Measuring Consensus with Linguistic Self-similarity

Jeremiah Milbauer
Carnegie Mellon University
jmilbaue@cs.cmu.edu

Abstract

As discussions of polarized echo chambers continue to feature in headlines, using unsupervised techniques to identify the degree of consensus in online communities is rapidly becoming an important task in applied natural language processing (NLP). Recent progress with large pretrained language models has show that they can be used to mine for world-knowledge and opinions through the use of cloze tasks which ostensibly rely on world or opinion knowledge (for instance, `"a cat has [MASK] legs"`); additionally, they can be effectively fine-tuned to narrow domains. It stands to reason that the knowledge and opinions which can be correctly completed by the language model will reflect the distribution of facts and opinions observed by the model during training. In this work, we show how a suite of fine-tuned language models models can both capture differences in opinion across communities, and be used to compare the degree of consensus across those communities. This work opens the way for the use of language modeling as an essential tool for understanding and identifying echo chambers.

1 Introduction

In recent years, online social media has taken on an increasingly large role in shaping real-world political discourse. Social media platforms can be the birthplace of political movements, a hot-bed for political conspiracy theories, and even tools for powerful politicians to communicate with their constituents. For instance, in 2016, Reddit.com was the first place the "Pizzagate" conspiracy theory, which falsely claims that many high-ranking Democratic politicians were involved in a child sex trafficking ring operating out of a pizza restaurant. And Twitter served as the nexus of the "#MeToo movement," a massive social movement to expose sexual harassment, violence, and abuse by people in power.

The increased visibility and increased power of online social media does not come without scrutiny, however. Many argue that online social media promotes the formation of "echo chambers," communities where a single viewpoint, lacking in nuance, dominates – and all other views may be ridiculed, ignored, or deleted.

As we continue to rely on online communication to augment and (especially since the Covid-19 Pandemic began) replace in person communication, it is becoming essential to understand the dynamics of online consensus and echo chamber formation. And as the amount of social data continues to grow, and the number of online communities continues to multiply, computational techniques which allow echo chambers and consensus to be measured automatically and at large scale are essential.

1.1 Related Work

Network-based analysis.
Existing approaches to the measurement of echo chambers have often looked to social network measurements. [Morales *et al.*, 2015] [Garimella and Weber, 2017]. However, network-based approaches have many drawbacks. For one, they are not user-agnostic, as it is the behavior of the users themselves, not the content they post, that determines polarization and defines the echo chamber. Furthermore, they often assume a space where users can all interact with all communities, and where these interactions can be traced. An approach that instead relies on language understanding will be more flexible.

Language modeling.
Though traditionally a tool for evaluating and generating natural language, the recent success of transfer learning with large pretrained language models (LMs) [Vaswani *et al.*, 2017] [Devlin *et al.*, 2019] has seen the representations they learn become an essential component of many natural language processing tasks. At the same time, recent work has also shown that the language models *per se* are multitask learners, capable of use in natural language understanding by inventing ways of framing traditional language understanding problems as an appropriate input to an LM. [Radford *et al.*, 2019] This approach can render language models useful for tasks including summarization, machine translation, question answering, knowledge graph construction, and many other tasks.

Knowledge and opinion mining.
Large language models have been demonstrated to act as knowledge bases, and are capable of learning information

contained within the text they are trained on. [Petroni *et al.*, 2019]. (Recently, [Carlini *et al.*, 2020] demonstrated that the training data itself can be extracted from the language model)

Additional work has also shown that language models can be used for opinion mining [Palakodety *et al.*, 2020], and that these method is effective in political domains.

1.2 Contributions

Multi-community opinion mining. In this work, we show that fine-tuned language models can learn community-specific opinions.

Measuring consensus. We then introduce a framework for using language models to evaluate and compare degrees of consensus across multiple communities, relying on a certainty-sampling strategy to select polarized comments.

2 Experimental Framework

When communities exhibit a high degree of consensus, the opinions present in that community are predictable. A consensus is essentially knowledge of a community that can inform us of the probability of certain opinions occurring within some community. When people seek to align themselves with a community, they are essentially evaluating whether their opinions match the consensus. But knowledge of a consensus is conditioned on previously observed opinions within a community. Effectively, consensus manifests as a scenario in which:

$$P(\text{opinion}_n|\text{opinion}_1, \ldots, \text{opinion}_{n-1})$$

closely matches the actual distribution of opinions within a community.

