

Synthetically generated text for supervised text analysis

Andrew Halterman
Michigan State University
ahalterman0@gmail.com

March 15, 2024
(First draft: June 2022)

Abstract

Large language models are a powerful tool for conducting text analysis in political science, but using them to annotate text has several drawbacks, including high cost, limited reproducibility, and poor explainability. Traditional supervised text classifiers are fast and reproducible, but require expensive hand annotation, which is especially difficult for rare classes. This article proposes using LLMs to generate synthetic training data for training smaller, traditional supervised text models. Synthetic data can augment limited hand annotated data or be used on its own to train a classifier with good performance and greatly reduced cost. I provide a conceptual overview of text generation, guidance on when researchers should prefer different techniques for generating synthetic text, a discussion of ethics, a simple technique for improving the quality of synthetic text, and an illustration of its limitations. I demonstrate the usefulness of synthetic training through three applications: synthetic news articles describing police responses to communal violence in India for training an event detection system, a multilingual corpus of synthetic populist manifesto statements for training a sentence-level populism classifier, and generating synthetic tweets describing the fighting in Ukraine to improve a named entity system.

1 Introduction

Researchers in political science are rapidly adopting generative large language models (LLMs) to assist in the analysis of text. Generative large language models, such as GPT-3, Llama, and many others, show great promise in labeling or extracting information from documents (Ornstein, Blasingame, and Truscott 2022; Wu et al. 2023; Ziems et al. 2023), tasks that researchers have traditionally done by hand-labeling documents and training supervised learning models. A common approach to using LLMs for text analysis is to provide a prompt describing the task and available categories, along with the text to be analyzed. The LLM generates a response, including the category label it assigns to the document document or the other information the prompt elicits.

While using LLMs to label documents directly is simple to implement and performs well on some tasks (Gilardi, Alizadeh, and Kubli 2023; Ziems et al. 2023; Rytting et al. 2023), it faces several drawbacks. The most capable LLMs are closed source models behind commercial APIs. These models change rapidly and can be deprecated with little warning, making it difficult to reproduce previous results (Spirling 2023). This has implications beyond reproducibility: researchers who update text-derived datasets over time with new documents cannot apply a consistent coding scheme if the original LLM is not available. Second, LLMs are difficult to combine with hand labeled data without a separate and technically challenging fine-tuning step (Longpre et al. 2023). Third, while LLMs are clearly cheaper than hiring human annotators (Ornstein, Blasingame, and Truscott 2022), running them over a large corpus can be expensive in time, equipment, or API fees. Finally, the process by which the LLM provides a document label is inscrutable, particularly in the case of closed-source commercial LLMs, and even state-of-the-art LLMs fail on specific tasks (Peskoff and Stewart 2023).

The traditional approach to supervised text analysis—hand-labeling text and training a relatively small supervised classifier—avoids many of these issues. Traditional supervised

models are well understood, can often be interpreted, and can be iteratively improved through well understood processes of hyperparameter tuning and model selection. However, it depends on extensive efforts to collect labels. The annotation process is especially onerous if researchers are annotating a rare class, which requires that a large number of documents be annotated to obtain a sufficient number of positive examples.

This article suggests an alternative use of LLMs for supervised text analysis tasks: to generate synthetic text, which researchers then use as training data to fit a traditional supervised learning model. The synthetic text can augment a small set of hand-labeled data, be used on its own to train a classifier, or, in some cases, be hand annotated. Using synthetic training text addresses some of the problems with using LLMs to directly analyze text. Generating text, as opposed to analyzing it, is a much more straightforward task for LLMs, allowing researchers to use smaller local LLMs, greatly improving reproducibility and lowering costs. Researchers can publish their synthetic training data and models, allowing others to reproduce their results. Finally, researchers can easily validate the quality of the synthetic text and fit faster or more interpretable models than they can the output of an opaque LLM-based classifier.

To use synthetic text as training data, researchers must be able to guide or control its content. By *prompting*—changing the input to the LLM to guide its generation—or *adapting*—updating the model weights using a specific text corpus—researchers can guide the style and content of the synthetic text the LLM produces. After it is produced, the synthetic training text can *augment* an existing set of hand-labeled data to improve a classifier’s performance, or, in some cases, the synthetic training text can be used without hand labeling to train a document classifier, addressing the labeling issue. In the latter case, this use of synthetic text can be seen as a form of *model distillation*—using a larger, more capable model to train a smaller, narrowly focused model. This approach to using LLMs allows researchers to use them for their strengths—generating text—while avoiding

their weaknesses—opaque classification, high computational costs, a lack of reproducibility, and difficulty in combining with hand-labeled data.

The rest of the article describes how researchers in political science can use large language models to lower the costs of supervised text analysis by using a synthetic text approach. It provides guidance on the decisions that researchers face in generating and using synthetic text, when synthetic text can augment existing data and when it can be used on its own as a form of model distillation, introduces a technique for measuring and improving the quality of synthetic text, and discusses the ethical pitfalls inherent in using synthetic text. It presents three short applications from political science, illustrating how the approach can create synthetic news articles for training a model to classify police responses to violence in India, a multilingual sentence-level populism classifier with no hand-labeled examples, and synthetic tweets describing the war in Ukraine for a named entity recognition model. It shows that augmenting hand-labeled text with synthetic data improves performance and that high-quality synthetic on its own can be used to train a model without any hand labeling. It also shows that synthetic text can differ from real text in quantitatively measurable ways that a marginal synthetic example is usually less informative than a marginal real example. However, the near-zero cost of producing a marginal synthetic example offers major benefits for applied researchers developing text classifiers.

2 Using language models to generate synthetic text

To be useful as training data, synthetic text must have the content and style that researchers require. For example, to preview the first application, a researcher training a classifier to identify reports of Indian police making arrests requires a set of positive examples (stories containing arrests) and negative examples. To (partially) replace annotation with synthetic training text, a researcher needs a way to reliably produce documents with the desired context. A brief formalization of LLMs reveals the three options they have to

do so.

2.1 Formalizing text generation

Generative language models learn to produce text by optimizing a language modeling objective: conditional on a sequence of tokens (words), they predict which token is likely to follow. Formally, given a set of tokens $W = \{w_1 \dots w_n\}$, a language model assumes that the probability of the sequence can be decomposed into the probability of each token given the previous sequence of tokens: $p(W) = \prod_{i=1}^n p(w_i | w_{i-1}, \dots, w_2, w_1)$. We model the conditional probability of the next token given the previous tokens, trainable parameters θ , and a function f :

$$\hat{p}(w_i) = f(w_{i-1}, w_{i-2}, \dots, w_1, \theta). \quad (1)$$

To generate text from a language model, we then sample a token \hat{w}_i from the predicted distribution over the next word $\hat{p}(w_i)$ and a set of generation parameters γ :

$$\hat{w}_i \sim \hat{p}(w_i), \gamma. \quad (2)$$

The generation parameters γ control how words are sampled from the probability distribution over the next word. For transformer-based models (e.g., GPT, Llama), these include the “temperature”, “top K”, and “top P”, which control whether to sample a high-probability next token (leading to simple, repetitive text) or favor low probability next tokens (leading to more creative but potentially nonsensical text).¹

¹The full details of GPT’s generation parameters are beyond the scope of this paper. See Platen (2020).

2.2 Controlling synthetic text generation

Thus, applied researchers who would like to influence which token \hat{w}_i is produced next have three options: they can *adapt* the parameters θ used to change the distribution $\hat{p}(w_i)$, they can *prompt* by changing the previous sequence of tokens $(w_{i-1}, w_{i-2}, \dots)$, or they can vary γ to change how the next token is sampled from the distribution over the next token. These techniques are general, working on current transformer-based neural networks, but also on older technologies such as recurrent neural networks like LSTMs and likely on future language models as well.

The *adaptation* approach updates the weights θ of a pretrained model to affect the content or style of generated text. Off-the-shelf pretrained language models may not reflect a researcher’s desired style or content, especially if it was not present in the original pretraining data. Adapting an off-the-shelf model consists of providing it with additional unlabeled text from a specific domain and updating the weights θ in the model to guide the text that the model produces.² An adaptation approach is used in the third application to generate synthetic tweets about the war in Ukraine.

Second, a researcher can use *prompting* to guide synthetic text generation. If a researcher provides the beginning of a document or an instruction (w_{i-1}, \dots, w_n) , an LLM can generate a plausible continuation of the document. For example, a researcher can generate a news story describing police making arrests by hand-writing a headline related to arrests. A prompting approach is illustrated in the first application to generate news stories describing police actions in India by providing manually written headlines to elicit stories with the desired event types.

Recent “instruction tuned” language models can generate text from general prompts that describe the desired output, rather than simply the starting tokens of some desired text

²The natural language processing literature uses several terms to describe the process of updating a pretrained model’s weights using new text, including “fine-tuning”, “domain adaptation”, or “additional pretraining”. To avoid confusion, I use the term “adaptation” to refer to updating weights in a pretrained model to perform better on the language modeling task on a new corpus of unlabeled text.

