star	**0**	1	**0**	**0**
3 star	4	8	7	21
4 star	6	31	29	90
5 star	6	29	27	100

Table 5: Vegetarian ground truth positive review counts

NEGATIVE	no	some	several	many
1 star	21	1	1	6
2 star	24	6	8	18
3 star	13	6	7	31
4 star	1	2	2	8
5 star	**0**	2	**0**	3

Table 6: Vegetarian ground truth negative review counts

are able to accurately *extrapolate* text in response to prompts that do not appear in the training data, a critical element if we are to be able to use these models for polling and survey purposes.

4.3 Extrapolation

In our ground truth Yelp dataset, some queries result in very few reviews. When looking at only reviews with the keywords "some vegetarian options" or "no vegetarian options" there are only a handful or in the most extreme cases **no** related reviews. We can see this in the sample from the Yelp data in Tables 5 and 6.

This problem often occurs with datasets where questions may not have been asked, conditions have changed (such as the rapidly evolving information space surrounding COVID-19) or where the structure of the data makes certain responses unlikely. This makes obtaining information about these cases difficult or impossible with traditional methods.

Extrapolation can address this problem by letting the model extrapolate from "adjacent" information to generate relevant, zero-shot data as we saw in the Floober example in the introduction.

To demonstrate this, we trained a new set of isolated star models on a 50k corpora that had all reviews containing the phrase "vegetarian options" *removed*, or masked. These models then generated *extrapolated* responses to the "no/some/several/many" prompts.

We then compared the behavior of the *interpolating* model that had been trained on corpora "vegetarian options" reviews, and a baseline of statistical samples taken from the known ground truth of 97 samples of all three-star reviews in our set of "no/some/several/many" samples. We chose baseline sample sizes of 8, 18, and 24 because those were the average size of the number of negative, positive, and combined reviews in our samples. Each sample (baseline and GPT) was randomly sampled 1,000 times and averaged for subsequent calculations. Because the GPT is able to produce unlimited reviews, we were able to use a sample size of 1,000 for these synthetic reviews.

We derived the l2 distance from POS/NEG percentage calculated from the Known Ground Truth (40.25% / 59.74%) for the GPT and baseline versions, which is shown in Table 7. We can clearly see that the baseline(8) has the highest l2 error (20.01%), while the GPT trained on the unmasked corpus has the lowest. Remarkably, the masked, *extrapolating* GPT model has the second-lowest error, and has less than half the error of the baseline(26) evaluation.

This is important because it demonstrates that the GPT (no veg) model is able to generate text related to vegetarian options *despite being trained on data with no reviews related to vegetarian options*. These results are substantially better than the baseline even when the baseline includes over 25% of the existing vegetarian samples. The model's ability to generate matching sentiment reviews is based purely on extrapolating between the rest of the reviews it was trained on.

These results mean that we can use language models such as the GPT to effectively learn the linguistic behaviors of the population and generate

	Pos %	Neg %	Error l2
Ground Truth	40.25%	59.74%	0.00%
GPT	40.71%	59.28%	1.89%
GPT (no veg)	37.58%	62.41%	3.88%
baseline(26)	39.38%	60.12%	9.16%
baseline(18)	40.55%	59.44%	11.78%
baseline(8)	39.87%	60.12%	20.01%

Table 7: Ground Truth vs. Extrapolation vs. Baseline

accurate responses to questions that have never been asked of the original group but are *latent* in the weights of the model. This technique creates a powerful new capability for polling and survey purposes.

5 Discussion

Polling transformer language models has provided us with a new lens to assess public attitude/opinions well beyond dining options. The same technique can be used on to determine social, political and public health issues using corpora from a variety of sources. It is dynamic and can be used to answer questions using latent information. Further, is computationally inexpensive and does not require any costly human annotated ground truth to train.

The strength of the GPT is also a weakness. Because it stochastically generates each new token based on the ones that preceded it, but also on randomness-introducing parameters such as temperature, it can be difficult to make it behave in ways that are both predictable and dynamic. A temperature of zero will produce the same result repeatedly, but then the distribution of responses to the prompt will be lost. The best way to use these models may be to focus on the statistics of large-scale patterns rather than looking at individual responses. Stochasticity ensures that some percentage of texts will untrustworthy, but at scale such outliers can be identified and handled appropriately.

In addition, prompt design is tricky. Small changes in prompts may result significant changes in results (e.g., “some vegetarian options” versus “many vegetarian options”). Limitations of the TLMs themselves may also prevent them from providing accurate information. For example, although humans can link affordances (*I can walk inside my house*) and properties to recover information that is often left unsaid (*the house is larger than me*), TLMs struggle on such tasks (Forbes et al., 2019).

TLMs are also vulnerable to *negated* and *misprimed* probes. Simply adding “not” to a probe (e.g. “The theory of relativity was *not* developed by”, often generates “Albert Einstein”. Mispriming, or the addition of unrelated content to the prompt (e.g. “Dinosaurs? Munich is located in” the probe can produce highly distorted results. (Kassner and Schütze, 2019)

In this paper, we have shown that TLMs such as the GPT can be used as an effective data collection technique to gain a deeper understanding of sample populations. We believe these techniques can also be used to explore social, political and health issues, but it is important to understand their limitations.