Thus, we hypothesize that in domains with a high degree of consensus, fine-tuned language models will predict unseen text with high accuracy. As a corollary, we hypothesize that in a multi-domain (or perhaps more accurately, a single domain consisting of sub-domains) setting, this approach will reveal the comparative degree of consensus within each sub-domain.

In essence, we anticipate that subdomain-specific fine-tuning will bias the model to the subdomain more when that subdomain contains a higher degree of consensus.

3 Data

In order to determine whether or not language models perform better in communities with consensus, we aimed to use a corpus that: (1) contained data from multiple communities; (2) existed in a domain with a notable consensus effect; and (3) could be immediately useful for understanding political "echo chambers."

3.1 Dataset

Our analysis in this work is based on a corpus of YouTube comments collected from videos posted to four channels: CNN, Fox News, MSNBC, and One America News Network (OAN). Analyses of partisanship in the U.S. media landscape have typically categorized CNN as center-left, Fox News right, MSNBC left, and OAN far-right [Faris *et al.*, 2020].

	Lines	Tokens	Types
CNN	19.29M	480.67M	2.57M
FOX	20.16M	473.10M	2.22M
MSNBC	10.24M	274.06M	1.56M
OAN	1.30M	32.02M	.033M

Table 1: Corpus statistics

Unlike a standard language modeling corpus, these comments are mostly extremely short, often informally written, contain a high proportion of slang words and emoji characters, and contain many repeated popular slogans. Because of the choice of YouTube channels, the comments are mainly focused on discussion of political topics. Statistics about the dataset and channels are included in Table 1.

Given the known biases of these four sources, as well as their respective agenda-setting preferences, we initially hypothesized that CNN would have the least consensus, then MSNBC, then Fox News, and finally OAN would demonstrate the most consensus.

4 Methods and Results

Our approach consisted of fine-tuning a language model for each of the four YouTube channels (as well as a mixture of the four), and then evaluating the performance of each language model on data from the corresponding channel, the non-corresponding channels, and a subsample of sentences that were highly characteristic of each channel.

4.1 Language Modeling

We initially partition the data by source YouTube channel, resulting in four sub-corpora of uneven size. We also create an artificial fifth "source", consisting of 2M comments drawn equally from each of the four channels' comments. Next, we reduce the size of each sub-corpus so that they all contain the same number of comments by sampling approximately 1.2M comments from each source. We will refer to these equal sized sub-corpora as $\mathcal{C}_{cnn}$, $\mathcal{C}_{fox}$, $\mathcal{C}_{msnbc}$, $\mathcal{C}_{fox}$, and $\mathcal{C}_{all}$.

Each of these 1.2M comment sets were further partitioned: a 1M comment training set $\mathcal{C}^{train}$, a 100K comment validation set $\mathcal{C}^{val}$, and a 100K comment evaluation set $\mathcal{C}^{eval}$.

We then fine-tuned five `DistilBERT` [Sanh *et al.*, 2019] models, each initialized from the same pretraining checkpoint.[1] Although a more powerful model, such as `BERT` [Devlin *et al.*, 2019] may have been preferable, `DistilBERT` was chosen both because a lower-parameter model might become more quickly biased to the fine-tuning data, and to reduce computational costs. Each model was fine-tuned for 3 epochs on the appropriate 1M comment training set from either $\mathcal{C}_{all}$, $\mathcal{C}_{cnn}$, $\mathcal{C}_{msnbc}$, $\mathcal{C}_{fox}$, or $\mathcal{C}_{oan}$, producing five fine-tuned models: $\texttt{DistilBERT}_{all}$, $\texttt{DistilBERT}_{cnn}$, $\texttt{DistilBERT}_{msnbc}$, $\texttt{DistilBERT}_{fox}$, and $\texttt{DistilBERT}_{oan}$.