(Liu et al. 2021). For instance, GPT-3.5 or 4 can be provided with a prompt such as “write a political manifesto supporting free trade” and obtain a plausible output without the need for a specific prompt or adapting a language model on a corpus of political manifestos. This approach is useful when generating the desired content requires a definition or explanation and for types of text that do not have a natural summary–text format like news stories with their headlines. Prompting with an explanation of the desired text is used in the second application to generate populist party manifestos in 22 languages for 27 European countries.

Third, a researcher can vary the generation parameters γ to change how the next token is sampled from the distribution over the next token. Varying the generation parameters generally changes the style of the text, rather than its content, and is useful in conjunction with either adaptation or prompting to produce more realistic text. This approach is described more fully below and is used in the final application to generate more realistic tweets about the war in Ukraine.

When should researchers use *adaptation* and when should they *prompt* to guide a document’s content? Table 1 provides an overview of how each technique addresses the three challenges. Adaptation may be necessary when a researcher’s text is outside the domain of the training data of the pretrained model, can expand a small set of labeled documents into a large corpus, and provide them with a version of their dataset that they can share freely, or while avoiding the need for prompts. However, adaptation is more technically challenging, requiring an existing corpus of text and GPU infrastructure for training.

Prompting has several other advantages over adaptation, beyond its ability to work without an existing reference corpus. Prompts are relatively transparent and can be published, allowing other researchers can assess whether descriptive prompts accurately describe the concept being prompted (see, for example, the definition of populism used in the second application). Finally, prompts are easy to write and tweak to produce synthetic text that

	Advantages/Uses	Limitations	Examples
Adaptation $p(w_i w_{i-1}, \dots, \theta)$	<ul style="list-style-type: none"> • Provides text very similar to existing corpus • Useful for expanding rare document classes 	<ul style="list-style-type: none"> • Requires existing corpus • Computationally expensive 	<ul style="list-style-type: none"> • Ukraine war tweets (App. 3)
Prompting $p(w_i w_{i-1}, \dots, \theta)$	<ul style="list-style-type: none"> • No existing corpus needed • Simple to implement • Easily controllable 	<ul style="list-style-type: none"> • Local, open source models with abstract prompts • Requires manual experimentation 	<ul style="list-style-type: none"> • News stories about police responses to communal violence (App. 1) • Populist sentences (App. 2)
Sampling $\hat{w}_i \sim \hat{p}(w), \gamma$	<ul style="list-style-type: none"> • Can improve text quality • Used in conjunction with adaptation or prompting • No training of the model required 	<ul style="list-style-type: none"> • Controls style more than content • Requires existing corpus for adversarial method • If no corpus is available, requires manual tuning 	<ul style="list-style-type: none"> • Ukraine war tweets (App. 3)

Table 1: *Overview of the three approaches to controlling synthetic text generation.*

matches the desired context (for example, in the first application to ensure that a range of police actions are included in the training corpus.) However, it requires manual experimentation to find good prompts, and will fail if the task is outside the LLM’s pretraining data.

2.3 Improving synthetic text quality with an adversarial classifier

Varying γ , the sampling or generation hyperparameters, greatly affects the quality of the generated text, but little theoretical guidance exists on how to select them (Fu et al. 2021). I introduce a simple “adversarial” procedure for selecting the best generation

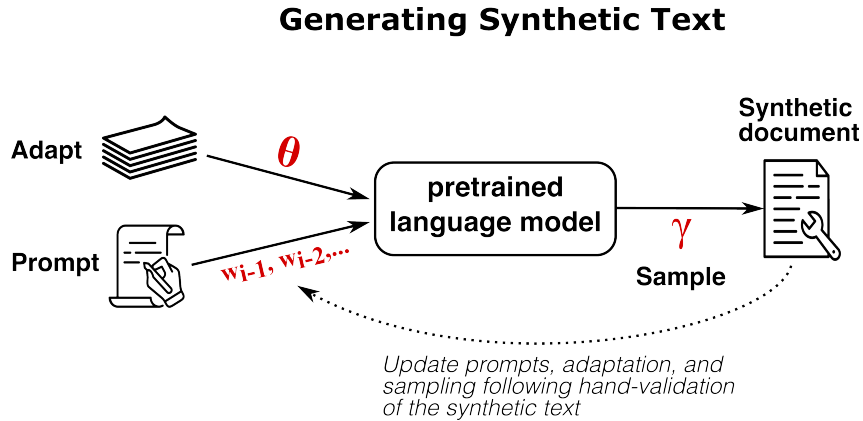


Figure 1: Overview of options for controlling synthetic text generation. Researchers can affect the content and style of synthetic documents by changing language model parameters (θ), by providing new prompts (w_{i-1}, w_{i-2}, \dots), or by changing the sampling parameters (γ). Researchers then decide how to use the synthetic text as training data.

hyperparameters for generating text. The less a classifier can distinguish between real and synthetic text, the higher the quality of synthetic text, and thus the more useful the synthetic text will be for training supervised learning models.

When using the technique, for each set of hyperparameters γ , a researcher generates n synthetic documents from γ and samples n real documents from the existing corpus. They then train a classifier on a training set of both synthetic and real documents with the objective of predicting whether the document is real or synthetic. The set of hyperparameters that results in the lowest classification accuracy is the set that generates the most realistic synthetic documents and should be used to generate the final synthetic text.

2.4 Overview of the synthetic text pipeline

Figure 1 shows an overview of the options that researchers have for controlling the content and style of synthetic text. The adversarial technique helps researchers select the best generation parameters if they have access to an existing corpus of real text, and in any case, they can conduct hand-validation of the synthetic text and adjust their guidance strategy to improve the quality of the synthetic text.

2.5 Using synthetic text as training data

After generating synthetic text, researchers have three options for how to use it in a supervised learning pipeline. First, they can use the synthetic text as a form of *data augmentation*. Adding synthetic data to a limited set of hand-labeled data can increase the size and diversity of the training set. Application 1 demonstrates that synthetic augmentation produces classifiers that are more accurate in classifying police responses to communal violence in India than classifiers trained on hand-labeled data alone, or, equivalently, can produce a desired level of accuracy with fewer human annotations. The augmentation approach is especially valuable when the concept of interest is rare, as it allows researchers to generate a large number of positive synthetic examples.

Second, they can use synthetic text for a form of *model distillation*, training a supervised model directly on the synthetic text. Model distillation refers to the process of transferring knowledge from a large, complex model to a smaller, simpler model (Hinton, Vinyals, and Dean 2015). In labeling documents, researchers do not need the full capabilities of a large language model, but only the portion of the network that encodes the desired information about the political science document annotation task. By generating synthetic text from the LLM and training a smaller supervised model on the synthetic text, researchers can distill the relevant knowledge of the LLM into a smaller, simpler model that can be used to annotate real documents. This approach is used in Application 2—the knowledge that GPT 3.5 has of populist rhetoric is transferred using synthetic text to a specialized, efficient, and accurate sentence-level model.

Third, in some limited circumstances, researchers can hand-label synthetic text directly to use as training data. Application 3 shows that this approach carries costs in accuracy. A hand-labeled synthetic tweet is less informative to the model than a labeled real tweet. In some cases, however, this approach might be useful, for instance, to produce a hand-labeled, copyright-free benchmark dataset for other researchers, in situations where real

text is too sensitive or restricted to show annotators, or when researchers do not have direct access to the raw text data.

2.6 Using the classifier

Finally, researchers discard the synthetic training data and apply the newly trained classifier to real text to obtain predicted labels. Training a smaller model on synthetic text allows researchers to use the full capabilities of a large language model to generate synthetic text, but then use a smaller, simpler, reproducible model to make predictions. The small model is more interpretable, faster, and cheaper to run than the large language model, can be used in the future to classify new documents, and can be shared with other researchers.

2.7 Ethics

Generating synthetic text presents serious ethical concerns. Synthetic text can include factual errors, conspiracy theories, or offensive statements. To avoid any possibility of synthetic text being mistaken for real text, researchers working with synthetic text should always attach a disclaimer directly to any synthetic text any time it is saved or stored, clearly indicating that the text is synthetic.³ Annotators should be briefed on the use of synthetic text and the annotation interface should clearly state that they are working with synthetic text. Any synthetic training text reported in published work must be clearly marked (e.g. [SYNTH]).

Language models learn the biases present in their training data (Caliskan, Bryson, and Narayanan 2017). Researchers should validate that the models that they train on synthetic text are not relying on group stereotypes when making predictions. Deliberately writing prompts that break the association between groups and stereotyped traits offers a partial

³For example, `<!--SYNTHETIC TEXT! Do not trust the factual content of this text. Generated by <author>, <email> to train a populist speech classifier. -->`

solution, but greater research into the prevalence and mitigation of these harms is needed.

Despite some recent proposals to use language models to approximate survey responses (Argyle et al. 2023), synthetically generated text should *never* be used to draw substantive conclusions. While synthetic text may be difficult to distinguish from real text and thus useful for training a model to recognize certain linguistic patterns, its factual content will be imaginary and thus completely unsuitable for answering substantive questions on its own. Instead, synthetic text should only be used to train a model, then discarded. Only real text should be used for analysis after the model is trained.

