Narratives to Numbers: Machine Learning Applications for Employing Text as Data in Economics

By Ethan Hartley*

Despite extraordinary advancements in artificial intelligence for text analysis, these methods are scarcely applied in economics. This study discusses the potential of Large Language Models (LLMs) as tools for extracting economically relevant data from text. I provide insights on fine-tuning LLMs and explore their effectiveness in utilizing text as data through a case study evaluating the well-established Economic Policy Uncertainty index. LLMs provide 53% to 811% relative improvements in performance over dictionary-based alternatives and prove more robust in data constrained environments while better capturing the perceptions of underlying economic agents.
JEL: C45, C38, D8, E6
Keywords: text as data, large language models, LLM, deep learning, machine learning, natural language processing, policy uncertainty, bag-of-words

Current projections indicate by 2025, approximately 480 million terabytes of data will be created each day (IDC, 2019). The rapid growth of this resource can largely be attributed to the inextricable link between data, technological innovation, and the rise of the Internet of Things. Yet another key factor, driven in part by the desire to utilize this vast amount of information, has been a fundamental shift in how we perceive practically relevant data. What once was limited to the confines of tabular formats has shifted to an understanding that nontraditional sources, be that images, non-digitized documents, or text, can be vital sources of information. In the context of economic analyses, which have an increasingly interdisciplinary focus, the potential to leverage this information to create new and more diverse datasets is invaluable in expanding our understanding of human behavior. Despite the value of such information, economists have been exceptionally slow to take-up contemporary artificial intelligence (AI) methods of leveraging nontraditional data in favor of simple alternatives. However, considering the emphasis placed on unbiased and properly identified econometric estimation, it is only fitting that we apply the same rigorous evaluation to the methods of this

* Hartley: University of Hawaii at Manoa (email: ehartley@hawaii.edu) Acknowledgements: The technical support and advanced computing resources from University of Hawaii Information Technology Services Cyberinfrastructure, funded in part by the National Science Foundation CC* awards 2201428 and 2232862 are gratefully acknowledged. The author is funded by the Department of Energy Innovation in Buildings Graduate Research Fellowship. There are no other sources of financial support or conflicts of interest to report.

underlying data-generating process.[1]

A key area of research where AI is being heavily utilized is the development of new methods and open-source solutions to extract and digitize information from physical documents. Optimal Character Recognition (OCR) is a sub-field of machine learning which has gained popularity within the economics literature and aims to convert text in images into a machine-readable format. Prominent examples of such developments include open-source document digitization solutions and accessible and flexible frameworks for linking data to text via Large Language Models (LLMs) (Bryan et al., 2023; Arora and Dell, 2024). Shen, Zhang and Dell (2020) demonstrate the versatility of these methods in digitizing documents of various formats and discuss the implications of resulting datasets for future model development and real-world applications. One such application being Dell et al. (2023), which develops a deep learning paradigm for extracting article texts from images of newspapers and creates a freely available digitized dataset containing millions of historical articles. Such a dataset, when paired with the fine-tuned model discussed in Section II, could have extraordinary implications for economic historians interested in the role of uncertainty and economic policy in the late 19th century—a period that has historically lacked practically relevant data.[2] In contrast to these more technocratic studies, I explore the role of deep learning in pragmatically utilizing text as data in economics. Specifically, this study evaluates the potential of LLMs to extract economically relevant indices from text. The proposed method is contrasted with simple machine learning and dictionary-based approaches through a case study of the newspaper-based Economic Policy Uncertainty (EPU) index of Baker, Bloom and Davis (2016).

In comparison with the work on generating textual data, this domain of *text as data* has a far richer history in the literature. Loughran and McDonald (2011)'s work on sentiment analysis in financial contexts established that not all applications of textual data are alike and that domain context is critical. Considering the nuance and domain specific vocabulary surrounding discussion of economic principles and policy, economists should be particularly attuned to this issue. It is also important to recognize this issue is not limited to linking sentiment to terms but poses a challenge for all keyword-based strategies. A related example of keywords carrying specific connotations and latent information can be found in evaluating the political slant of U.S. media (Gentzkow and Shapiro, 2010). The authors adopt a keyword approach to measure the political alignment of newspaper outlets based on the frequency of politically aligned phrases in published articles. Baker, Bloom and Davis (2016) build upon the notion of extracting numerical data from text while mitigating potential bias by using empirically-based

[1]Regardless of whether you are directly involved in this line of inquiry, the growth of commercialized AI is expanding the volume of AI-generated information and researchers should be cognizant of where their data comes from and what limitations that might impose.

