

Synthetically generated text for supervised text analysis

Andrew Halterman (Michigan State University)

September 15, 2023

Abstract

Large language models are a powerful tool for conducting text analysis in political science, but face transparency, reproducibility, and explainability issues. This article proposes using LLMs to generate synthetic text, which can then be used to train a supervised text model. Using synthetic text addresses several obstacles to supervised text analysis, including the expense of labeling documents, the difficulty of retrieving rare documents for annotation, and the obstacles to sharing documents. I provide a conceptual overview of text generation, guidance on when researchers should prefer different techniques for generating text, a discussion of ethics, and a simple technique for improving the quality of synthetic text. I demonstrate the usefulness of synthetic text with three applications: generating synthetic tweets describing the fighting in Ukraine, synthetic news articles describing specified political events for training an event detection system, and a multilingual corpus of populist manifesto statements for training a sentence-level populism classifier.

Word count: 9,822 (excluding title page, references, and supplemental information)

1 Introduction

Researchers in political science are rapidly adopting large language models (LLMs) to assist in the analysis of text. Large language models, such as GPT-2, GPT-3, and many others, show great promise in document classification and information extraction Ziems et al. (2023), tasks that researchers have traditionally done by hand-labeling documents and training supervised learning models. A common use of LLMs for document classification is for the researcher 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 this technique is simple to implement and performs well on several studies, it imposes costs, including problems of reproducibility (Spirling 2023), cost, interpretability and accuracy. The current state-of-the-art LLMs are closed source commercial offerings, which can be deprecated with little warning and can change several times per year (Kapoor and Narayanan 2023), making it impossible to reproduce their output in the future (Spirling 2023). While using LLMs to classify documents is cheaper than hiring human annotators (Ornstein, Blasingame, and Truscott 2022), the cost of running large models, either commercial or self-hosted, can be considerable. Moreover, even state-of-the-art LLMs fail on specific tasks (Peskov and Stewart 2023). Finally, the process by which the LLM provides a document label is inscrutable, particularly in the case of closed-source commercial LLMs.

This article suggests an alternative use of LLMs for supervised text analysis tasks: to generate synthetic text, which researchers then (optionally) label and use to train a supervised text model. Older large language models can already generate news text or political text that is indistinguishable from human-produced text (Zellers et al. 2019; Kreps, McCain, and Brundage 2022) and rapid improvements in LLMs

will continue this trend. 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. They can then annotate the text with their desired labels and train a classifier on the labeled synthetic text, or, in some cases, directly train a classifier on the synthetic text. This approach 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 models, greatly improving reproducibility and lowering costs. Furthermore, because the LLM produces synthetic text, as opposed to labels, 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 more interpretable models than they can the output of an opaque LLM-based classifier.

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, introduces a technique for measuring and improving the quality of synthetic text, and discusses the serious ethical pitfalls inherent in using synthetic text. It presents three short applications from political science, illustrating how the approach can create synthetic tweets describing the war in Ukraine for a named entity recognition model, synthetic news articles for training an event data detection model, and a multilingual sentence-level populism classifier with no hand-labeled examples. It demonstrates that synthetic data can be used to create tweets that are difficult to distinguish from real tweets and that synthetic documents can be used to train text classification models with no human annotation at all. In general, a marginal labeled synthetic document does not improve model performance as much as a marginal real document, meaning that researchers face a tradeoff between accuracy and the benefits

produced by using synthetic text. However, the tradeoff can be reduced by using a simple procedure I introduce for improving the quality of synthetically generated text.

1.1 Obstacles to using supervised text analysis

Researchers in political science use supervised text analysis for a range of tasks, including document classification, sentiment analysis, extracting event information, or identifying named entities. Researchers creating any kind of supervised text analysis model face several obstacles. The primary cost in developing a supervised model is *labeling*, or annotating text to serve as training data. Human annotations on documents, such as document labels for classification or labeled spans of text for information extraction, are expensive to collect. Researchers must define their concept of interest, create a codebook, train annotators, pay them, and conduct quality assurance on the labels they provide. Faced with the costs of obtaining labels, researchers may turn to unsupervised techniques or avoid text analysis altogether.

A second obstacle is *retrieval*, namely that annotators need to be provided a set of relevant documents to annotate. Because many of the concepts of interest in political science are rare classes (Miller, Linder, and Mebane 2020), a simple random sample from a corpus will often not retrieve enough relevant documents to annotate. For instance, if a researcher is training a classifier to recognize police violence in news text and it is only described in 0.5% of news stories, a researcher annotating a random sample of news stories would need to annotate 20,000 randomly sampled stories to expect to obtain 100 documents describing police violence. The existing approaches to addressing the rare class issue involve over-sampling relevant documents using keywords (e.g. Mueller and Rauh 2017), using active learning techniques (Miller, Linder, and Mebane 2020), or by exhaustively annotating an entire corpus (Halterman et al. 2021). Each of these techniques carries drawbacks in annotation cost, low recall,

or dependence on a model to suggest documents to label (Halterman et al. 2021).

Finally, researchers face *copyright* restrictions, license agreements, or privacy concerns that limit their ability to share annotated documents with annotators or other researchers. Most news articles are copyrighted, while other text, such as social media posts, carry extra terms of service requirements that they not be shared.¹ The inability to publish annotated training data makes it difficult or impossible to build on existing datasets or replicate existing methodological work.

Generating synthetic training data in a controllable way, that is, with the ability to direct the content and style of the text, can partially address these issues. Researchers can generate synthetic training text that contains the (potentially rare) concepts of interest, augmenting their potentially limited set of real validation text and addressing the rare class retrieval problem. This training text can be annotated for document classes, sentiment, or spans of information. In some cases, the synthetic training text can be used without hand labeling to train a document classifier, addressing the labeling issue. The synthetic training data they generate is not subject to copyright restrictions, allowing them to share their training data with other researchers, improving reproducibility. This approach to using LLMs allows researchers to use them for their strengths—generating text—while avoiding their weaknesses—opaque classification, high computational costs, and a lack of reproducibility.

¹Even if researchers have the legal right to share text, ethics and privacy concerns can preclude sharing documents such as personal narratives, free-form survey responses, social media posts containing personal information or discussions of sensitive subjects with annotators or other researchers.

2 Using language models to generate synthetic text

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.² The parameters that control the predicted probability of the next token in a sequence are learned empirically from a large collection of training text. As an example, a language model might learn that, given the input sequence “the capital of Germany is_____”, the token that has the highest probability of coming next is “Berlin”.

2.1 Formalizing text generation

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: $p(W) = \prod_{i=1}^n p(w_i | w_{i-1}, \dots, w_2, w_1)$.³ We can approximate the conditional probability of the next token given the previous tokens and trainable parameters θ :

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

To build intuition, we can consider a Markov chain, one of the simplest techniques for generating a sequence of tokens. A Markov chain limits the number of tokens used

²For ease of explication, this paper focuses on the task of predicting the next token that follows a sequence of tokens, which is often referred to the computer science literature as a “causal” language model. Other language models, such as those in the BERT family (Jacob Devlin et al. 2018), are bidirectional, meaning they predict the token that fills a gap in the middle of a sequence of tokens, conditioning on tokens that appear on either side of the missing word. This “masked language modeling” approach produces general purpose models that are well suited to a wide range of tasks, but generally do not perform as well on next word prediction models.

³A bidirectional model such as BERT predicts w_i in an n length sequence using tokens on either side: $p(w_i | w_1, w_2, \dots, w_{i-1}, w_{i+1}, \dots, w_n)$.

in predicting the following token $p(w_i|w_{i-1}, w_{i-2}, \dots, w_{k < n})$ and uses the raw empirical frequency of each sequence in the training data as its parameters θ . The advantages of newer language models are that they allow longer sequences of text to inform their prediction of the next token and their use of contextual word embeddings allow for more efficient representations of words.