[1] This paper uses a standard `DistilBert` initialization and checkpoint, via `distilbert-base-uncased` provided as part of the Hugging Face `transformers` library. [Vaswani *et al.*, 2017]

	CNN	Fox	MSNBC	OAN	All
I think Trump is [MASK]	right	right	right	right	right
	stupid	great	stupid	correct	crazy
	lying	good	insane	winning	stupid
I think Hillary is [MASK]	right	right	right	right	right
	lying	crazy	great	dead	corrupt
	good	nuts	winning	evil	crazy
I think Bernie is [MASK]	right	right	right	right	right
	winning	crazy	great	crazy	crazy
	good	nuts	awesome	great	wrong
Our biggest enemy is [MASK]	trump	china	russia	china	china
	america	russia	america	islam	russia
	russia	trump	china	america	isis
[MASK] lives matter	black	all	black	all	all
	all	black	all	black	black
	white	blue	american	blue	blue

Table 2: Evaluation of cloze prompts by fine-tuned language models, with the top three candidates shown

	Hit@1	Hit@3	Hit@10	MR	MR*
CNN	.5343	.6883	.7965	49.06	4.38
FOX	.5812	.7343	.8329	36.88	2.46
MSNBC	.5840	.7322	.8279	40.05	2.70
OAN	.5725	.7264	.8238	41.50	2.90

Table 3: $\texttt{DistilBERT}_{all}$ performance on each evaluation set

	Hit@1	Hit@3	Hit@10	MR	MR*
CNN	.5306	.6859	.7960	47.90	4.07
FOX	.5733	.7272	.8275	38.04	2.66
MSNBC	.5883	.7364	.8323	38.19	2.49
OAN	.5984	.7489	.8412	36.45	2.14

Table 4: Fine-tuned language model performance on matching evaluation set

To evaluate the ability of this fine-tuning approach to reflect the broad opinions of a community, we hand inspected a few politically-charged cloze prompts. A selection of those prompts, and the suggestions made by each model, are given in Table 2.

4.2 Random Sampling

First, we evaluate the performance the language models to predict unseen text from the same corpus on which it was trained. We use 100,000 sentences from each community-specific $\mathcal{C}^{eval}$. Within each sentence, we randomly mask one token with "[MASK]," and use the language model to predict the masked token. Table 3 contains the results of $\texttt{DistlBERT}_{all}$ on each of $\mathcal{C}^{eval}_{cnn}$, $\mathcal{C}^{eval}_{msnbc}$, $\mathcal{C}^{eval}_{fox}$, and $\mathcal{C}^{eval}_{oan}$. Table 4 contains the results of $\texttt{DistlBERT}_{cnn}$, $\texttt{DistlBERT}_{fox}$, $\texttt{DistlBERT}_{msnbc}$, and $\texttt{DistlBERT}_{oan}$ on the matching $\mathcal{C}^{eval}$.

In each table, H@N represents how often the correct token was in the top N predictions of the language model. MR is the mean rank of the correct token (ranks were capped at 1000; if the rank of the correct token was 1001, it was instead treated as 1000). MR* is the mean rank of the correct token, if the lowest 10% and highest 10% of ranks are ignored.

Discussion

The results of the evaluation experiment using $\texttt{DistilBERT}_{all}$ reveals that there is an inherent difference in the predictability of the text across the four channels. The text from $\mathcal{C}^{eval}_{cnn}$ is predicted with a significantly lower accuracy than the other texts. Because the comment section of CNN is less predictable, while the other three comment sections are, it is likely that the reason for CNN's lack of predictability has to do with patterns not contained within the training data rather than a political difference with CNN. This may be a result of the agenda-setting decisions of the different news channels. MSNBC, Fox, and OAN are focused almost exclusively on political content; as a network with a broader focus, CNN may contain more out-of-distributions topics. Another possible explanation is that as the most "mainstream" source, commenters of all types are drawn to CNN. However, an investigation of the four channels' agenda-setting and commenter demographics is beyond the scope of this work.

When evaluating each fine-tuned DistilBERT model against the corresponding $\mathcal{C}^{eval}$, we would expect to see general improvements to the prediction scores. This is largely supported by the data; however $\texttt{DistilBERT}_{fox}$ performs worse than $\texttt{DistilBERT}_{all}$ on $\mathcal{C}^{eval}_{fox}$.

If we consider the performance of each fined-tuned language model on their corresponding data, we can measure the degree of self-consistency, and thereby consensus, in each community. These results indicate that CNN has the least consensus, Fox the second least, MSNBC the second most, and OAN the most consensus. This is largely consistent with the our initial guess. CNN, with the broadest coverage and the closest to a balanced viewpoint and audience, has the lowest consensus. OAN, with a very specific far-right focus, view-

point, and audience, has the greatest degree of consensus.