[2]The LLM methodology discussed herein is particularly well suited to this task given its increased capability of processing "dirty" textual data. An area, as discussed in Section II, which can have considerable performance implications for dictionary-based methods.

keyword sets in their Bag-of-Words (BOW) classifier of EPU. It is noteworthy that, despite significant advancements in Natural Language Processing (NLP) and recognition of the complexity of text, as well as the increasing role of sophisticated modeling in using text as data discussed in Gentzkow, Kelly and Taddy (2019), the majority of applications in the economics literature continue to use dictionary-based methods.

Despite the nascency of the literature, there are a few key examples of using LLMs to employ textual data within the economic lens of analysis. A recent study out of Google DeepMind evaluates the efficacy of using LLM embeddings directly in high-dimensional regression frameworks in place of traditional feature engineering (Tang, Yang and Song, 2024). Vafa et al. (2024) and Du et al. (2024) focus on the applications of transformer-based architectures and LLMs, respectively, for predictive modeling of career trajectories using resumes. Both studies highlight the capability of deep learning in using text as data to refine modeling strategies and advance our comprehension of labor markets. More pertinent to this study is Bybee (2023)'s work on generating "economic expectations" from news articles using LLMs. However, unlike the "economic expectations" framework, which employs LLMs in a zero-shot generative capacity, this study demonstrates how to ensure consistency in LLMs and produce probabilistic predictions for human-designated labels.[3]

I contribute to the existing literature on utilizing deep learning to derive economic insights from text in five key ways. First, I explicitly contrast the performance differences between traditional dictionary-based and sophisticated NLP methods. Second, I address the fallacy that non dictionary-based methods are inherently constrained by data availability, demonstrating that LLMs match or surpass simple classifiers using only articles titles or brief summaries. Third, I show that within the context of the EPU case study, LLMs more effectively emulate human perceptions of economic policy and uncertainty relative to dictionary-based and simple machine learning classifiers. Fourth, I discuss the advantages and disadvantages of various fine-tuning strategies and provide insights on the limitations and necessary precautions of generating economic indices from text. Lastly, I provide a simple, open-source modeling strategy that is easily applied and adaptable to new contexts, capable of running on a personal laptop and free of coding barriers.

This paper is organized as follows. Section I provides an overview on Natural Language Processing (NLP) and contemporary language modeling. Section II presents a case study exploring the proficiency of Large Language Models (LLMs) in extracting economic data in relation to the Economic Policy Uncertainty (EPU) index. Section III concludes.

[3]Zero-shot tasks are defined as the model being used "as is", i.e. the user does not provide any additional training data. "Generative capacity" in this context refers to querying the model such that the output is generated text with no explicit restriction on what is returned.

I. Utilizing Text as Data

Natural Language Processing (NLP) broadly refers to the ability of modeling paradigms to interface with and comprehend human language and has origins dating back to the original conception of *machine intelligence* (TURING, 1950). Soon after, Weizenbaum (1966) introduced ELIZA as one of the earliest examples of "natural language communication between man and machine." Though the concept of an interactive program with which people could communicate was revolutionary, ELIZA was, at its core, merely a collection of rules constructed around a set of keywords designed to build upon user input. This "style" of language modeling—embodying elements of human language through strict linguistic guidelines and rigid heuristics—dominated much of the NLP literature for the next 50 years.

Faced with these restrictive methods, which often underperformed and were extremely costly, there was an overcorrection toward simple count- and dictionary-based approaches. Rather than implementing complex rules of response triggered by keywords for programmatic interfacing with humans, scholars sought to assign labels to textual data using simple rules defined by the existence of keywords within a text. The basal structure of these methods rely on the existence, frequency, or co-occurence of pre-defined keywords. In its simplest form, this approach searches for a single keyword in a text and, if it is found, the text is assigned a positive classification. More sophisticated count-based methods build upon this notion to include continuous measures, such as the normalized percentage of keyword occurrence within a text. A secondary avenue of improving keyword modeling strategies was the development of dictionary-based methods, such as Bag-of-Words (BOW), which enforce the co-occurence of multiple keywords from different sets (bags) to trigger a positive classification.