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 :

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

The generation parameters γ control how words are sampled from the probability distribution over the next word. For instance, γ might specify a “greedy” sampling strategy where the highest probability word is always drawn, that words are sampled in proportion to their predicted probability, or a more complicated process that jointly generates several following tokens at once. While θ , the parameters that govern the probability distribution over words, are learned during training, γ can be varied later to change how tokens are drawn. A later section of the paper provides a technique for researchers to select optimal values of γ to produce more realistic text.

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 $\hat{p}(w_i)$ given the previous tokens, 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 on future language models as well.⁴

The *adaptation* approach updates the weights θ of a pretrained model to affect the content or style of generated text.^[^fine_tuning_note] Researchers can download generic pre-trained language models that have been trained on a diverse set of text, including Wikipedia and unpublished novels (J. Devlin et al. 2019), outbound links from Reddit (Radford et al. 2019), or academic articles, crawled web pages, code repositories, movie subtitles, and internet forums (e.g., Gao et al. 2020). Off-the-shelf pretrained models may not reflect a researcher’s desired style or content, however, 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, changing $p(\hat{w}_i)$ as a result. The 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”. Here, 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.

When a researchers use adaptation, they can employ a simple procedure introduced below to maximize the similarity between the real and synthetic text. The adaptation approach is used in the paper’s first application to generate synthetic tweets reporting battlefield updates from the war in Ukraine.

Second, rather than updating the weights of a model, a research can instead use *prompting* to guide synthetic text generation. If a researcher can provide the beginning of a document $(w_{i-1}, \dots w_n)$, a large language model can generate a plausible continuation of the document. For example, a researcher can generate a news story describing a protest by hand-writing a headline related to protests. Researchers can

⁴Note that ChatGPT restricts users to prompting alone. This makes it simple to use but limits its flexibility.

also use prompting to augment an small set of real documents by using real headlines as a prompt to generate additional news stories, augmenting their potentially limited set of real stories. A prompting approach is illustrated in the second application to generate news stories describing armed conflict or violence by providing manually written headlines to elicit stories with “assault”-type events.

Recent off-the-shelf language models can generate text from abstract prompts that describe the desired output, rather than simply continuing from the starting tokens of some desired text (P. Liu et al. 2021). For instance, GPT-3 can be provided with a prompt such as “write a press release in the style of a Republican House member” and obtain a plausible output without the need for a specific prompt or adapting a language model on a corpus of Congressional Republican press releases. 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 third 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 alongside both adaptation or prompting to produce more realistic text. This approach is described more fully below and is used in the first application to generate more realistic tweets.

When should researchers use adaptation and when should they prompt? Table 1 provides an overview of how each technique addresses the three challenges. Adaptation can provide them with a version of their dataset that they can share freely, or can expand a small set of labeled documents into a large corpus. Adaptation also

avoids the need for prompts—it is well known that language model output is sensitive to changes in prompts (Zhao et al. 2021), potentially producing text that does not match the context or style a researcher wants. Adaptation may be necessary when a researcher’s text is outside the domain of the training data of the pretrained model. When adapting, researchers can also quantitatively evaluate the quality of their synthetic text by comparing it with the existing reference corpus. Adaptation is usually more technically challenging, requiring researchers to write code to update the pretrained model and often requires greater computing resources, including access to a GPU.

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 (e.g. the definition of populism used in the third application). Finally, prompts are easy to write and tweak to produce synthetic text that matches the desired context (for example, in the second application, to ensure that all aspects of the desired political violence event class are included in the training corpus.)

The prompting approach is simple, but will fail if either the domain is not covered well in the language model’s training data (for example, events that occur after the end point of the model’s training data) or if the desired text is difficult to prompt (for example, tweets lack the convenient structure of news stories, where a headline usefully summarizes the content of the story). Knowing if a domain is outside the model’s training data is sometimes clear (e.g., the Russia-Ukraine falls outside GPT-2’s temporal coverage), but is increasingly difficult as training data becomes more opaque (Chang et al. 2023). The model’s ability to generate high quality text can be established by the researcher carefully reading a selection of generated text (Grimmer and

	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. 1)
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 armed conflict (App. 2) Populist sentences (App. 3)
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"> Requires existing corpus for adversarial method If no corpus is available, requires manual tuning 	<ul style="list-style-type: none"> Ukraine war tweets (App. 1)

Table 1: Overview of the three approaches to controlling synthetic text generation with benefits and limitations of each.

Stewart 2013). Some work in computer science discusses automated out-of-domain detection (e.g., Xu et al. 2021; Lang et al. 2023), but generally relies on access to the original corpus of training text.

2.3 Improving synthetic text quality with an adversarial classifier

Language models have a set of parameters γ that affect how a word is sampled from $\hat{p}(w)$. For GPT-2 and GPT-3, 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).⁵ Varying these hyperparameters greatly affects the quality of the generated text, but little theoretical guidance exists on how to select generation hyperparameters (Fu et al. 2021).⁶

I introduce a simple “adversarial” procedure for selecting the best generation hyperparameters for generating text, drawing on the intuition that the harder it is for a classifier to 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. The technique is adversarial in the sense that the worse the classifier performs at the task of discriminating between real and synthetic text, the more realistic the synthetic text is. The procedure uses the same concept behind generative adversarial networks (Goodfellow et al. 2014), which are a machine learning technique that use a “generator” model to produce data and a “discriminator” model to distinguish between real and generated data. The worse the discriminator performs, the better the generator is at producing realistic data. However, the approach described here is simpler in that it does not update the generator and discriminator models iteratively, instead it evaluates the quality of a single batch of synthetic text using a single classifier.

When using the technique, for each set of hyperparameters, a researcher generates n synthetic documents 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. Evaluating

⁵The full details of GPT’s generation parameters are beyond the scope of this paper. See Platen (2020).

⁶Recall from Eq. 2 that the generation hyperparameters γ control the generation of text from an existing, trained language model. A separate set of parameters θ are learned during training and affect the predicted probability of a following token (see Eq. 1). Because large language models are expensive and time consuming to train from scratch, a discussion of the *training* hyperparameters that affect how language models are trained is beyond the scope of this article. For more details on language model training see Dodge et al. (2020).

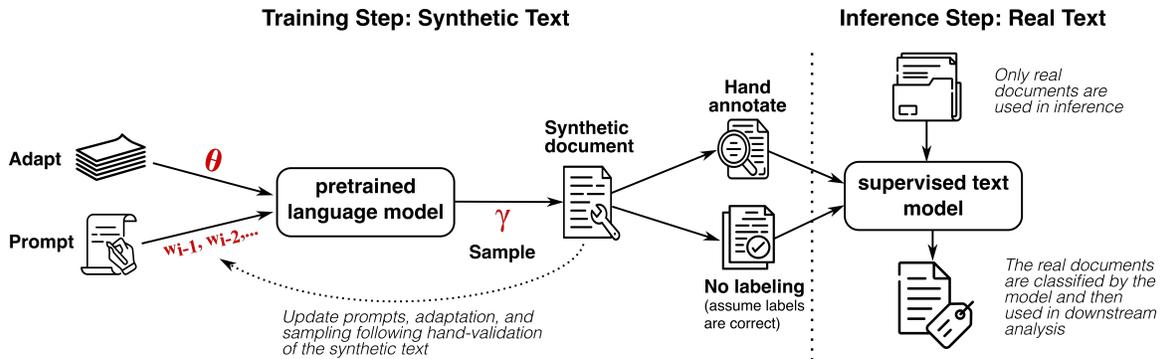


Figure 1: Overview of the synthetic text process for supervised learning. Researchers can affect the content and style of synthetic documents by changing language model parameters (θ), by providing new prompts (w_{i-1}, w_{i-2}, \dots), or by changing the sampling parameters (γ). Researchers then decide whether to hand-label the resulting synthetic documents or to use them as-is to train a model without any hand-labeled data. After the supervised model is trained, synthetic text is discarded and only real text is used for analysis.

the classifier’s performance on a test set provides a quantitative measure of the synthetic text’s quality. Decreasing model performance indicates that real and synthetic documents are increasingly difficult to distinguish. The set of hyperparameters that results in the lowest classification accuracy is thus the set that generates the best documents. The researcher can then use this batch of text for training a supervised learning model or for hand-labeling.