2.8 Previous work on synthetic data

The idea of using synthetic data to train a model is well established in the machine learning literature, especially in the context of data augmentation. In the NLP literature, researchers have generated semi-synthetic text using rules to change words (Wei and Zou 2019), or by using language models to generate similar sentences (Schick and Schütze 2021). More recent work in the natural language processing literature has used language models to generate synthetic text training data (Meng et al. 2022; Ye et al. 2022; He et al. 2022). In recent months, computer science researchers have begun using synthetic instruction-tuning text as a form of model distillation, generating training data from a large, capable model such as GPT-3.5 and using the synthetic text to train smaller, open weight models (Taori et al. 2023; Gunasekar et al. 2023).

3 Applications

The following section presents three short applications that illustrate the decisions researchers face in generating synthetic text, including when to adapt and when to prompt. Where possible, I use local LLMs to enhance reproducibility and to demonstrate the efficacy of even small or obsolete models in generating synthetic text.

3.1 Application 1: Labeling rare event types with synthetic augmentation: Identifying police actions in the 2002 Gujarat violence

Following the death of 59 Hindu pilgrims in a train fire in late February 2002, the state of Gujarat experienced widespread communal violence, directed primarily at Muslims. The qualitative research on the violence emphasizes the role of the police in the violence, including their failure to respond to ongoing violence (Wilkinson 2006; Subramanian 2007). Halterman et al. (2021) introduce a new annotated corpus, the IndiaPoliceEvents dataset, focusing on police actions taken during the 2002 violence in Gujarat, India. The dataset includes 1,257 articles (21,391 sentences) from the *Times of India* covering the period after the train fire. Each article is labeled at the sentence level by two or three trained annotators for a set of actions taken by police. The dataset also reflects the common rare class problem in social science document classification: only 9.7% of sentences involve any police action, and specific actions are rarer (only 0.45% of sentences describe police making arrests).

I conduct an experiment to evaluate the benefit of using synthetic data to augment hand-labeled data. I simulate a best-case process of hand annotation by sampling sentences and their gold standard labels from the IndiaPoliceEvents dataset to create a training set. The current best practice in annotating rare documents is to use active learning, where the classifier is used during the annotation process to select the most informative document for hand labeling (Miller, Linder, and Mebane 2020). I use uncertainty-based active learning to select the most informative documents for “annotation” (or, in this case, to use the label provided by the dataset). For a classifier, I use an efficient neural-based classifier, embedding sentences using a sentence transformer (Reimers and Gurevych 2019) model and fitting a logistic regression model on the (frozen) embeddings.

To augment the real annotated documents, I generate 3,000 synthetic examples on police

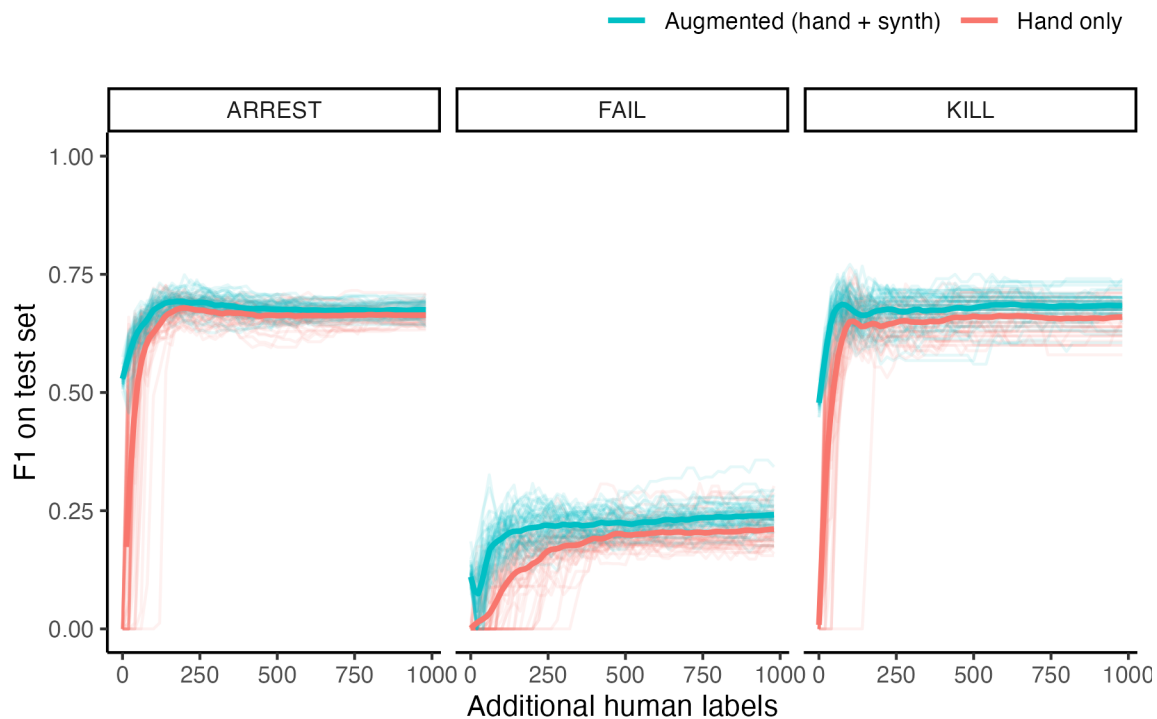


Figure 2: *A classifier trained on hand-labeled data augmented with synthetic data outperforms a classifier trained on hand-labeled data only. As the number of hand-labeled examples increases, the F1 score on a held-out test set of real data increases. The “augmented” classifier includes the full collection of synthetic data at each step, along with the same number of hand-labeled examples as the “hand only” classifier. Each thin line represents an different set of hand-labeled examples used to start the active learning process, with the average overlaid.*

actions by prompting GPT-2 (large) with a set of hand-written headlines to elicit police actions, other mentions of police, and non-police events. I write 2-6 headlines for each event type in the India Police Events event schema (see SI A.1) and generate multiple stories for each headline. I use GPT-2 in order to demonstrate that even a relatively small generative model, run locally, can produce useful synthetic data.

Figure 2 shows the a comparison of classifiers trained on real text only and synthetically augmented text. The augmented classifier trained on both hand-labeled and synthetic text outperforms a classifier trained on only hand-labeled data for every amount of human labeling for all three event classes. While the performance is poor for the police FAIL to act class (as Halterman et al. (2021) also report), the relative improvement with synthetic augmentation is still substantial. Given the reported annotation time in Halterman et al. (2021), annotating the full set of 1,000 sentences with all annotators would take around 6.6 hours of work.

This application demonstrates that even a small set of synthetic augmentation data, generated using a relatively old LLM, improves the performance of a supervised model trained on hand-annotated text alone, even when using active learning techniques.

3.2 Application 2: Synthetic data without hand labeling—training a sentence-level populist classifier

As attention to populist parties has grown, so too has the methodological work on identifying populism in text, including in party manifestos (Rooduijn and Pauwels 2011; Hawkins et al. 2019; Di Cocco and Monechi 2021; Dai and Kustov 2022; Jankowski and Huber 2023; Breyer 2022). A key challenge has been to identify populism in short text, such as a sentence or a paragraph, in order to estimate the degree or amount of populism in a longer document. This application illustrates a new method for identifying populism at the sentence level across 27 European countries in 22 languages. It use a *prompting*

approach to generate synthetic populist manifesto statements using GPT-3.5-turbo. It then fits a classifier on the synthetic data alone to identify populist statements in real manifesto text. I find evidence that the classifier trained on synthetic text alone reliably identifies populist rhetoric in text at the sentence level.

3.2.1 Measuring populism

I employ a conceptualization of populism drawing on Mudde’s (2004) “thin” definition of populism, which focuses on its rhetorical aspects and worldview rather than on specific policy positions. Because GPT-3.5 was trained on a multilingual corpus, a prompt can specify a country and language to obtain non-English training text, even with an English language prompt. By inserting each country and its associated language(s) into the prompt and varying the sampling hyperparameters (γ), I generate a set of 5,357 synthetic populist sentences.

Description	Prompt
Populist (thin definition)	<p>Populist rhetoric sees politics as a conflict with good, common, or “real” people on one side, and out-of-touch, evil, or self-serving elites on the other.</p> <p>Write ten statements that a populist party in {country} might make (in {language}):</p> <p><i>Example output: [SYNTH] “We’re committed to giving voice to those who have been ignored or left behind by mainstream politics.”</i></p>
Populist (style prompt)	<p>A populist party in {country} believes that politics is corrupted by self-interested elites, unelected bureaucrats, croynism, and big business. It wants to take power back for ordinary people. Write 12 statements that a {country_adjective} populist party might make (in the {language} language) in the style of a political manifesto:</p> <p><i>Example output: [SYNTH] “We believe that the people of Ireland are sovereign, and that the government should be accountable to them.”</i></p>

Table 2: Prompts used to generate populist text with language and country placeholders.