The advent of methods for generating numerical representations of text, often referred to as tokenization or vectorization, facilitated a transition from traditional keyword NLP classification methods to data-driven strategies.[4] The adoption of numerical text representations, i.e. each text is assigned a N-dimensional vector representation, laid the groundwork for more robust and empirically-driven classification strategies that consider the full text instead of a subset of keywords. Support Vector Machines (SVMs) provide a straightforward example, as they project the vector representations of each text into an N-dimensional space and then attempt to draw a boundary that distinguishes texts of different labels. In this way, certain elements of the text may contribute more to the resulting classification, but all elements of the text contribute to the resulting prediction. Although the SVM approach offers both conceptual and functional improvements over keyword alternatives, a key weakness of all aforementioned classifiers is their binary nature.[5] The probabilistic-NLP movement has focused on translating these

[4]There are various potential tokenization strategies, with common methods focusing on term frequency within a document (Count) and inverse frequency of terms within a corpus (TF-IDF).

[5]Strategies do exist for calculating pseudo-probabilistic estimates based on the distance of the vector

numerical representations into probabilistic predictions, enhancing the information retrieved from each text. Prominent methods for probabilistic classification of texts include Naive Bayes, logistic regression, and neural networks.

A. *Large Language Models*

Large Language Models (LLMs) mark the latest development in the field of NLP and are widely recognized as the most successful application of deep learning for modeling complex sequences and embodying elements of human language. In contrast to traditional recurrent neural networks (RNNs), which impose a rigid sequential structure on the data, contemporary LLMs dynamically assign weights to all elements within a sequence. Equations 1 and 2 provide a stylized example of the flexibility offered by LLMs, where the RNN prediction Y_N in Equation 1 is simply a set of arbitrary functions (such as activation functions) $g()$ and $\sigma()$ applied to the observation X_N and the weighted previous representation $\omega_{N-1}X_{N-1}$. In contrast, the LLM prediction represented by Equation 2 simultaneously weights all elements of $X \in [X_1, ..., X_N]$.

$$Y_N = g(\sigma(X_N, \omega_{N-1}X_{N-1})) \tag{1}$$

$$Y_N = g(\omega_n X_n \ \forall \ n \in [1, N]) \tag{2}$$

This flexibility, and in large the boom of modern NLP, is attributable to the concept of attention and the work of Vaswani et al. (2017) on the Transformer architecture. Intuitively, these mechanisms allow the model to learn robust representations of texts, identifying and dynamically weighting relationships among elements with respect to all other elements. Theoretically, if a model were developed to identify elements of economic policy in a text containing the phrases 'there is wide disagreement among policymakers on inflation' and 'thank you for reading,' the model would assign much greater weight to the first phrase when making a prediction, regardless of its position in the text. The ability of these models to capture universal relationships with remarkable precision has lead to extensive adoption of deep learning approaches. Today, most NLP techniques utilize Transformers as simple building blocks and rely on vast amounts of data to learn robust representations of human language. When millions, in some cases trillions, of tokens are fed through this framework, the resulting models become exceptionally powerful tools with the ability to emulate human-like language comprehension and aptitude.

B. *Fine-Tuning*

Transfer learning (typically referred to as fine-tuning in the context of LLMs) takes a model which has processed large quantities of related data, commonly

from the separating boundary; however, these implementations are less straightforward than probabilistic classifiers.

referred to as a pre-trained or foundation model, and further fits the model on a new dataset for a specific task. During the pre-training stage, the model processes extremely large corpora of texts to learn the general patterns and nuances of language. Subsequently, a portion of model weights—intuitively, the fundamental semantic relationships learned during pre-training—are held fixed while a smaller dataset is used to update the remaining weights for a new, more specialized objective.[6] Effectively, the model retains relationships learned during pre-training but is able to adapt which portion of the sequence should now be weighted more heavily for the task at hand. This fine-tuning procedure can be applied in a number of ways including: generative frameworks for creating new texts, classifying texts into categories, and direct predictions of numerical indices.