Alternatively, a researcher could assess the synthetic document quality by fitting a structural topic model (Roberts et al. 2013) with the document’s real/synthetic label as a covariate and examining the difference in topics. This would allow a researcher to check whether the synthetic text covers the entire domain of real text or if certain topics are differentially represented between the corpora.

2.4 End-to-end synthetic text pipeline

Figure 1 shows a high-level overview of the entire process. Researchers can control the content and style of text by adapting, prompting, or changing the generation parameters of a language model. The adversarial technique helps researchers select

the best generation parameters if they have access to an existing corpus of real text. In any case, they should also conduct hand-validation of the synthetic text and adjust their guidance strategy to improve the quality of the synthetic text.

After they generate satisfactory synthetic text, researchers then have two options for how to use it to train a supervised text model. First, they can treat the text as unlabeled and collect annotations on it in the same way they would with real text. Annotations are required if, for instance, they are training an information extraction model, rather than a document classifier. By using controlled synthetic text, they have addressed the copyright or usage restrictions of real text and can share their annotated text freely, and have hopefully addressed the retrieval problem as well. This hand-annotation approach is used in the first and second applications below. A second option exists when researchers are conducting document classification and believe that their prompting strategy reliably generates documents with the desired class label. In this situation, researchers can train their document classifier directly on the synthetic text without hand labeling, assuming that their guidance strategy reliably produces documents from the desired class. This approach is used in the third application below. After the model is trained on synthetic text, the researcher then sets aside the synthetic training data and only uses real text to conduct their final analysis.

Finally, a researcher might wonder why they need to generate synthetic text at all if the prompted or adapted LLM encodes the desired information. Could they not just use the LLM's encoded information directly, for instance, by sampling words from the LLM's hidden state? While appealing, this approach has several drawbacks. First, it is difficult in practice to extract token probabilities from an LLM because the LLM's output is a sequence of tokens, where each token's probability is conditioned on the previous tokens. If a researcher is prompting the LLM, the token probabilities

are also conditioned on the prompt, and a single prompt may not be sufficient to accurately capture the entire distribution of token probabilities. In cases where the task is more complicated, such as extracting information from within documents, the LLM’s per-token probabilities will not be sufficient to do the task. The proliferation of open source “instruction-tuned” LLMs, which are trained to perform a set of tasks including document classification or information extraction, may make the approach of directly using the prompted or adapted LLM more feasible. However, the approach of generating synthetic text and then training a supervised model on the synthetic text has advantages in the size of the LLM that is required, the transparency of the approach, and the ability of researchers to annotate the synthetic text according to their specific needs, rather than relying on a pre-trained LLM’s output.

2.5 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.⁷

The disclaimer should only be removed temporarily as a final step before fitting a model to avoid the possibility of synthetic data being mistaken for real text. Annotators should be briefed on the use of synthetic text and the annotation interface should clearly state that they are working with synthetic text, which likely contains factual errors. Any synthetic text reported in published work must be clearly marked (e.g. [SYNTH]) and the accompanying text clearly explain its use and potential bias.

While synthetic text may be difficult to distinguish from real text in its style or writing

⁷For example, `<!--SYNTHETIC TEXT! Do not trust the factual content of this text. Generated by <author>, <email> to train a populist speech classifier. -->`

and thus useful for training a model to recognize certain linguistic features, its factual content will be imaginary and thus completely unsuitable for answering substantive questions on its own.

Third, it is well known that 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. Curating the text using to adapt the model or using prompts that break the association between groups and stereotyped traits offers a partial solution, but greater research into the prevalence and mitigation of these harms is needed.

Finally, 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. 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.

2.6 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, where synthetic data is used to supplement a small amount of real data. In computer vision, a standard practice is to augment a small set of real images by applying transformations such as cropping, rotation, or color changes (Shorten and Khoshgoftaar 2019), and recent work has used fully synthetic face images to train face recognition models (Wood et al. 2021). 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 has used language models to generate fully synthetic text. For

example, Anaby-Tavor et al. (2020) fine tune GPT on a sample of text and use a classifier trained on the original data to eliminate synthetic text with low confidence labels before fitting a model. Others have generated synthetic text by prompting, assume the labels are correct, and fit a zero shot model Ye et al. (2022). He et al. (2022) discuss when to use unlabeled synthetic text and when to annotate the text.

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. The first application, on generating synthetic tweets describing the war in Ukraine, illustrates an adaptation approach and introduces a simple method for improving the quality of the generated text. The second and third applications use two approaches to prompting to generate synthetic news stories describing political violence and to create data for training a sentence-level populism classifier. The third application also highlights limitations in studying populism using manifestos.

3.1 Adapting Language Models for Copyright-Free Tweets: Identifying Weapons in the Ukraine War

Social media posts are a major source of text for political scientists but platform terms of use and privacy concerns greatly limit researchers' ability to share or publish the posts they have annotated. Researchers have a legal and ethical obligation to protect the privacy of social media users and delete posts on request, which is an inherent limitation of social media data and can make substantive replication difficult. A more narrow issue arises in the situation where researchers collect annotations on posts to introduce new methods for analyzing social media text. Validating these methodological improvements require access to the annotated training data to replicate. If

researchers collect these annotations on synthetic text, as opposed to posts by real users, they will be able to provide transparency and reproducibility while protecting the privacy of social media users.

As political scientists collect more data about the ongoing war in Ukraine (Zhukov 2022), supervised models trained on tweets will provide an important source of information.⁸ This application shows that synthetic tweets can be generated that are difficult to distinguish from real tweets and can thus serve as useful training data. A named entity recognition system to detect mentions of weapons trained on annotated synthetic tweets reaches the same accuracy as one trained on real annotated tweets, but requires approximately 50% more annotations to reach the same performance.

I collect a set of around 20,000 real tweets from four Twitter accounts that report detailed information on the fighting in Ukraine.⁹ Because the synthetic tweets should closely match the actual tweets and because tweets are more difficult to prompt than news articles, which have a convenient headline–body structure, I opt for an adaptation approach to text generation. I adapt a large language model on this set of tweets to produce a language model that is well suited to generating tweets about the war (Wolf et al. 2020). Specifically, I use GPT-2 because it can be run and adapted locally with standard hardware, and performs well on the task of generating tweets. Adaptation is especially valuable for this application because the original training data for GPT-2 (and GPT-3.5/ChatGPT) was collected before the war, making it impossible to produce accurate synthetic tweets about the war without adapting the model on tweets written after February 2022.¹⁰ By adapting the model, I can both ensure that the generated text is similar to real tweets about the war in Ukraine, as well as eliminating the need to provide a specific prompt to generate text.

⁸Concerns about the reproducibility of Twitter-based methodological work are only heightened after recent changes to Twitter’s API.

⁹Specifically, @uaweapons, @osinttechnical, @oryxspioenkop, and @markito0171.

¹⁰See SI section A.1.2 for example synthetic tweets produced by GPT-3.5 and GPT-4.

[SYNTH] The system is relatively good at engaging low/medium armored targets, like **BTRs**, **MT-LBs**, **APCs** and **SPGs**

[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.

[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.