I also generate 36,509 negative examples—instances of non-populist text for the 22 countries by prompting the LLM with policy position descriptions from the Manifesto Project codebook (Volkens et al. 2021) and an additional set of ten hand-written prompts to cue criticism of other political parties and dissatisfaction with current policies.

Putting all 1.1 million sentences through a commercial LLM could cost several thousand dollars and its classification performance would depend on the ability of the model to precisely apply technical definitions from the codebook.⁴

I then train a supervised text classifier on the synthetic sentences. Specifically, I use SetFit, an efficient transformer-based model for short text classification (Tunstall et al. 2022) to fine-tune a RoBERTa-based multilingual transformer model (Reimers and Gurevych 2020). In training the model, I assume that the statements generated with the populist prompt are indeed examples of populist rhetoric, and that non-populist prompts generate non-populist rhetoric.

3.2.2 Validating the populism classifier—performance on known manifesto labels

As a first validation, I examine how well a classifier trained on synthetic text can recover known labels from the Manifesto Project (Volkens et al. 2021). For each of the ten most commonly used policy codes, I train a binary classifier using the same architecture on the synthetic text generated from the Manifesto Project codebook descriptions discussed above. Because each of these codes are applied to real text by the Manifesto Project, I can perform an evaluation of the model on the real, hand-labeled text. I compare the results to the performance of a classifier trained on the same number of real manifesto statements, finding that the synthetic-only classifier performs almost as well on each category (Table 3). This serves as a validation that the model can recover known labels from real text,

⁴1.1 million documents, with around 1,000 tokens per document+codebook prompt at 0.005 dollars per 1,000 tokens equals \$5,500. Generating the synthetic training data cost around \$2.

Code	Description	Synth F1	Hand F1
201	Freedom and Human Rights	0.74	0.77
304	Political Corruption	0.80	0.83
402	Incentives: Positive	0.66	0.74
403	Market Regulation	0.62	0.66
414	Economic Orthodoxy	0.75	0.80
416.2	Sustainability: Positive	0.77	0.82
502	Culture: Positive	0.85	0.86
504	Welfare State Expansion	0.71	0.75
703	Agriculture and Farmers: Positive	0.83	0.83
705	Underprivileged Minority Groups	0.70	0.78

Table 3: *Performance of classifiers trained on synthetic text and real labeled text and evaluated on real Manifesto Project text with gold-standard labels.*

even when trained on purely synthetic text.

3.2.3 Validating the populism classifier–hand labeling

I then validate the populism scores directly using a small set of hand-annotated statements. I apply the newly trained populism classifier to each of the 1.04 million sentences or phrases in the Manifesto Project corpus (Volkens et al. 2021), producing a continuous predicted $[0, 1]$ populism score for each sentence in the manifesto corpus. I then collect hand annotations on a sample of 450 English-language manifesto statements from the Manifesto Project and evaluate the predictions of the model. It achieves an accuracy of 0.87 and a macro F1 score of 0.76. Validating the model’s cross-lingual performance is more difficult, given the lack of a labeled dataset of populist sentences in languages other than English and is left for future work. However, because the model is trained on text in all 22 languages, it does not need to do the challenging task of classifying languages outside the training set.

Examining the scored sentences by hand provides some further validity for the method. Given all manifesto sentences from the United Kingdom Independence Party, a populist party, the model identifies the three sentences as having the highest populism scores (see SI Table A4):

- “Politics is corrupted by self-interest and big business.”
- “These professional politicians don’t want us to run our own country or control our own lives.”
- “An unaccountable elite revels in mutual back-scratching and cronyism.”

3.2.4 Applying the Classifier Populism in Populist Party Manifestos

Finally, I use the model’s sentence-level predictions to generate manifesto-level populism scores for each party in the Manifesto Project dataset. I calculate the average populism score of sentences in each party’s manifesto, along with the proportion of sentences with a predicted populism score above 0.5.

Investigating the apparent false negatives of the classifier provides some further validation for the method. Table A5 lists the 10 parties that are coded as populist in the PopuList dataset (Rooduijn et al. 2019) that have no sentences with a predicted populism score above 0.5. I randomly select the Northern League’s 2013 manifesto for hand coding. While some of the sentences include some statements that are populism-adjacent, including opposition to bank bailouts, or right wing, such as support for the traditional family, none contain overt statements of hostility to elites or support for “common people” against elites (SI A.3).

This application shows that synthetic data can address a methodological issue that has vexed the study of populism: how can we identify populist rhetoric in short text across languages? Employing substantive knowledge about populism, we can generate populist and non-populist text from an LLM. Training a supervised classifier on this text allows us to distil the knowledge of the LLM into a smaller, tailored model with high accuracy.

3.3 Application 3: Generating Synthetic Tweets with Adapted Language Models to Identifying Weapons in the Ukraine War

As political scientists collect more data about the ongoing war in Ukraine (Zhukov 2022), supervised models trained on social media posts will provide an important source of information. This application shows a named entity recognition model trained on hand-labeled synthetic text can match the accuracy of a model trained on real tweets. However, it highlights a limitation of synthetic data: a marginal hand-annotated synthetic tweet is less useful for training a supervised model than an annotated real tweet. This gap in performance is lessened by tuning the generation parameters to produce more realistic text, but suggests that hand annotation of synthetic text should only be used in limited circumstances, such as when real training text is unavailable, is too sensitive to show to annotators, or when reproducibility of a specific methodological approach is a top priority.

I collect a set of around 20,000 real tweets from four Twitter accounts that report detailed information on the fighting in Ukraine from the first two months after the invasion.⁵ Because the synthetic tweets should closely match the actual tweets, I opt for an adaptation approach to text generation. I adapt GPT-2 (large) because it can be run and adapted locally with standard hardware, and performs well on the task of generating tweets.⁶

To improve the quality of the synthetic tweets, I apply the adversarial method introduced above. Across 56 combinations of hyperparameters, the classification accuracy of a BERT-based classifier ranges from 0.99 to 0.67, indicating a large effect of hyperparameters of tweet generation.⁷ An ideal set of synthetic tweets would be indistinguishable from real tweets: the fact that they are distinguishable indicates that while useful for the reasons discussed above, a marginal synthetic tweet is less useful for training a supervised model

⁵Specifically, @uaweapons, @osinttechnical, @oryxspioenkop, and @markito0171.

⁶See SI section A.5.1 for example synthetic tweets produced by GPT-3.5 and GPT-4.

⁷See SI Figure SI 1.

than an annotated real tweets.

3.3.1 Comparing performance of real and synthetic tweets

How well do these synthetic tweets work in practice to train a supervised model? Existing named entity recognition systems do not identify weapons in text, meaning that researchers who wish to identify weapons automatically must train their own NER model. I hand annotate 1,600 tweets with span-level labels on the specific weapons systems described in the tweets. The annotated set includes 200 real tweets, 600 synthetic tweets generated from non-optimized parameters, 600 tweets generated using the parameters selected by the adversarial tuning method, with an additional 200 real tweets as evaluation data. I train a named entity recognition (NER) model to identify mentions of specific weapons in the text.⁸ Figure 3 reports the accuracy for the NER model trained on actual and synthetic tweets at different training set sizes and evaluated on labeled actual tweets. A marginal non-optimized synthetic tweet is significantly less valuable than a real labeled tweet: the model requires 500 non-optimized synthetic training examples to reach the performance it can attain with 200 labeled real tweets. Applying the adversarial technique to select the optimal generation hyperparameters reduces the gap significantly: only 300 synthetic tweets are required to reach the same performance as the real tweet model.

While labeling synthetic tweets carries a cost, namely that they are somewhat less useful than a marginal actual tweet in training a classifier, it provides a major benefit in that the labeled training data can be published without violating the Twitter terms of service. Publishing the complete training data is especially important for methodological work, where replicating a model’s performance and allowing others to build on previous datasets is a key component of progress. Unfortunately, the actual tweets that a researcher then applies the model to cannot be published, meaning that only their methodological, not

⁸I use spaCy 3.1.2’s small `en_core_web_sm` model as a base and the default training values set by Prodigy (Montani and Honnibal 2018). Better absolute performance could be achieved with a larger model, but I expect the relative performance to be the same.