4.3 Certainty Filtering

However, we observed that although the comments are sourced from political YouTube channels, not all comments are political in nature. Furthermore, many comments on a given channel actually run counter to the apparent political opinion of that channel. This can be explained by the fact that some users will purposely cross community boundaries to interact in a way that violates the consensus of other communities.

To avoid sampling these non-characteristic, non-political comments, we introduce a method of Certainty Filtering. By training a multiclass classifier to predict the channel of origin for sentences in our dataset, we can then sample sentences from each evaluation set which are characteristic of the appropriate channel by selecting those with high probability of the correct class.

We built a simple simple bag-of-words classifier. First, we selected the 10,000 most frequent words in the corpus as our vocabulary, ignoring NLTK stopwords. Then, each of the 1,000,000 comments in each $\mathcal{C}^{train}$ was converted into a bag of words, represented as a vector in $\mathcal{R}^{10000}$. We used a simple feedforward neural network implemented in `PyTorch` [Paszke *et al.*, 2017] with a single hidden layer (using ReLU activation and .5 dropout [Srivastava *et al.*, 2014]) and an output dimension of 4. Softmax was used to obtain probabilities. The classifier was optimized using Adam [Kingma and Ba, 2015], with a negative log likelihood loss, and achieved an overall accuracy of .6289 after 10 epochs.

Using the classifier, we then predicted the labels for each $\mathcal{C}^{eval}$. Figure 1 gives an example of the classifier's performance for comments in $\mathcal{C}^{eval+val}_{cnn}$

Within each $\mathcal{C}^{eval}$, we ranked the comments by the predicted probability of the correct label. We then selected the top 10% of comments from each ranking, and evaluated each fine-tuned `DistilBERT` model on these comments, as before. The results of $\texttt{DistilBERT}_{all}$ on each filtered $\mathcal{C}^{eval}$ are shown in Table 5. The results of each fine-tuned $\texttt{DistilBERT}_n$ model on each $\mathcal{C}^{eval}_n$ are shown in Table 6.

Discussion

Compared to its performance on the unfiltered comments, the $\texttt{DistilBERT}_{all}$ generally performs worse on the filtered comments. This indicates that these comments are, as anticipated, more polarized and less likely to fall within the overall distribution of the YouTube data.

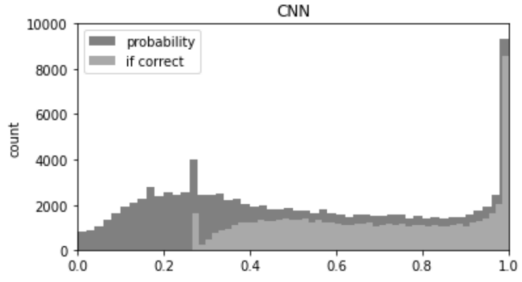

Figure 1: Distribution of Classifier's P(CNN) for CNN-origin comments

	Hit@1	Hit@3	Hit@10	MR	MR*
CNN	.5408	.6752	.7806	68.47	6.73
FOX	.5645	.7210	.8185	47.43	3.321
MSNBC	.5537	.6985	.7963	58.95	5.13
OAN	.5908	.7547	.8443	24.83	2.03

Table 5: $\texttt{DistilBERT}_{all}$ performance on each evaluation set, after certainty filtering

	Hit@1	Hit@3	Hit@10	MR	MR*
CNN	.5510	.6905	.7843	69.12	7.37
FOX	.5649	.7111	.8048	52.17	3.98
MSNBC	.5532	.6982	.7913	61.46	5.44
OAN	.6249	.7489	.8412	24.99	1.24

Table 6: Fine-tuned language model performance on matching evaluation set, after certainty filtering

The consensus effect of the fine-tuned models is also more pronounced on the filtered data. Comparatively, these results indicate that CNN has the least consensus, and MSNBC the second least. It indicates that OAN has the most consensus – by a wide margin – and Fox has the second most.

5 Conclusion

In this work, we have demonstrated a simple approach to the use of language modeling for consensus measurement. This approach differs significantly from previous approaches to measuring consensus, as it relies on the content, semantics, and values of communities – rather than observations of their network properties.