[SYNTH] Tanks on the other side of the Irpin River

Table 2: *Selected synthetically generated tweets from a random sample of 10 generated from a GPT-2 model adapted/fine tuned on 20,000 tweets reporting open source intelligence on the war in Ukraine. Weapon annotations shown in bold. See the SI for the full list of 10 randomly selected tweets. Due to Twitter’s restrictions on including tweets in published work, no actual tweets are shown. GPT-2 generation parameters (γ): top_p= 0.90, top_k= 50, temperature= 1.5*

To improve the quality of the synthetic tweets, I apply the adversarial method introduced above. Across 56 combinations of hyperparameters, the classification accuracy of an SVM bag-of-words classifier ranges from 0.98 to a low of 0.64 and a BERT-based classifier ranges from 0.99 to 0.67, indicating a large effect of hyperparameters of tweet generation, but similar performance between adversary models.¹¹ 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, using synthetic tweets to train a supervised model will require more annotations than using real tweets. Table 2 reports a random sample of synthetic tweets using the best hyperparameters (that is, the ones producing the lowest accuracy for the discriminator model).

3.1.1 Comparing performance of real and synthetic tweets

How well do these synthetic tweets work in practice to train a supervised text model? I hand annotate 1,600 tweets with span-level labels on the specific weapons systems

¹¹See SI Figure SI 1.

described in the tweets. Existing NER systems do not identify weapons in text, meaning that researchers who wish to identify weapons automatically must train their own NER model. 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.¹² Figure 2 reports the accuracy for the model trained on actual and synthetic tweets at different training set sizes and evaluated on labeled actual tweets.¹³ The figure shows that the performance of the NER model improves as it has access to more labeled training examples across actual tweets, synthetic tweets generated without any generation parameter tuning, and synthetic tweets generated with optimal parameters. 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 actual 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 actual tweet model, rather than 500 without optimizing.

While using 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

¹²I use spaCy 3.1.2’s small `en_core_web_sm` model as a base and the default training values set by Prodigy (Honnibal and Montani 2017; Montani and Honnibal 2018). Better absolute performance could be achieved with a larger model, but I expect the relative performance to be the same.

¹³More specifically, I use span-based F1 score as an accuracy measure, where precision is the proportion of identified named entities that are correct, recall is the proportion of named entities identified by the model, and F1 is the harmonic mean of the two: A per-token F1 score would be higher, but the relevant metric here is the relative performance from the two data sources, not the absolute performance of the NER model.

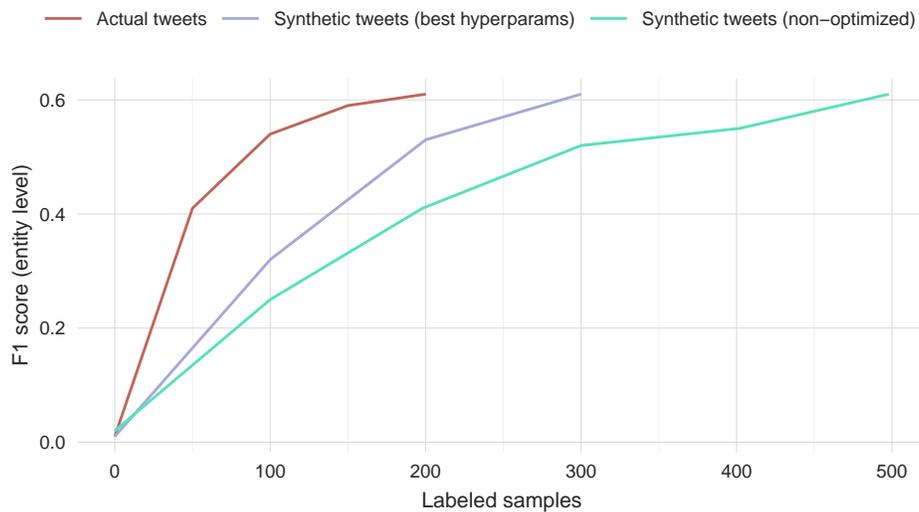


Figure 2: *Test set performance of a named entity recognition model detecting a WEAPON class, trained on annotated actual tweets and annotated synthetic tweets. A model trained on annotated synthetic tweets generated from the optimal hyperparameters found using the adversarial technique requires around 300 examples to reach the performance of a model trained on 200 annotated actual tweets. In contrast, when using non-optimized synthetic tweets, 500 labeled synthetic examples are required to reach the performance of 200 actual tweets.*

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 substantive, claims can be reproduced.

3.2 Application 2: Generating rare documents for human labeling—training an event data classifier

Event data is a major source of quantitative information for researchers in international relations and comparative politics (Beieler et al. 2016). Automated event data systems increasingly rely on machine learning models to identify events, but despite decades of investment, no corpus of news text with comprehensive, hand-validated political event labels exists.¹⁴ The lack of a gold-standard set of labeled news articles makes it difficult to evaluate event classification models or to develop new classifiers as machine learning techniques improve. This application shows how the three obstacles to supervised text analysis can be overcome with *prompting*. The adaptation approach used in the Ukraine war tweets example is not suited to this application, because adaptation requires an existing set of articles with a known event type, which we do not have. News text is well-represented in the training corpora for many language models, making them suitable to generating synthetic news articles.

A simple way to generate synthetic news articles with desired content is write a headline that reflects the event type or concept we would like to have a story about. For

¹⁴Existing large event datasets, such as GDELT (Leetaru and Schrodt 2013), ICEWS (Boschee et al. 2015), or POLECAT (Halterman et al. 2023) are machine generated and too noisy to use as training or validation data, and do not make the underlying text available. Some annotated datasets with limited coverage exist, e.g. annotation on actions taken by police in India (Halterman et al. 2021), but are limited in scope. Event datasets in computer science/computational linguistics such as ACE (Doddington et al. 2004) have event definitions that are largely not relevant for political science. Other hand-annotated datasets in political science, such as ACLED (Raleigh et al. 2010) or GED (Sundberg and Melander 2013), do not provide the raw source text, and, in the case of ACLED, include restrictive terms of service that prohibit their use here.

example, to generate stories about disinformation or information operations, we can write the headline “Foreign ‘information operation’ spreading disinformation uncovered”. We can provide this headline to a large language model, in this case GPT-2 (large) (Radford et al. 2019) and generate a synthetic news story prompted by the title.¹⁵ We can increase the diversity of the training text by modifying the byline in the prompt to refer to different news sources and cities.

[PROMPT]

Foreign ‘information operation’ spreading disinformation uncovered

BELGRADE (Reuters)

[- SYNTHETIC STORY -] In an unusual development this week, Serbian President Tomislav Nikolic called the foreign media a threat to the nation’s security.

“Foreign ‘information operations’ against us and our country (are) the ones spreading the disinformation against us,” Nikolic said in [...]

[- SYNTHETIC STORY -]

If we keep the same headline but change the byline to “BRUSSELS (local sources)”, we can generate a story about misinformation occurring in a completely different context (although note that the story itself is misinformation):

[PROMPT]

Foreign ‘information operation’ spreading disinformation uncovered

BRUSSELS (local sources)

¹⁵ Later models such as GPT-3.5 or GPT-4 could potentially generate higher quality news text. However, as discussed in the introduction, these models are not available locally, raising replicability issues (Spirling 2023). More advanced open source/local models could also provide potentially higher quality text as well.

[– SYNTHETIC STORY –]

A report by Belgian daily La Presse has shown that the United States is planning on deploying military ‘information operations’ in Europe, which could be used for ‘hybrid war’ scenarios. The report said that the Pentagon is sending US forces to Western Europe to create ‘fake news’ and create a ‘vast digital surveillance network’ on political parties, activists, media commentators and ‘subversives’ with [...]

[– SYNTHETIC STORY –]

Halterman et al. (2023) introduce a new dataset of hand annotations on 12,952 news stories for 16 defined event classes, including an **ASSAULT** event class that includes a broad range of political violence and armed conflict, which is used as a training and evaluation set for this application.¹⁶ Figure 3 compares the performance of classifiers trained on three sets of data: human-annotated actual news stories, human-annotated synthetic stories, and unlabeled synthetic stories. In the last case, synthetic documents generated with headlines meant to prompt **ASSAULT** are assumed to indeed include an **ASSAULT** event, and stories generated with non-**ASSAULT** headlines are assumed to not contain **ASSAULT** events.¹⁷

Figure 3 shows that a marginal labeled real document improves out-of-sample classifier performance more than an additional labeled synthetic document, indicating that researchers face a tradeoff between accuracy and the retrieval/copyright benefits of synthetic text.¹⁸ The result also show, however, that a model trained exclusively on unannotated synthetic documents performs at least as well as one that uses annotated

¹⁶According to the dataset’s event ontology, “**ASSAULT** events are deliberate actions which can potentially result in substantial physical harm”, including military assaults, attacks, kidnappings, terrorist attacks, ethnic cleansing, torture, beatings, etc.