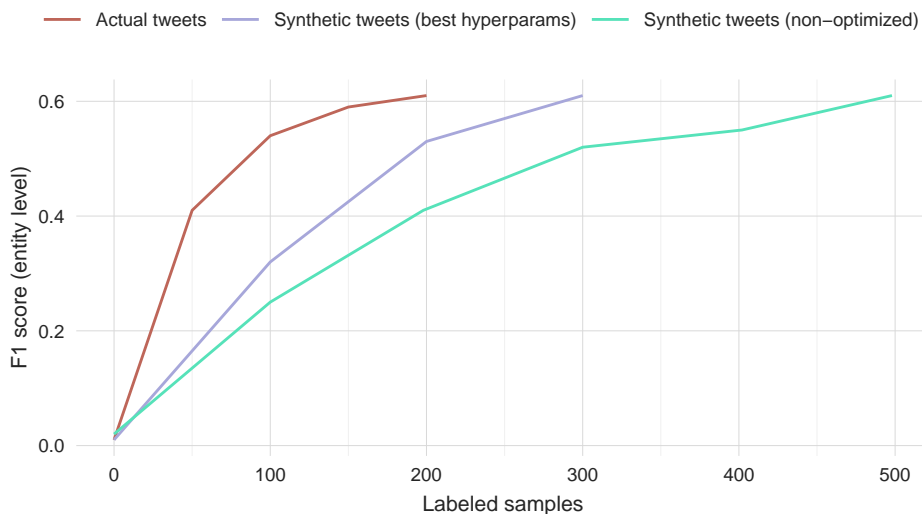


Figure 3: *Test set performance of a named entity recognition model detecting a WEAPON class, trained on annotated actual tweets and annotated synthetic tweets. A model trained on annotated synthetic tweets generated from the optimal hyperparameters requires around 300 examples to reach the same performance as 200 annotated real tweets. Non-optimized tweets require 500 annotations to reach the same performance.*

substantive, claims can be reproduced.

4 Conclusion

As political scientists increasingly turn to large language models as useful tools for analyzing text, they face decisions about how to use these models in a transparent, reproducible, and explainable way. This paper shows that using LLMs for their original strength—generating text—can help researchers overcome the challenges of labeling text for supervised text analysis. By using large language models to generate synthetic text and training classic supervised models, researchers can evaluate the quality of the generated text and employ classifiers that are faster to run and easier to understand.

By controlling the content and style of synthetic text through adaptation, prompting, and sampling parameters, researchers can generate synthetic text that represents the domain of text they are interested in. This synthetic text can be used to augment hand-labeled data

or used directly as a form of model distillation. In limited circumstances where privacy or reproducibility concerns necessitate it, researchers may wish hand-label synthetic text, although this approach carries a cost in accuracy.

Researchers in natural language processing are developing more sophisticated techniques for controlled text generation to allow covariates to guide text generation or ensure the factuality of the generated text. Once they mature, these methods will offer additional options for applied researchers to generate controlled text. However, these methods will still operate either by modifying the conditioning sequence of text ($w_{i-1}...w_1$), modifying the language model’s parameters (θ), or changing how words are sampled from the probability distribution (γ).

5 Funding

This work was supported by the Political Instability Task Force (PITF). The PITF is funded by the Central Intelligence Agency. The views expressed in this paper are the author’s alone and do not represent the views of the US Government.

6 Acknowledgements

Thank you to Benjamin Bagozzi, Jill Irvine, Michael Jankowski, Adam Lauretig, Erin Rossiter, Brandon Stewart, and prior anonymous reviewers for helpful comments. This work was first presented at PolMeth 2022 and New Directions in Text as Data 2022. MIT’s Lincoln Laboratory provided computing resources (Reuther et al. 2018).

7 Data Availability Statement

All code and uncopyrighted text used in this paper will be available on the Harvard Dataverse.

8 References

- Argyle, Lisa P, Ethan C Busby, Nancy Fulda, Joshua R Gubler, Christopher Rytting, and David Wingate. 2023. “Out of One, Many: Using Language Models to Simulate Human Samples.” *Political Analysis* 31 (3): 337–51.
- Breyer, Magdalena. 2022. “Populist Positions in Party Competition: Do Parties Strategically Vary Their Degree of Populism in Reaction to Vote and Office Loss?” *Party Politics*, 13540688221097082.
- Caliskan, Aylin, Joanna J Bryson, and Arvind Narayanan. 2017. “Semantics Derived Automatically from Language Corpora Contain Human-Like Biases.” *Science* 356 (6334): 183–86.
- Dai, Yaoyao, and Alexander Kustov. 2022. “When Do Politicians Use Populist Rhetoric? Populism as a Campaign Gamble.” *Political Communication*, 1–22. <https://doi.org/https://doi.org/10.1080/10584609.2022.2025505>.
- Di Cocco, Jessica, and Bernardo Monechi. 2021. “How Populist Are Parties? Measuring Degrees of Populism in Party Manifestos Using Supervised Machine Learning.” *Political Analysis*, 1–17.
- Fu, Zihao, Wai Lam, Anthony Man-Cho So, and Bei Shi. 2021. “A Theoretical Analysis of the Repetition Problem in Text Generation.” In *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:12848–56. 14.
- Gilardi, Fabrizio, Meysam Alizadeh, and Maël Kubli. 2023. “ChatGPT Outperforms Crowd-Workers for Text-Annotation Tasks.” *arXiv Preprint arXiv:2303.15056*.
- Gunasekar, Suriya, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, et al. 2023. “Textbooks Are All You Need.” *arXiv Preprint arXiv:2306.11644*.
- Halterman, Andrew, Katherine A Keith, Sheikh Muhammad Sarwar, and Brendan O’Connor. 2021. “Corpus-Level Evaluation for Event QA: The IndiaPoliceEvents Corpus Covering the 2002 Gujarat Violence.” *Findings of the Association for*

Computational Linguistics.

- Hawkins, Kirk A, Rosario Aguilar, Bruno Castanho Silva, Erin K Jenne, Bojana Kocijan, and Cristóbal Rovira Kaltwasser. 2019. “Measuring Populist Discourse: The Global Populism Database.” In *EPSA Annual Conference in Belfast, UK, June, 20–22*.
- He, Xuanli, Islam Nassar, Jamie Kiros, Gholamreza Haffari, and Mohammad Norouzi. 2022. “Generate, Annotate, and Learn: NLP with Synthetic Text.” *Transactions of the Association for Computational Linguistics* 10: 826–42.
- Hinton, Geoffrey, Oriol Vinyals, and Jeff Dean. 2015. “Distilling the Knowledge in a Neural Network.” *arXiv Preprint arXiv:1503.02531*.
- Jankowski, Michael, and Robert A Huber. 2023. “When Correlation Is Not Enough: Validating Populism Scores from Supervised Machine-Learning Models.” *Political Analysis*, no. 1–15. <https://doi.org/doi:10.1017/pan.2022.32>.
- Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. “Pre-Train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing.” *arXiv Preprint arXiv:2107.13586*.
- Longpre, Shayne, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, et al. 2023. “The Flan Collection: Designing Data and Methods for Effective Instruction Tuning.” In *International Conference on Machine Learning*, 22631–48. PMLR.
- Meng, Yu, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. “Generating Training Data with Language Models: Towards Zero-Shot Language Understanding.” *Advances in Neural Information Processing Systems* 35: 462–77.
- Miller, Blake, Fridolin Linder, and Walter R Mebane. 2020. “Active Learning Approaches for Labeling Text: Review and Assessment of the Performance of Active Learning Approaches.” *Political Analysis*, 1–20.
- Montani, Ines, and Matthew Honnibal. 2018. “Prodigy: A New Annotation Tool for Radically Efficient Machine Teaching.” *Artificial Intelligence* to appear.

- Mudde, Cas. 2004. “The Populist Zeitgeist.” *Government and Opposition* 39 (4): 542–63.
- Ornstein, Joseph T., Blasingame Elise N., and Jake S. Truscott. 2022. “How to Train Your Stochastic Parrot: Deep Language Models for Political Texts.” *PolMeth Conference Paper*.
- Peskoff, Denis, and Brandon M Stewart. 2023. “Credible Without Credit: Domain Experts Assess Generative Language Models.” In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 427–38.
- Platen, Patrick von. 2020. “How to Generate Text: Using Different Decoding Methods for Language Generation with Transformers.” *Hugging Face Blog*.
- Reimers, Nils, and Iryna Gurevych. 2019. “Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks.” In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. <https://arxiv.org/abs/1908.10084>.
- . 2020. “Making Monolingual Sentence Embeddings Multilingual Using Knowledge Distillation.” In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. <https://arxiv.org/abs/2004.09813>.
- Reuther, Albert, Jeremy Kepner, Chansup Byun, Siddharth Samsi, William Arcand, David Bestor, Bill Bergeron, et al. 2018. “Interactive Supercomputing on 40,000 Cores for Machine Learning and Data Analysis.” In *2018 IEEE High Performance Extreme Computing Conference (HPEC)*, 1–6. IEEE.
- Rooduijn, Matthijs, and Teun Pauwels. 2011. “Measuring Populism: Comparing Two Methods of Content Analysis.” *West European Politics* 34 (6): 1272–83.
- Rooduijn, Matthijs, Stijn Van Kessel, Caterina Froio, Andrea Pirro, Sarah De Lange, Daphne Halikiopoulou, Paul Lewis, Cas Mudde, and Paul Taggart. 2019. “The Populist: An Overview of Populist, Far Right, Far Left and Eurosceptic Parties in Eu-

rope.”