The measurements of consensus provided by this approach are largely consistent with our expectations about the communities built around four notable U.S. news networks. We attribute this to both the agenda-setting choices of the networks, as well as the consistency of the commenters' political opinions.

Text-based analysis of consensus is a fruitful new direction of research. It has many advantages over network-based measurement of "echo chambers": it allows for the direct mining of community beliefs, it allows for comparisons of communities where users are anonymous or do not participate in a broader social network, and it could even be extended to an online algorithm for tracking consensus.

6 Challenges and Self-criticism

Doing this work was not without challenges. One of the challenges was managing the necessary computational resources for the project. Future versions of this project might use a language model with more parameters than `DistilBERT`, such as `BERT` or `BART` [Lewis *et al.*, 2020] (which would also allow for more complex prediction prompts). It would also be good to run experiments with the Certainty-filtering

classifier across a few hyperparamerizations, perform multiple samplings for the evaluation, or evaluate this approach with language models other than large pretrained LMs.

This project also had limitations. Working with the data from YouTube meant that there was not a known ground-truth for consensus, other than the opinions of political commentators and perhaps myself (though I admit to a liberal bias). Furthermore, a set of only 4 communities was somewhat limiting.

7 Further Work

One way to correct some of these problems would be to consider different strategies for constructing artificial "communities" for evaluation. One possibility would be to take two opposing communities from the YouTube dataset (say, CNN and OAN) and artificially construct corpora by sampling from each of the two in different proportions. Then, use the strategies developed in this paper with the expectation that model accuracy will correspond to the imbalance of the sampling strategy. Another similar approach might be to sample data from Twitter networks, either by taking comments from users in a dense region of the Twitter network, or a wide one (this could be accomplished by some kind of parameterized random walk over users; we would then expect the consensus measurement to correspond to the parameterization of the sampling strategy).

Another direction I would like to see this work go in is the use of language models to compare communities. If the model trained on Fox has a high accuracy when predicting unseen OAN sentences, perhaps that is an indication that Fox and OAN are similar.

Still another direction this work could go in would be an always-on online algorithm for consensus and community detection. The system would be seeded by trained language models, and as new sentences came in they would be assigned the label of the language model that best predicted their contents. However, if performance on some large set of sentence was low enough, those sentences would then be used to seed a new language model, and all the sentences would then be re-evaluated. In this way we could continuously build a suite of language models which define communities.

Acknowledgements

Thank you to Ashique, Tom, Mark, and Rupak for creating a great curriculum and an exciting course!

References

[Carlini *et al.*, 2020] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models, 2020.

[Devlin *et al.*, 2019] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[Faris *et al.*, 2020] Robert Faris, Justin Clark, Bruce Etling, Jonas Kaiser, Hal Roberts, Carolyn Schmitt, Casey Tilton, and Yochai Benkler. Partisanship, impeachment, and the democratic primaries: American political discourse, january - february 2020. *SSRN*, 3717670, 2020.

[Garimella and Weber, 2017] Venkata Rama Kiran Garimella and Ingmar Weber. A long-term analysis of polarization on twitter. In *ICWSM*, 2017.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[Lewis *et al.*, 2020] M. Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, A. Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *ArXiv*, abs/1910.13461, 2020.

[Morales *et al.*, 2015] A. Morales, J. Borondo, J. C. Losada, and R. Benito. Measuring political polarization: Twitter shows the two sides of venezuela. *Chaos*, 25 3:033114, 2015.

[Palakodety *et al.*, 2020] Shriphani Palakodety, Ashiqur R. KhudaBukhsh, and J. Carbonell. Mining insights from large-scale corpora using fine-tuned language models. In *ECAI*, 2020.

[Paszke *et al.*, 2017] Adam Paszke, S. Gross, Soumith Chintala, G. Chanan, E. Yang, Zachary Devito, Zeming Lin, Alban Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

[Petroni *et al.*, 2019] F. Petroni, Tim Rocktäschel, Patrick Lewis, A. Bakhtin, Y. Wu, Alexander H. Miller, and S. Riedel. Language models as knowledge bases? *ArXiv*, abs/1909.01066, 2019.

[Radford *et al.*, 2019] A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[Sanh *et al.*, 2019] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey E. Hinton, A. Krizhevsky, Ilya Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929–1958, 2014.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, L. Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.