¹⁷See SI A.2 for details on the headlines used and the total number of synthetic stories.

¹⁸I use a sentence transformer model, specifically specifically `all-MiniLM-L12-v2` (Reimers and Gurevych 2019), to represent each news story as a 384-dimensional vector, with a logistic regression classifier on top of the embeddings.

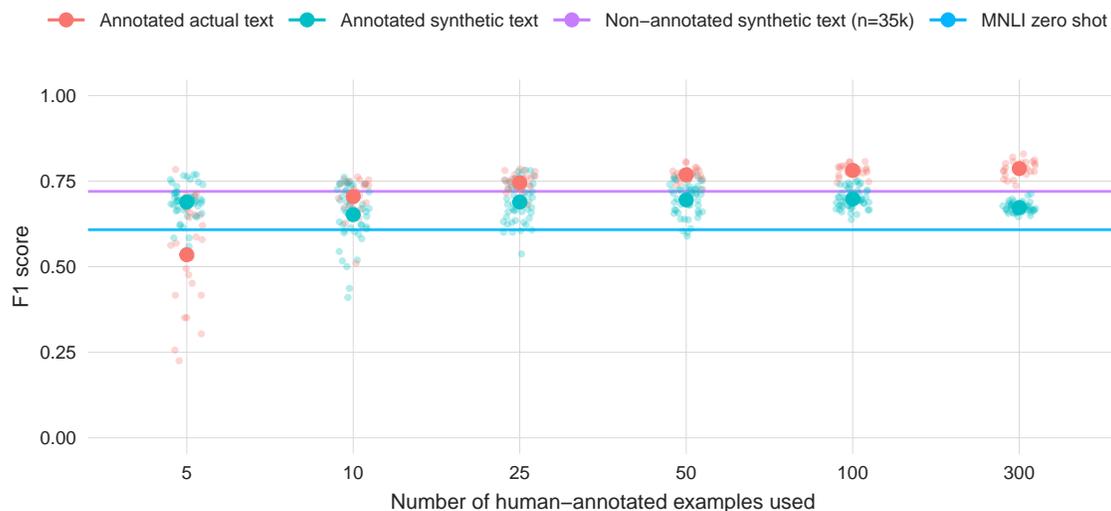


Figure 3: Performance of a transformer-based classifier predicting the ASSAULT class using three sets of training data: annotated real news articles, annotated synthetic articles, and non-annotated synthetic articles (no human labels). A zero-shot model (*roberta-large-mnli*) is included as a baseline. Using hand-labeled synthetic documents incurs an accuracy cost compared to hand-labeled actual documents, but the accuracy across different train/test splits overlaps until 300 labeled examples. Using 35,000 non-annotated synthetic articles (i.e., assuming that every article includes the desired event type) performs at least as well as labeled synthetic stories (horizontal line), suggesting that no hand labeling is required if that accuracy is sufficient. Smaller points indicate 25 random train/test splits and initializations, lines show mean performance.

synthetic documents. Because synthetic documents are essentially free to produce, a large number can be generated and used to train a classifier if the articles' content matches the desired event type prompted by the headline. The blue line in Figure 3 shows the performance of a zero-shot article classifier, which is not trained on any labeled data.¹⁹ The zero-shot model performs worse than any model trained on more than 10 labeled examples and worse than the model trained on 35,000 documents without hand-validated labels. The trained model can then be applied to real text to measure assault events.

Thus, synthetic text can partially address all three obstacles to building a supervised text classifier for identifying political events in text. It addresses the *retrieval* problem that researchers face in collecting annotations on rare classes. Even if a text generation process is not completely accurate in generating articles with the desired class, it still produces a corpus with a much greater concentration of relevant documents than a random sample from the corpus would generate. In this case, it also partially solves the *labeling* problem. If a researcher is willing to accept a classifier with somewhat worse performance, they can forgo an annotation process entirely and use only the labels from their prompts. Regardless of if they hand annotate or take the synthetic labels as-is, they will now have a labeled dataset that they can freely share, improving reproducibility and allowing other researchers to develop improved classifiers with their data.

3.3 Application 3: 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

¹⁹Further details are provided in SI section A.3.

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 document. Given that no dataset exists that labels populist speech at the sentence level, recent work has proposed training sentence-level supervised classifiers to identify populism using manifesto-level labels (Di Cocco and Monechi 2021). This approach has been criticized, for, among other things, for relying on document-level labels to train a sentence-level classifier when most sentences in a populist party’s manifesto will not be recognizably populist (Jankowski and Huber 2023). This application illustrates a new method for identifying populism at the sentence level across 27 European countries in 22 languages. I use a *prompting* approach to generate synthetic populist manifesto statements, and then train a classifier on the synthetic data to identify populist statements in real manifesto text. I find evidence that the classifier reliably identifies populist rhetoric in text and also identify a major limitation in studying populism using manifesto text. Several populist parties have virtually no measured populist rhetoric in their manifestos, which is borne out through careful reading of the manifestos. This finding has implications for the study of populism and highlights the inherent limitations of using manifesto text to assess parties’ populism.

I employ a prompting approach and GPT-3 to generate populist manifesto statements. Political manifestos are much rarer than news stories, making it difficult for smaller language models or those trained on less diverse text to accurately generate manifesto text. Moreover, manifestos cannot be easily prompted with headlines in the same way that news stories are. Larger language models such as GPT-3 and GPT-4 (Brown et al. 2020) can use more abstract prompts than smaller models like GPT-2, including definitions or descriptions of the desired text. Including a description of the desired text allows researchers to incorporate abstract explanations of political concepts

to generate relevant text and allows other researchers to examine the definitions employed in the prompting.

3.3.1 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. Mudde defines populism as “an ideology that considers society to be ultimately separated into two homogeneous and antagonistic groups, ‘the pure people’ versus ‘the corrupt elite’, and which argues that politics should be an expression of the *volonté générale* (general will) of the people” (2004, 543). Both Rooduijn (2019) and Hunger and Paxton (2022) caution against conflating populism with right-wing populism, the “radical right,” nativism, or anti-political establishment parties. 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 (see Table 3). Because GPT-3 was trained on a multilingual corpus, I can specify the desired country and language in the prompt 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.

To train a populism classifier, I also require non-populist sentences, which I generate using two kinds of prompts. First, I draw policy position descriptions from the Manifesto Project (Volkens et al. 2021), taking the ten most frequently identified codes in their dataset and using the policy positions’ descriptions from their codebook as prompts.²⁰ Jungkunz, Fahey, and Hino (2021) caution that measures of populism

²⁰See SI Table SI 2 examples.

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 3: Prompts used to generate populist text.

often pick up on opposition to the current ruling party, as opposed to populism *per se*. To mitigate this issue, I add an additional set of ten hand-written non-populist prompts to cue criticism of other political parties and dissatisfaction with current policies, which were lacking in the Manifesto Project prompted text.²¹ By once again varying the countries, languages, and generation hyperparameters used in the prompts, I obtain 36,509 non-populist synthetic sentences. Note that neither the populist nor non-populist text is generated with party names in the prompts, mitigating the risk of the model picking up party names as a predictive feature (Jankowski and Huber 2023).