- Rytting, Christopher Michael, Taylor Sorensen, Lisa Argyle, Ethan Busby, Nancy Fulda, Joshua Gubler, and David Wingate. 2023. “Towards Coding Social Science Datasets with Language Models.” *arXiv Preprint arXiv:2306.02177*.
- Schick, Timo, and Hinrich Schütze. 2021. “Generating Datasets with Pretrained Language Models.” In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 6943–51.
- Spirling, Arthur. 2023. “Why Open-Source Generative AI Models Are an Ethical Way Forward for Science.” *Nature* 616 (7957): 413–13.
- Subramanian, Kadayam Suryanarayanan. 2007. *Political Violence and the Police in India*. SAGE Publications India.
- Taori, Rohan, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. “Alpaca: A Strong, Replicable Instruction-Following Model.” *Stanford Center for Research on Foundation Models*. <https://Crfm.Stanford.Edu/2023/03/13/Alpaca.Html> 3 (6): 7.
- Tunstall, Lewis, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. “Efficient Few-Shot Learning Without Prompts.” *arXiv Preprint arXiv:2209.11055*.
- Volken, Andrea, Tobias Burst, Werner Krause, Pola Lehmann, Theres Matthieß, Nicolas Merz, Sven Regel, Bernhard Weels, and Lisa Zehnter. 2021. “The Manifesto Data Collection. Manifesto Project (MRG/CMP/MARPOR).” *Version 2021a, Berlin: Wissenschaftszentrum Berlin für Sozialforschung (WZB)*. <https://doi.org/https://doi.org/10.25522/manifesto.mpds.2021a>.
- Wei, Jason, and Kai Zou. 2019. “EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks.” *arXiv Preprint arXiv:1901.11196*.
- Wilkinson, Steven I. 2006. *Votes and Violence: Electoral Competition and Ethnic Riots in India*. Cambridge University Press.

- Wu, Patrick Y, Joshua A Tucker, Jonathan Nagler, and Solomon Messing. 2023. “Large Language Models Can Be Used to Estimate the Ideologies of Politicians in a Zero-Shot Learning Setting.” *arXiv Preprint arXiv:2303.12057*.
- Ye, Jiacheng, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. “ZeroGen: Efficient Zero-Shot Learning via Dataset Generation.” In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 11653–69.
- Zhukov, Yuri M. 2022. “VIINA: Violent Incident Information from News Articles on the 2022 Russian Invasion of Ukraine.” In *Ann Arbor: University of Michigan, Center for Political Studies*. <https://github.com/zhukovyuri/VIINA>.
- Ziems, Caleb, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. 2023. “Can Large Language Models Transform Computational Social Science?” *arXiv Preprint arXiv:2305.03514*.

Image credit for Figure 1: Flaticon.com

% Reset table counter % Reset table counter

A Supplemental Information

A.1 Headlines for prompting **IndiaPoliceEvents** events

Event Type	Hand-Written Title for Prompting
KILL	Three killed in police firing during communal violence
KILL	Police kill three protesters during Gujarat unrest
KILL	Two killed after police open fire on Gujarat crowd
KILL	One killed, many injured in police firing
ARREST	Two arrested after communal violence
ARREST	Police detain scores in Gujarat violence
ARREST	Dozens detained after night of communal unrest
ARREST	Police announce arrest of protest leaders
ARREST	Police book four for inciting violence
ARREST	Police round up activists in Gujarat
ARREST	Police arrest 20 in Gujarat unrest
FAIL	Police stand by helplessly as communal violence breaks out
FAIL	Police fail to respond to violence
FAIL	Police fail to step in, allowing violence to spread
FORCE	Police lathi-charge crowd in Gujarat
FORCE	Police beat protestors
NONE	Police announce new hiring initiative
NONE	Local assembly holds hearing on police budget
NONE	Four killed in industrial accident
NONE	Six killed in traffic accident
NONE	Four dead after building collapse
NONE	BJP forms new government in Gujarat
NONE	Electricity prices rise in Gujarat
NONE	Agriculture minister announces new subsidies
NONE	Court rules on Gujarat land dispute
NONE	Local assembly holds hearing on police budget
NONE	Commissioner congratulates police academy class
NONE	New hospital opens in Gujarat
NONE	Gujarat student wins national spelling bee
NONE	Gujarat wins national cricket championship

Table A1: *Hand-written headlines used to prompt GPT-2 to generate synthetic text for the India Police Events dataset.*

A.2 India Police Events classifier extensions

Figure 2 in the main text shows the results of a classifier using active learning. It begins with a set of 10 labeled real documents, including one positive document for the class. Figure A1 shows the results of a classifier trained on the same data, but without active learning. The classifier never exceeds an F1 score of 0.0, given the rarity of the positive class. Figure A2 shows the results of a classifier trained on the same data, but with realistic annotator error. Each annotation is a random draw from the set of individual annotator answers for each document. The performance is slightly lower overall, and the best performance is achieved with synthetic augmentation and a relatively small number of hand annotations (fewer than 100).

The full set of events in the India Police Events dataset also includes two aggregated event types: **FORCE** (either **ARREST** or **KILL**) and **ANY** (**ARREST**, **KILL**, or any other mention of police events identified by the annotator).

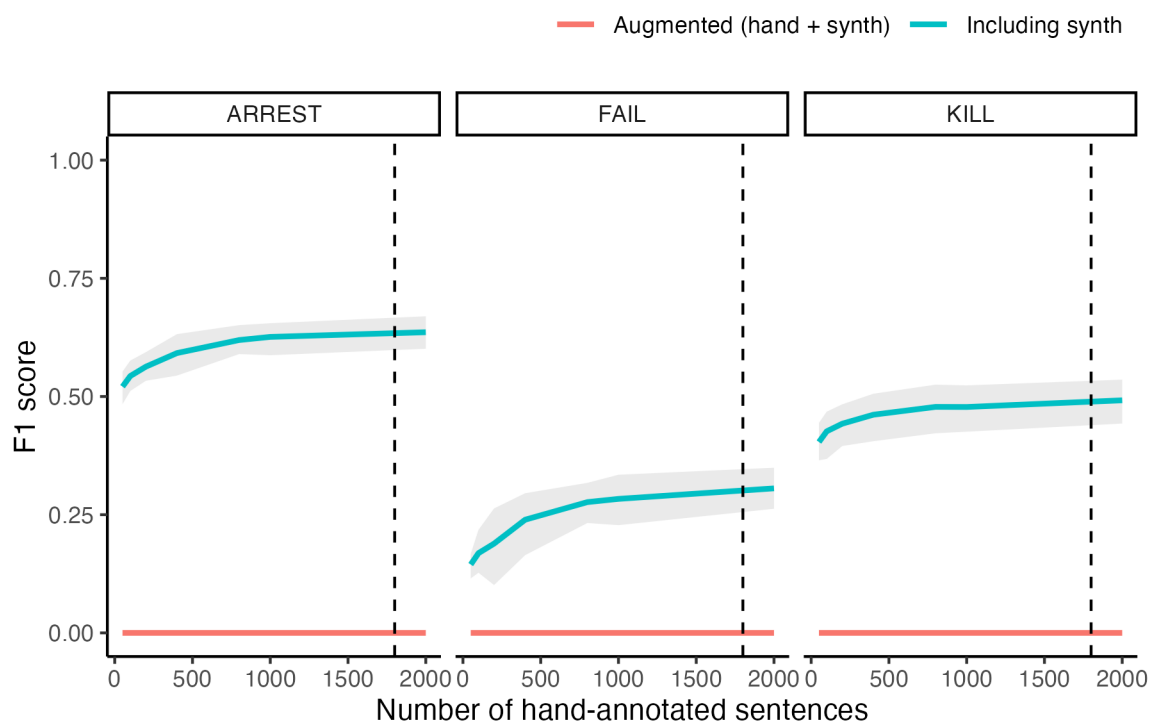


Figure A1: *Alternative results for Figure 2 without active learning. Due to the rarity of the classes, a model trained only on hand-annotated data never exceeds an F1 score of 0.0.*