6 Conclusions

In this paper, we described a new method for polling online data sources that uses broad keywords (e.g. cuisine = "American", stars = "3") to extract a corpora that is used to train a TLM such as the GPT. The finetuned model captures sociolinguistic patterns of the group polled that can then be accurately queried using highly targeted prompts such as "no vegetarian options".

This unique method of querying a population on content that may not exist explicitly in the ground truth can be achieved due to TLMs capacity for memorization (learning repeating patterns), interpolation (creating variations on existing values), and extrapolation (inferring new content from existing).

We demonstrated that using TLMs in this way is actually more reliable/accurate than using ground truth queries that produce sparse results, even if the TLM model is not trained on the specific topics of interest. This opens up a tremendous opportunity for textual research where relevant data is missing, in small quantity, or volatile.

7 Future Work

So far, we have only scratched the surface trying to probe and understand the latent knowledge captured in a transformer language model. Our next work will involve using this technique to poll latent information on Twitter regarding public health issues. This will involve training our models on left-wing, right-wing and other groups participating in the ongoing COVID-19 online discussions. We will also be exploring the effects of negation, mispriming, and other techniques that may distort the latent knowledge captured by these models.

8 Ethical Considerations

Large Transformer Language Models' capacity to rapidly generate unethical or dangerous content (e.g. realistic mass-shooter manifestos) is well understood. Beyond the risk of the generation of credible fake content, there are additional risks for social research using TLMs.

The methods by which the latent information is stored in the model weights is a form of dimension reduction that cannot incorporate all of the nuance in the data it has trained on as it learns linguistic patterns in the data. As a result, it will inevitably fail to capture outlier behaviors in the model weights.

Even for patterns which are largely correct, the models are capable of making informational errors, such as the improper attribution demonstrated in our Floober example in Section 1. The model followed the highly credible linguistic pattern of an academic or Wikipedia description of an animal, complete with a likely animal family, and attributed the discovery to a person, but it was the *wrong* person.

This class of error makes the latent information in TLMs valuable for population scale questions, but potentially dangerous for attributable content. The results of the model are that of generalized linguistic behaviors, and not attributable to a specific individual. Prompt tuning the model with quotes from a particular individual might provide salacious or unethical content which not only has never been produced by the individual, but includes ideas they may abhor. In fact, persistent biases or stereotypical behaviors often exist within the model's weights. (Abid et al., 2021) (Nadeem et al., 2020)

As a result, it would be extremely dangerous to utilize this sort of latent information to perform predictive actions on individuals as a result of the output of these models. AI is increasingly being applied to predictive tools for law enforcement, employment screening, and other systems that judge individuals based on an algorithmic assessment (Broadhurst et al., 2019) (Ponce et al., 2021). Attempting to leverage the techniques we've demonstrated for a system of that nature would be potentially misleading, possibly dangerous, and certainly unethical.

References

Abubakar Abid, Maheen Farooqi, and James Zou. 2021. Persistent anti-muslim bias in large language models. *arXiv preprint arXiv:2101.05783*.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Roderic Broadhurst, Donald Maxim, Paige Brown, Harshit Trivedi, and Joy Wang. 2019. Artificial intelligence and crime. *Available at SSRN 3407779*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Elanor Colleoni, Alessandro Rozza, and Adam Arvidsson. 2014. Echo chamber or public sphere? predicting political orientation and measuring political homophily in twitter using big data. *Journal of communication*, 64(2):317–332.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tom Dietterich. 1995. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27(3):326–327.

Maxwell Forbes, Ari Holtzman, and Yejin Choi. 2019. Do neural language representations learn physical commonsense? *arXiv preprint arXiv:1908.02899*.

Floyd J Fowler Jr. 2013. *Survey research methods*. Sage publications.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Nora Kassner and Hinrich Schütze. 2019. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. *arXiv preprint arXiv:1911.03343*.

Kris McGuffie and Alex Newhouse. 2020. The radicalization risks of gpt-3 and advanced neural language models. *arXiv preprint arXiv:2009.06807*.

Miriam Meyerhoff. 2018. *Introducing sociolinguistics*. Routledge.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2020. Stereoset: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456*.

Shriphani Palakodety, Ashiqur R KhudaBukhsh, and Jaime G Carbonell. 2020. Mining insights from large-scale corpora using fine-tuned language models. In *ECAI 2020*, pages 1890–1897. IOS Press.

James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Aida Ponce et al. 2021. The ai regulation: entering an ai regulatory winter? why an ad hoc directive on ai in employment is required. *Why an ad hoc directive on AI in employment is required (June 25, 2021). ETUI Research Paper-Policy Brief*.

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

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Christopher Rytting and David Wingate. 2021. Leveraging the inductive bias of large language models for abstract textual reasoning. *Advances in Neural Information Processing Systems*, 34.

Luke Sloan, Jeffrey Morgan, Pete Burnap, and Matthew Williams. 2015. Who tweets? deriving the demographic characteristics of age, occupation and social class from twitter user meta-data. *PloS one*, 10(3):e0115545.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.