I then train a supervised text classifier on the synthetic sentences. I fine-tune a multilingual sentence transformer model (Reimers and Gurevych 2020) and classifier on the sentences using SetFit, an efficient model for short text classification (Tunstall

²¹See SI Table SI 3 for the ten prompts.

et al. 2022).²² 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. As a first evaluation, the model achieves an accuracy of 0.93 and macro F1 score of 0.85 in identifying populist statements in synthetic validation 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. For that, I turn to an evaluation on real manifesto text.

3.4 Validating the populism classifier

After training the classifier, I set aside the synthetic training data and apply the model to real text taken from party manifestos provided by the Manifesto Project. I apply the newly trained populism classifier to each of the sentences or phrases in the Manifesto Project corpus (Volkens et al. 2021), producing a continuous predicted $[0, 1]$ populism score for each sentence. 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:

- “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.”²³

As a further validation, I collect hand annotations on a sample of 450 English-language manifesto statements from the Manifesto Project. Because populist statements are

²²Note that in this context, “fine tuning” refers to updating the weights of the model on a downstream classification task, not to “adaptation”. See the discussion of terminology on page 8.

²³See SI Table SI 4 for 10 randomly selected UKIP sentences and their associated populism scores.

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. I then evaluate the predictions of the models against the hand annotated dataset. Using a threshold of 0.5 to convert the model’s predicted probabilities into binary labels, the model achieves an accuracy of 0.87 and a macro F1 score of 0.76. Depending on the researchers’ needs, they could adjust the threshold to identify a greater share of populist sentences at the cost of false positives, or identify a smaller set of populist sentences with greater precision. Table SI 6 in the Supplemental Information provides the full results. 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.

3.4.1 Populism in Populist Party Manifestos

Finally, 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 4 lists the parties that are coded as populist in the PopuList dataset (Rooduijn et al. 2019) that do not have any 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

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 4: *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.*

or support for “common people” against elites.²⁴

These results are in line with existing work on manifestos, which finds that they are primarily intended for elite consumption Harmel (2018), making them a less natural place for populist rhetoric to appear. Future substantive work should attempt to provide a better theoretical account of why populist parties make different decisions about how much populist rhetoric to include in their manifestos. Populist parties moderate their manifestos in expectation of future coalition membership (Harmel 2018) or engage in a “campaign gamble”, increasing their use of populism when their electoral chances are lower (Dai and Kustov 2022) or when in opposition (Breyer 2022). Future methodological work could extend work on populist rhetoric in campaign speeches and materials (e.g., Hawkins et al. 2019; Dai and Kustov 2022) to measure the discrepancies between these materials and manifestos.

²⁴See Table SI 5 for details on the hand coding.

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 primary strength—generating text—can help researchers overcome the challenges of labeling text for supervised text analysis.

Three main obstacles to supervised text analysis in political science, the problems of labeling, retrieval, and copyright, can be addressed in part by generating synthetic text with the content or style that a researcher desires. Different applications will call for different approaches to generating synthetic text, including using off-the-shelf language models, adapted language models, or using very large models that can be prompted with directions about the desired text. A new adversarial technique helps researchers select the optimal hyperparameters to generate synthetic text that is difficult to distinguish from real text. Each of these approaches is illustrated in three short applications from political science, demonstrating that synthetic text can address the retrieval and copyright issues, and sometimes the problem of labeling, albeit with some penalty in accuracy.

A researcher might wonder whether the step of generating synthetic text is needed at all. If a large language model can reliably generate text with a desired label, it should also be able to apply that label directly to actual text (Ornstein, Blasingame, and Truscott 2022). While models are likely to improve greatly in the near future, the synthetic text approach has several benefits over a zero-shot classifier approach, where an off-the-shelf model provides labels without any example data. First, many of the best models, such as GPT-3.5 or GPT-4, are hosted by third parties that require payment for their use. While still cheaper than hand-coding a large corpus of text (Ornstein, Blasingame, and Truscott 2022), paying to obtain annotations on a

large corpus of text can quickly become expensive. Second, hosted models change rapidly, raising difficulties for future researchers in replicating earlier work. Finally, and most significantly, the classification outputs of language models are often opaque and difficult to evaluate. By using large language models to generate synthetic text and using more well understood models for classification, including bag-of-words models, researchers can evaluate the quality of the generated text and employ classifiers that are faster to run and easier to understand.

This approach to generating synthetic text is applicable to a wide range of tasks. Future work can explore the use of synthetic text to evaluate pre-analysis plans for analyzing free-form text in survey responses, to allow greater transparency in interviews or field notes while preserving privacy, and in developing improved techniques for guiding the content and quality of the synthetic text. Researchers can also explore hybrid approaches that use a mixture of real and synthetic text. Existing labeled text can be augmented with synthetic text, using the existing text to prompt or adapt a model, and then augment the original training corpus with synthetic examples to increase its size and diversity.

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 (e.g. Dathathri et al. 2020; Prabhumoye, Black, and Salakhutdinov 2020; Yogatama, Masson d’Autume, and Kong 2021). 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 ($w_{i-1} \dots w_1$), modifying the language model’s parameters (θ), or changing how words are sampled from the probability distribution (γ). Even as the technology to generate synthetic text improves, applied researchers in political science will still face the same decisions about when and how to generate

synthetic text and how to obtain labels for their synthetic text.

4.1 Acknowledgements

Thank you to Benjamin Bagozzi, Jill Irvine, Michael Jankowski, Adam Lauretig, Erin Rossiter, Brandon Stewart, and the anonymous reviewers for helpful comments. This work was first presented at PolMeth 2022 at Washington University in St. Louis and at New Directions in Text as Data 2022 at Cornell Tech. Thanks also to the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing the computing resources (Reuther et al. 2018).

This work grew out of two ongoing collaborations. The event classification application uses hand-annotated data from a project with Benjamin Bagozzi, Phil Schrodtt, Andy Beger, and Grace Scarborough. The populism application draws on ongoing substantive work with Shahryar Minhas, Christian Houle, and Nicolas Bichay.

Portions of this paper were sponsored 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.

4.2 References

- Anaby-Tavor, Ateret, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. “Do Not Have Enough Data? Deep Learning to the Rescue!” In *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:7383–90. 05.
- 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.
- Beieler, John, Patrick T Brandt, Andrew Halterman, Erin Simpson, and Philip A Schrodt. 2016. “Generating Political Event Data in Near Real Time: Opportunities and Challenges.” In *Computational Social Science*, edited by R. Michael Alvarez. Cambridge University Press.
- Boschee, Elizabeth, Jennifer Lautenschlager, Sean O’Brien, Steve Shellman, James Starz, and Michael Ward. 2015. “ICEWS Coded Event Data.” Harvard Dataverse. <https://doi.org/10.7910/DVN/28075>.
- 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.
- Brown, Tom B, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. 2020. “Language Models Are Few-Shot Learners.” *Advances in Neural Information Processing Systems* 33 (1877–1901).
- Caliskan, Aylin, Joanna J Bryson, and Arvind Narayanan. 2017. “Semantics Derived Automatically from Language Corpora Contain Human-Like Biases.” *Science* 356 (6334): 183–86.
- Chang, Kent K, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. “Speak, Memory: An Archaeology of Books Known to ChatGPT/GPT-4.” *arXiv Preprint arXiv:2305.00118*.

- 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>.
- Dathathri, Sumanth, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. “Plug and Play Language Models: A Simple Approach to Controlled Text Generation.” *ICLR*.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.” *arXiv Preprint arXiv:1810.04805*.
- Devlin, J., Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.” In *NAACL-HLT*.
- 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.
- Doddington, George R, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. 2004. “The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation.” In *LREC*, 2:1.
- Dodge, Jesse, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. “Fine-Tuning Pretrained Language Models: Weight Initialization, Data Orders, and Early Stopping.” *arXiv Preprint arXiv:2002.06305*.
- Dolezal, Martin, Laurenz Ennser-Jedenastik, Wolfgang C Müller, and Anna Katharina Winkler. 2012. “The Life Cycle of Party Manifestos: The Austrian Case.” *West European Politics* 35 (4): 869–95.
- Eder, Nikolaus, Marcelo Jenny, and Wolfgang C Müller. 2017. “Manifesto Functions: How Party Candidates View and Use Their Party’s Central Policy Document.” *Electoral Studies* 45: 75–87.