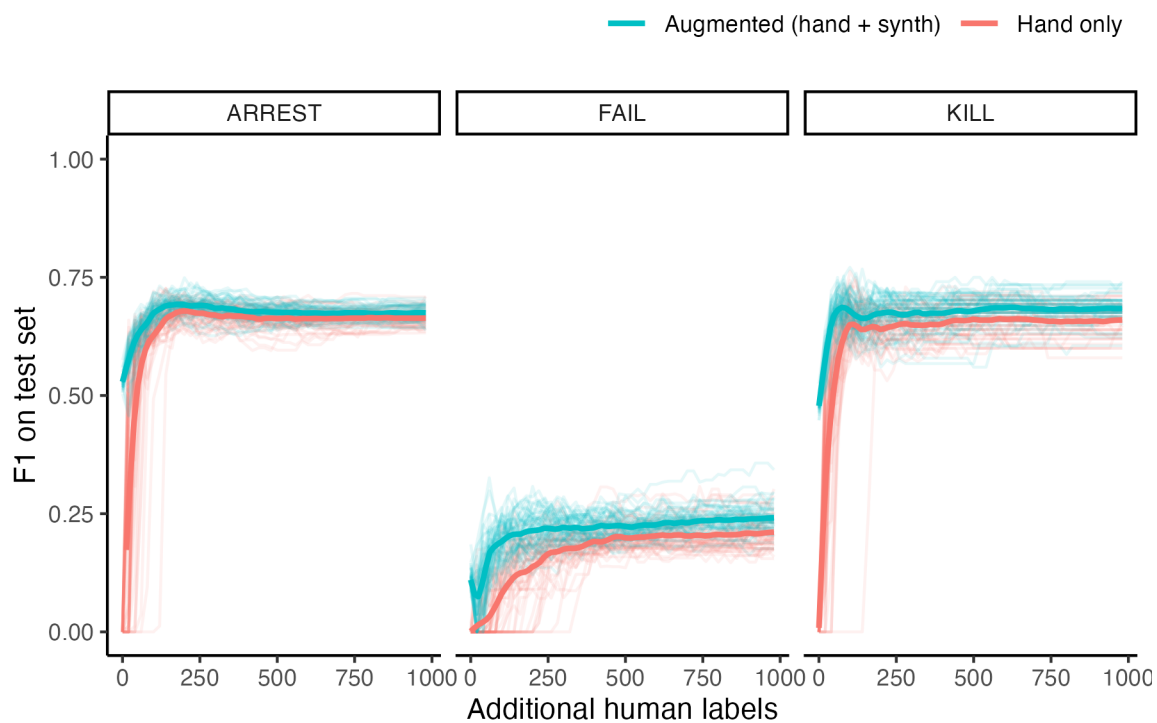


Figure A2: *Alternative results for Figure 2 with realistic annotator error. Rather than using gold-standard triple-annotated sentences, each annotation is a random draw from the set of individual annotator answer for each document. The performance is slightly lower overall, and the best performance is achieved with synthetic augmentation and a relatively small number of hand annotations (fewer than 100).*

A.3 Populism Prompts and Validation

I attempt to provide conceptually clear examples of populist rhetoric by writing prompts that are meant to elicit populist statements from the language model, without conflating populism with other political stances, such as anti-migrant, right-wing, or Eurosceptic positions, or opposition to the party in power.

As a first evaluation, the model trained on synthetic populist and non-populist text achieves an accuracy of 0.93 and macro F1 score of 0.85 in an internal train/test split of the synthetic data. This indicates that the classifier can reliably distinguish between synthetic sentences from populist and non-populist prompts, but does not necessarily generalize to the model’s ability to identify populist statements in real text. The main text includes more useful validations.

Table A7 provides full populism classifier results on the set of hand-annotated manifesto statements. Because populist statements are relatively rare, I stratify the sample to annotate by predicted populism score to take a sample of the 100 highest scoring sentences, 100 lowest scoring sentences, and 250 randomly selected sentences from middle of the predicted range. A small number of sentences are excluded because they were not relevant (e.g., section titles, word fragments).

Table A2 shows an example non-populist prompt using the Manifesto Project codebook. Table A3 below show the prompts used to generate non-populist text.

Table A4 shows 10 randomly selected sentences from United Kingdom Independence Party manifestos and their predicted populism scores.

Economic (414)	Orthodoxy	<p>A political party is calling for economic orthodoxy, the need for economically healthy government policy making. May include calls for:</p> <ul style="list-style-type: none"> • Reduction of budget deficits; • Retrenchment in crisis; • Thrift and savings in the face of economic hardship; • Support for traditional economic institutions such as stock market and banking system; • Support for strong currency. <p>Write a list of 10 statements that this party in {country} might make (in the {language} language) in its party platform:</p>
Welfare State (504.0)	Expansion	<p>A political party supports welfare state expansion. Favourable mentions of need to introduce, maintain or expand any public social service or social security scheme. This includes, for example, government funding of:</p> <ul style="list-style-type: none"> • Health care; • child care; • Elder care and pensions; • Social housing. <p>Note: This category does NOT include education.</p> <p>Write a list of 10 statements that this party in {country} might make (in the {language} language) in its party platform:</p>

Table A2: *Example Comparative Manifesto Project prompts for non-populist text. The ten most common codes are 414, 201.0, 416.2, 504.0, 403.0, 703.0, 304.0, 402, 705.0, 502.0. See Volkens et al. (2021) for details.*

“A political party supports existing political institutions, multiculturalism, globalization, and respect for the existing political process. It supports equality for all people and welcomes immigrants.

Write ten statements that this party in {country} might make (in {language}):”

“A party in {country} believes that taxes are too high. It wants to reduce taxes and cut government spending. Write 10 statements that {adjective} party might make in the style of a political manifesto (in the {lang} language):”

“A party in {country} wants to spend more money on schools, housing, and the military. Write 10 statements that {adjective} party might make in the style of a political manifesto (in the {lang} language):”

“A party in {country} believes that the government should spend more money on healthcare, education, and infrastructure. Write 10 statements that {adjective} party might make in the style of a political manifesto (in the {lang} language):”

“A party in {country} wants to raise the minimum wage. Write 10 statements that {adjective} party might make in the style of a political manifesto (in the {lang} language):”

“A party in {country} is calling for greater support for teachers and police. Write 10 statements that {adjective} party might make in the style of a political manifesto (in the {lang} language):”

“A party in {country} is criticizing the foreign policy of its opposing party. Write 10 statements that {adjective} party might make in the style of a political manifesto (in the {lang} language):”

“A party in {country} is criticizing populism as a threat to {country}. Write 10 statements that {adjective} party might make in the style of a political manifesto (in the {lang} language):”

“A party in {country} is criticizing the high rate of unemployment. Write 10 statements that {adjective} party might make in the style of a political manifesto (in the {lang} language):”

“A party in {country} believes the country is on the wrong path. Write 10 statements that {adjective} party might make in the style of a political manifesto (in the {lang} language):”

“A party in {country} is criticizing the party is is campaigning against. Write 10 statements that {adjective} party might make in the style of a political manifesto (in the {lang} language):”

Table A3: *Hand-written prompts used to generate non-populist text. “adjective” refers to the country adjective such as “Swiss” or “Dutch”.*

Sentence	Score
As a minimum, we will seek continued access on free-trade terms to the EU’s single market.	0.05
A GRAMMAR SCHOOL IN EVERY TOWN The state education system of grammar, secondary modern and technical schools was designed to make a high standard of education available to all, irrespective of social background.	0.06
Climate Change Act, the most expensive piece of legislation in history.	0.06
BRITISH CULTURE	0.05
3.	0.09
In the longer term, we will aim to restore the personal allowance to those earning over £100,000 and make 40 per cent the top rate of tax for all, as it used to be.	0.06
in particular for children, the elderly and people on low incomes.	0.04
We will also exempt foodbanks and charity shops from charges imposed by local authorities to dispose of unwanted food waste and other goods.	0.05
UKIP will restore Britain’s armed forces to their rightful place among the most professional, flexible and effective fighting forces in the world, and we will sign a new military covenant with our brave heroes.	0.12
Prisoners are encouraged to deal with drug addiction problems during their incarceration, and we do them no favours by not taking a tough line.	0.05

Table A4: *Ten randomly selected UKIP manifesto sentences and their predicted populist scores.*

Country	Party	Election Date	Mean Populism Score
Italy	People of Freedom	2013-02-24	0.05
Italy	Go Italy	2018-03-04	0.05
Italy	Brothers of Italy	2018-03-04	0.06
Italy	Northern League	2013-02-24	0.05
Greece	Independent Greeks	2015-01-25	0.10
Germany	Party of Democratic Socialism	1994-10-16	0.06
Latvia	Zatlers' Reform Party	2011-09-17	0.08
Poland	Self-Defence of the Polish Republic	2001-09-23	0.06
Romania	People's Party - Dan Dianconescu	2012-12-09	0.06
Slovakia	Alliance of the New Citizen	2002-09-20	0.05

Table A5: *Party-elections rated as populist by PopuList (Rooduijn et al. 2019) whose manifestos include no sentences with a predicted populism score above 0.5. Average populism score shown in the second column.*

“Far dimagrire lo Stato, i cittadini danno già troppo”

[translated] Major slimming down of the State: citizens already give too much
“Incremento della lotta per la legalità, per il contrasto ai fenomeni della immigrazione clandestina”

[translated] ”Increase in the fight for legality, to contrast the phenomena of illegal immigration”

“Nuove azioni per favorire la concorrenza nel settore energetico e contrastare gli oligopoli”

[translated] ”New actions to encourage competition in the energy sector and counter oligopolies”

“Razionalizzare la distribuzione territoriale degli istituti e degli insegnamenti universitari”

[translated] ”Rationalize the territorial distribution of institutes and university teaching”

“Eventuali salvataggi bancari devono essere solo a tutela dei risparmiatori e non degli azionisti di controllo”

[translated] ”Any bank bailouts must only protect savers and not controlling shareholders”

“Votare il dimezzamento degli emolumenti dei parlamentari.”

[translated] ”Vote to halve the salaries of parliamentarians.”