- 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.
- Gao, Leo, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, et al. 2020. “The Pile: An 800GB Dataset of Diverse Text for Language Modeling.” *arXiv Preprint arXiv:2101.00027*.
- Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. “Generative Adversarial Nets.” *Advances in Neural Information Processing Systems* 27.
- Grimmer, Justin, and Brandon M Stewart. 2013. “Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts.” *Political Analysis* 21 (3): 267–97. <https://doi.org/10.1093/pan/mps028>.
- 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*.
- Halterman, Andrew, Philip A Schrodtt, Andreas Beger, Benjamin E Bagozzi, and Grace Scarborough. 2023. “PLOVER and POLECAT: A New Political Event Ontology and Dataset.” *International Studies Association Conference Paper*. https://doi.org/https://andrewhalterman.com/files/Halterman_et_al_PLOVER_POLECAT_ISA2023.pdf.
- Harmel, Robert. 2018. “The How’s and Why’s of Party Manifestos: Some Guidance for a Cross-National Research Agenda.” *Party Politics* 24 (3): 229–39.
- 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 *EPISA 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.
- Honnibal, Matthew, and Ines Montani. 2017. “spaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing.” *To Appear*.
- Hunger, Sophia, and Fred Paxton. 2022. “What’s in a Buzzword? A Systematic Review of the State of Populism Research in Political Science.” *Political Science Research and Methods* 10 (3): 617–33.
- 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>.
- Jungkunz, Sebastian, Robert A Fahey, and Airo Hino. 2021. “How Populist Attitudes Scales Fail to Capture Support for Populists in Power.” *Plos One* 16 (12): e0261658.
- Kapoor, Sayash, and Arvind Narayanan. 2023. “OpenAI’s Policies Hinder Reproducible Research on Language Models.” *AI Snake Oil (Medium Blog)*. <https://doi.org/Accessed August 11, 2023>.
- Kreps, Sarah, R Miles McCain, and Miles Brundage. 2022. “All the News That’s Fit to Fabricate: AI-Generated Text as a Tool of Media Misinformation.” *Journal of Experimental Political Science* 9 (1): 104–17.
- Lang, Hao, Yinhe Zheng, Yixuan Li, Jian Sun, Fei Huang, and Yongbin Li. 2023. “A Survey on Out-of-Distribution Detection in NLP.” *arXiv Preprint arXiv:2305.03236*.
- Leetaru, Kalev, and Philip A Schrodtt. 2013. “GDELT: Global Data on Events, Location, and Tone, 1979–2012.” In *ISA Annual Convention*, 2:1–49. 4. Citeseer.
- 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*.
- Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. “RoBERTa: A Robustly Optimized BERT Pretraining Approach.” *arXiv Preprint arXiv:1907.11692*.
- 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.
- Mueller, Hannes, and Christopher Rauh. 2017. “Reading Between the Lines: Prediction of Political Violence Using Newspaper Text.” *American Political Science Review*, 1–18.
- 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*.
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. “Scikit-Learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12: 2825–30.
- Peskoff, Denis, and Brandon M Stewart. 2023. “Credible Without Credit: Domain Experts Assess Generative Language Models.” In *Proceedings of the 61st An-*

nual 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*.
- Prabhumoye, Shrimai, Alan W Black, and Ruslan Salakhutdinov. 2020. “Exploring Controllable Text Generation Techniques.” *arXiv Preprint arXiv:2005.01822*.
- Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. “Language Models Are Unsupervised Multitask Learners,” 9.
- Raleigh, Clionadh, Andrew Linke, Håvard Hegre, and Joakim Karlsen. 2010. “Introducing ACLED: An Armed Conflict Location and Event Dataset: Special Data Feature.” *Journal of Peace Research* 47 (5): 651–60.
- 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.
- Roberts, Margaret E, Brandon M Stewart, Dustin Tingley, Edoardo M Airoidi, et al. 2013. “The Structural Topic Model and Applied Social Science.” In *Advances in Neural Information Processing Systems Workshop on Topic Models: Computation, Application, and Evaluation*.

- Rooduijn, Matthijs. 2019. “State of the Field: How to Study Populism and Adjacent Topics? A Plea for Both More and Less Focus.” *European Journal of Political Research* 58 (1): 362–72.
- 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 Europe.”
- 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.
- Shorten, Connor, and Taghi M Khoshgoftaar. 2019. “A Survey on Image Data Augmentation for Deep Learning.” *Journal of Big Data* 6 (1): 1–48.
- Spirling, Arthur. 2023. “Why Open-Source Generative AI Models Are an Ethical Way Forward for Science.” *Nature* 616 (7957): 413–13.
- Sundberg, Ralph, and Erik Melander. 2013. “Introducing the UCDP Georeferenced Event Dataset.” *Journal of Peace Research* 50 (4): 523–32.
- 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, Nicolas AND Regel Matthieß Theres AND Merz, 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.mpd.2021a>.
- Wei, Jason, and Kai Zou. 2019. “EDA: Easy Data Augmentation Techniques

for Boosting Performance on Text Classification Tasks.” *arXiv Preprint arXiv:1901.11196*.

Williams, Adina, Nikita Nangia, and Samuel Bowman. 2018. “A Broad-Coverage Challenge Corpus for Sentence Understanding Through Inference.” In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1112–22. New Orleans, Louisiana: Association for Computational Linguistics. <http://aclweb.org/anthology/N18-1101>.

Wolf, Thomas, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, et al. 2020. “Transformers: State-of-the-Art Natural Language Processing.” In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45.

Wood, Erroll, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Thomas J Cashman, and Jamie Shotton. 2021. “Fake It till You Make It: Face Analysis in the Wild Using Synthetic Data Alone.” In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3681–91.

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*.

Xu, Keyang, Tongzheng Ren, Shikun Zhang, Yihao Feng, and Caiming Xiong. 2021. “Unsupervised Out-of-Domain Detection via Pre-Trained Transformers.” In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 1052–61.

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.

Yogatama, Dani, Cyprien de Masson d’Autume, and Lingpeng Kong. 2021. “Adaptive Semiparametric Language Models.” *Transactions of the Association for Computational Linguistics* 9: 362–73.

Zellers, Rowan, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. “Defending Against Neural Fake News.” In *Advances in Neural Information Processing Systems* 32.

Zhao, Zihao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. “Calibrate Before Use: Improving Few-Shot Performance of Language Models.” In *International Conference on Machine Learning*, 12697–706. PMLR.

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

A Supplemental Information

I estimate 40 hours of GPU usage for this paper. Using energy consumption data on the GPU I use and the local grid carbon intensity, this would produce around 3kg of CO₂ emissions, corresponding to around 7 miles driven by an average US passenger car.²⁵ This excludes OpenAI’s emissions for the use of GPT-3.

A.1 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%). I train an SVM classifier to discriminate between real and synthetic tweets by attempting to predict if a tweet is real or synthetic.²⁶

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. The basic model is a linear SVM with a bag-of-words representation of the tweets. An alternative model, shown Figure SI 2, uses a transformer model

²⁵<https://mlco2.github.io/impact/#compute>.

²⁶I use scikit-learn’s SVM implementation (Pedregosa et al. 2011)

to represent the tweets. The model uses 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. The transformer model performs slightly better than the bag-of-words, but the accuracy and best hyperparameters are similar across the two models.

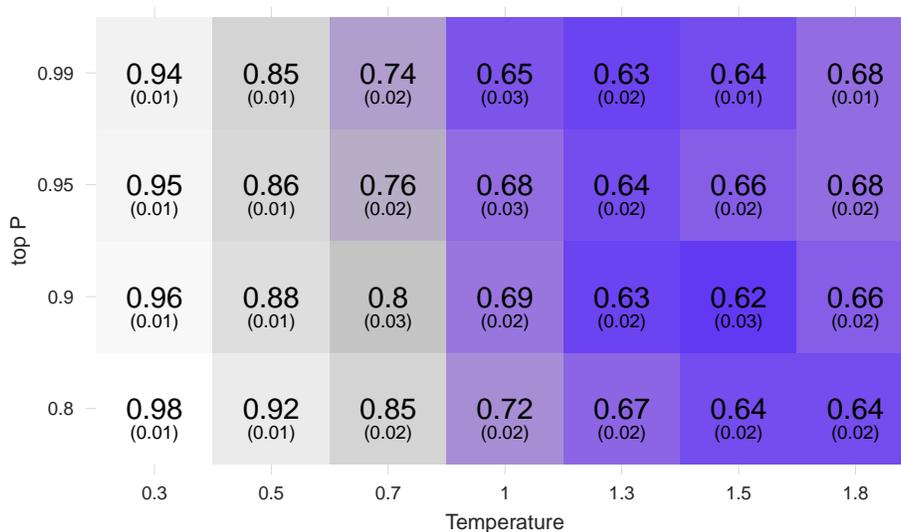


Figure SI 1: *Effect of GPT-2 generation hyperparameters on synthetic tweet quality. Accuracy is the mean out-of-sample accuracy of an SVM classifier trained to discriminate between real and synthetic tweets. Lower accuracy implies better synthetic tweet quality, as the model has a harder time distinguishing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

A.1.1 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 Go-Fund 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 SINCGARS system.
This basically tells you what its main purpose is.

Table SI 1: *Synthetically generated tweets from a GPT-2 model adaptated/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*

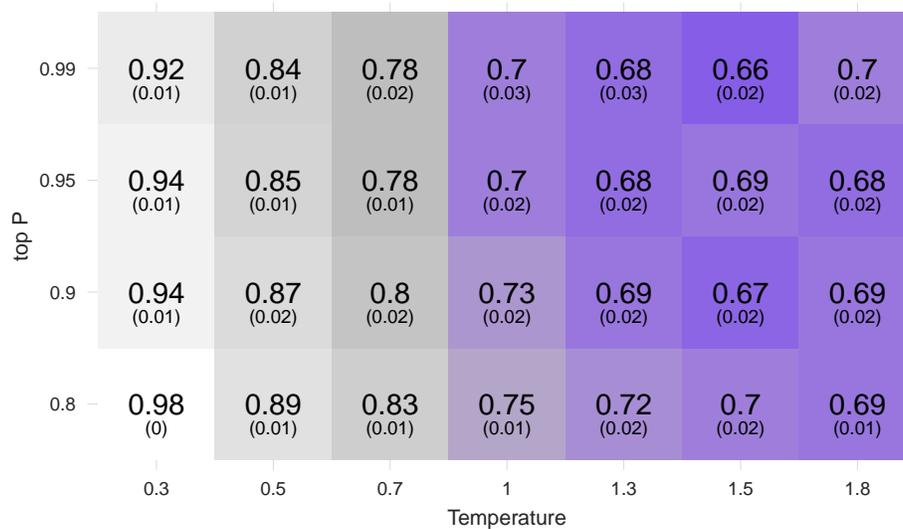


Figure SI 2: *Alternative model showing results using a transformer-based classifier.*

A.1.2 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 skir-

mishes. 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 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”

A.2 Headlines for prompting ASSAULT events

Synthetic news stories describing ASSAULT stories were prompted by using around 40 headlines (see below). To increase the geographic and stylistic diversity of the corpus, I followed the following process. For each headline, I sampled five cities from the list below to generate a dateline and varied the news source. For each unique (headline, city, source) combination, I then generated five random stories.

Negative examples were generated from headlines prompting other event types (protests, meetings, etc.), which are not shown for space reasons.

```
cities = {"Abuja", "Kabul", "Belgrade", "Zagreb", "Khartoum", "Vienna", "Dhaka",  
"Brussels", "Minsk", "Kinshasa", "Beijing", "Bogota", "Sao Paulo", "Havana",  
"Berlin", "Prague", "Moscow", "Washington", "Cairo", "Jerusalem", "Delhi",  
"Tehran", "Rome", "Amman", "Beirut", "Tokyo", "Nairobi", "New York", "Panama  
City", "Oslo", "Damascus", "Bangkok", "Istanbul", "London", "Abu Dhabi"}
```

```
sources = {"Reuters", "AFP", "local sources"}
```

```
"Activist held for ransom by rebel forces"
```

```
"Plane hijacked in suspected terrorist act"
```

```
"Police beat dozens of protestors"
```

```
"Syrian military tortured prisoners, new report states"
```

```
"Evidence of torture uncovered in government prison"
```

```
"Two prisoners put to death"
```

```
"Execution carried out in Iranian prison"
```

```
"Two women raped in capital"
```

```
"New reports of sexual violence in ongoing conflict"
```

```
"Al Qaeda leader killed in targeted drone strike"
```

```
"Iranian scientist assassinated"
```

```
"Russian dissident killed in apparent assassination"
```

"Two civilians hacked to death with machetes in rural area"
"Buildings destroyed in shelling"
"Israeli bulldozers clear houses"
"Four shot in ongoing gun battle"
"Police fire on demonstrating crowd"
"Truck bomb kills three soldiers"
"Explosions rock city"
"Landmine kills three"
"Bomb detonates in downtown capital"
"Dozens killed in suicide bombing"
"Suicide bomber kills three"
"Man detonates explosive vest at checkpoint"
"US tanks and armored vehicles begin assault in Iraqi town"
"Heavy artillery shelling continues"
"Police disperse protest with water cannons and tear gas"
"Police fire weapons in the air to disperse mob"
"Indian police use lathi charge to break up protest"
"Hundreds expelled from homes in ethnic cleansing"
"Ethnic cleansing ongoing in conflict"
"Serbian forces expel Bosnians from villages in cleansing operation"
"Civilians slaughtered in massacre"
"Syrian air force uses chemical weapons against civilians"
"Four killed in sarin gas attack"
"Anthrax attack infects three"
"Four killed in air strike"
"War planes pummel rebel positions"
"Allied aircraft enforce no-fly-zone, shooting down Iraqi fighter plane"

"Air Force UAV destroys enemy targets"

"Drone strikes increase as conflict intensifies"

"Man hacked to death with machete"

"Angry mob throws rocks and bottles"

"Local opposition leader beaten with baseball bat"

"Terrorist group releases poison gas, killing three"}"

Thus, an example of a complete prompt would be "Suicide bomber kills three HAVANA (local sources) –".

A.3 MNLI zero shot classifier

Figure 3 includes a baseline zero-shot ASSAULT event classifier. The model is an entailment model that takes a *context* (here, the news article), a hypothesis (“The story describes a violent attack”), and returns the probability that the hypothesis is *entailed* by the context. Specifically, the model is RoBERTa-large (Y. Liu et al. 2019), which has been fine tuned on Multi-Genre Natural Language Inference (MNLI) dataset (Williams, Nangia, and Bowman 2018), which contains 443,000 annotated entailment examples. The model is zero shot in that it is not trained on any annotated data from the specific task, but instead relies on general classification performance from its diverse training data. Entailment models, specifically RoBERTa fine tuned on MNLI, have previously been used for zero-shot model for event classification in news stories (Halterman et al. 2021).

A.4 Populism Prompts and Validation

Table SI 2 shows an example non-populist prompt using the Manifesto Project codebook. Table SI 3 below show the prompts used to generate non-populist text. Table SI 4 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 Expansion (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 SI 2: *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 it is campaigning against. Write 10 statements that {adjective} party might make in the style of a political manifesto (in the {lang} language):”

Table SI 3: *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 SI 4: *Ten randomly selected UKIP manifesto sentences and their predicted populist scores.*

“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 SI 5: *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 SI 6: *Populism classifier validation results.*