“Elezione diretta e popolare del Presidente della Repubblica”

[translated] Direct and popular election of the President of the Republic

“Dimezzamento dei costi della politica. Abolire il finanziamento pubblico dei partiti (nessun fondo pubblico ai partiti)”

[translated] ”Halving of the costs of politics Abolishing public funding of parties (no public funding to parties)”

“Più Europa dei Popoli, meno euro-burocrazia”

[translated] ”More Europe of Peoples, less Euro-bureaucracy”

“La difesa e il sostegno alla famiglia, comunità naturale fondata sul matrimonio tra uomo e donna”

[translated] ”The defense and support of the family, a natural community founded on marriage between a man and a woman”

Table A6: *Hand validation of populism in Italy’s Northern League 2013 manifesto, showing the sentences that appear most populist in a hand coding of the document. The populism classifier identifies no sentences with a predicted populism score above 0.5, while PopuList codes the party-election as populist. While some of the sentences contain slightly populist statements, none appear to be overtly anti-elite and pro-common person.*

	precision	recall	f1-score	support
Non-populist	0.91	0.94	0.92	338
Populist	0.66	0.55	0.60	71
accuracy	0.87	0.87	0.87	0.87
macro avg	0.78	0.75	0.76	409
weighted avg	0.87	0.87	0.87	409

Table A7: *Performance of a classifier trained on synthetic text, evaluated on a set of hand-annotated English language manifesto sentences.*

A.4 Selecting generation hyperparameters for synthetic tweets

I consider 56 combinations of generation hyperparameters and generate 1,000 synthetic tweets for each set.

- epoch $\in \{1, 3\}$
- top_p $\in \{0.8, 0.90, 0.95, 0.99\}$
- temperature $\in \{0.3, 0.5, 0.7, 1, 1.3, 1.5, 1.8\}$
- top_k $\in \{50\}$ (keep fixed)

For each batch of synthetic tweets, I sample an additional 1,000 real tweets and split the corpus into a training set (75%) and an evaluation set (25%). The classifier model is a sentence transformer model, specifically `all-MiniLM-L12-v2` (Reimers and Gurevych 2019), to represent each tweet as a 384-dimensional vector, with a logistic regression classifier on top of the embeddings. This model was chosen because it is within 1 percentage point of the accuracy of the current best transformer model, but is around 20% of the best model’s size.

Figure SI 1 shows the ability of a classifier to distinguish between real tweets and the synthetic tweets generated from each set of hyperparameters. Lower accuracy implies better synthetic tweet quality, as the model has a harder time distinguishing real and synthetic tweets.

A.5 Sample of generated tweets

The first ten tweets generated by a adapted/fine tuned GPT-2 model using the optimal generation hyperparameters obtained from the SVM process described above: top_p=0.90, top_k= 50, temperature= 1.5, training epochs=3.

-
1. [SYNTH] Yes, this is the USS Abraham Lincoln (DDG 71), which was in the Mediterranean a few weeks ago. And yes, you absolutely need to have a GoFund Ukraine account to donate money to charity organizations, including The Red Cross.
 2. [SYNTH] An Osint Bunker article detailing the recent Russian buildup near Kyiv
You can click on the link to be taken to the actual map
 3. [SYNTH] Russian military is getting massed in Belarus, most likely to set up new staging areas.
 4. [SYNTH] The system is relatively good at engaging low/medium armored targets, like BTRs, MT-LBs, APCs and SPGs
 5. [SYNTH] I think people got the wrong impression from today’s press conference, where Lukashenko said “I do not fear Western military threats but Russia is prepared to pay a heavy price for any military action.
 6. [SYNTH] And as usual people are falling for the bait and trying to equate this to some sort of new high crime, especially when we see examples of this already in the news.
We already saw it yesterday with @Nrg8000
 7. [SYNTH] This is mostly because air defence is weak, and even non TB2s could get shot down. Only a very few aircraft flew today, with the majority of them from the western part of Ukraine. In the north of Ukraine a lack of TB2s has caused large losses. The Ukrainians are probably using the drones to spot artillery strikes.
 8. [SYNTH] Tanks on the other side of the Irpin River
 9. [SYNTH] Russian forces pushed back from Kharkiv tonight
Kherson Oblast
 10. [SYNTH] It doesn’t even have infrared sensors - only a SINCARS system.
This basically tells you what its main purpose is.

Table SI 1: *Synthetically generated tweets from a GPT-2 model adapted/fine-tuned on 20,000 tweets reporting open source intelligence on the war in Ukraine. Due to Twitter’s restrictions on including actual tweets in published work, no comparison is provided for real tweets. GPT-2 generation parameters: top_p= 0.90, top_k= 50, temperature= 1.5, fine tuning epochs=3*

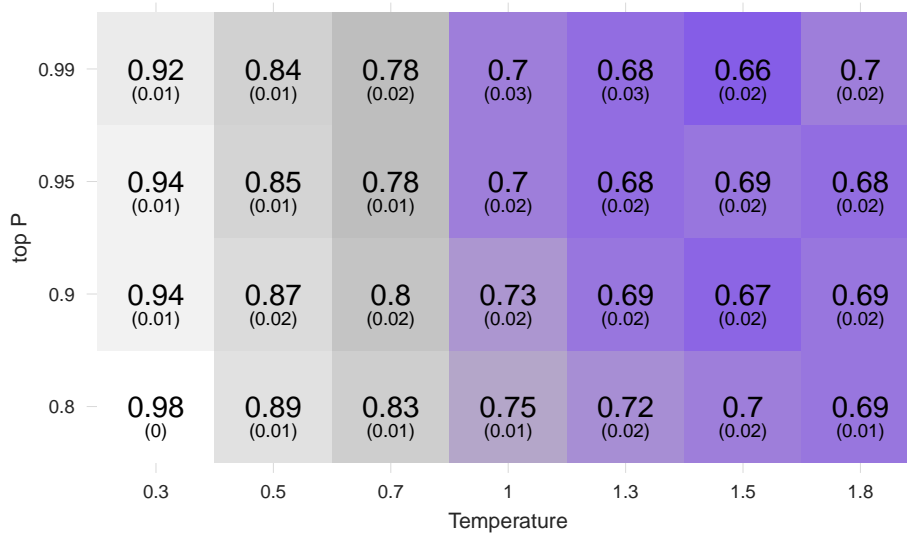


Figure SI 1: *Effect of GPT-2 generation hyperparameters on synthetic tweet quality. Accuracy is the mean out-of-sample accuracy of an embedding+logistic regression model to discriminate between real and synthetic tweets. Lower accuracy implies better synthetic tweet quality, as the model has a harder time distinguishing real and synthetic tweets. Numbers in parentheses report standard deviation across 10 runs (varying the sampled real tweets and the train/test split).*

A.5.1 GPT-3.5 and GPT-4

GPT-3.5 and GPT-4 both perform much better than GPT-2 in most tasks. However, their performance in generating tweets related to the post-2022 phase of the Russia-Ukraine war is much worse than GPT-2 than the fine-tuned GPT-2 discussed above. This is the result of the limitations of the training data of both GPT-3.5 and GPT-4, which do not include the post-2022 phase of the war.

I prompt GPT-3.5 with the prompt below, providing context about the war in Ukraine and providing a specific and well-known open source intelligence analyst who covers the war. Note that most of the tweets generated by GPT-3.5 discuss the pre-2022 phase of the conflict.

You're @markito0171, an OSINT analyst who covers the war in Ukraine.

Write 5 of their tweets reporting details on the war.

1. [SYNTH] “ BREAKING: Intense clashes reported in eastern Ukraine near the town of Mariupol. Local sources suggest heavy artillery fire and ground skirmishes. Monitoring the situation closely for updates. #Ukraine #WarUpdates #OSINT”
2. [SYNTH] “ UPDATE: Satellite imagery reveals a significant build-up of Russian military equipment near the border with Ukraine. The deployment includes tanks, artillery, and air defense systems. Tensions rise as the potential for further escalation increases. #Russia #Ukraine #OSINT”
3. [SYNTH] “ ANALYSIS: Recent drone footage confirms ongoing construction of trenches and fortifications by Ukrainian armed forces in the town of Avdiivka. These defensive measures indicate an anticipation of further hostilities in the region. #Ukraine #WarAnalysis #OSINT”

Repeating the exercise with GPT-4 (in July 2023) provides more fluent tweets, but are still limited by the lack of training data on the post-2022 phase of the war.

You’re @markito0171, an OSINT analyst who covers the war in Ukraine.

Write 5 of their tweets reporting details on the war.

1. [SYNTH] “#ConflictUpdate: Reports of increased shelling near Mariupol, indicating potential escalation by separatist forces. Residents are urged to stay vigilant and take necessary precautions. #UkraineConflict #OSINT”
2. [SYNTH] “#WarInUkraine: Recent satellite images reveal a build-up of heavy weaponry in Donetsk region. This suggests a significant escalation in the conflict. Please stay safe. #OSINT #Ukraine”
3. [SYNTH] “Recent OSINT evidence suggests a surge in cyber-attacks against Ukrainian govt websites. These acts of digital warfare imply an intensification of the conflict. Everyone, please keep your online data secure. #CyberWarfare #UkraineConflict”