It is important to note that there are any number of fine-tuning strategies, each with their own merit, that can accomplish essentially the same tasks. In the context of LLMs, a key consideration is whether you utilize prompt engineering to solicit a response from the model, or if you structurally enforce the response of the model via a classification head. Figure 1 illustrates the latter method, where texts are passed through the tokenizer and embedding layers with a context length N, not to exceed the maximum context length of L. This process generates a matrix of context length N and embedding dimension E that is fed through the baseline architecture and later a linear classification layer resulting in a probabilistic estimate of class assignment. Throughout this process, the hyperparameter space is searched and optimized using Optuna's optimization software framework.

Despite the seeming loss in flexibility by structurally implementing a classification head, there are a few key benefits of this tuning strategy. First, a classifi-

[6]Weights refer broadly to the learned links between nodes within a neural network.

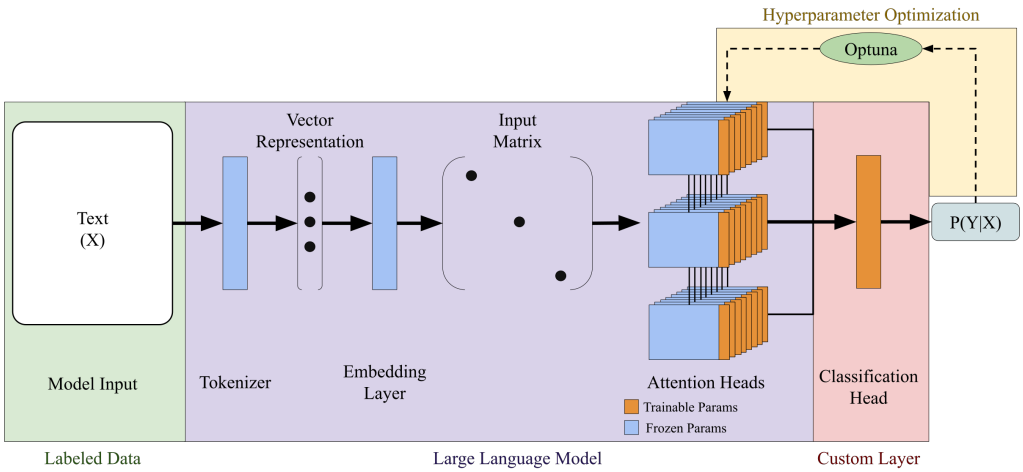


Figure 1. LLM Fine-Tuning Framework for Classification

cation head effectively restricts the space of possible responses and thus reduces the complexity of finding a minima of the loss function. Second, this method eliminates the potential for hallucination; the model may return an incorrect prediction, but it will not return something that has never, does not, or will never exist. Hallucination is of particular concern in a traditional zero-shot task as seen in Bybee (2023). In this case, there is no explicit truth-telling mechanism, i.e. the model will always return an answer and there is no incentive for that answer being correct. Moreover, by querying the model in this way, you are implicitly soliciting opinion rather than facts and yielding epistemological authority to an "agent" that may or may not actually have the capacity for comparison across examples or time. The temporal dimension becomes particularly problematic in the case of LLMs as these models are typically trained on corpora with a well-defined end date T, so expecting accurate insights for future time points $T + 1$ is likely unrealistic and largely outside the scope of the model's intended capabilities.[7]

II. Quantifying Economic Policy Uncertainty

To investigate the effectiveness of contemporary NLP techniques in extracting economically relevant insights from text, this case study focuses on modernizing the methods of an established newspaper-based indicator of uncertainty. The Economic Policy Uncertainty (EPU) index has garnered approximately 13,000 citations and is the source of inspiration for hundreds of economic indices permeating dozens of sub-fields including Macro, Trade, Health, and Environmental Economics. I compare the performance of dictionary-based classification methods with simple machine learning algorithms, Support Vector Machines (SVMs), and Large Language Models (LLMs). In addition, this work also explicitly considers the costs of each method: including computational efficiency, limitations on available data, and broader accessibility within the economics research community. For that reason, note that all model development (and final models) discussed here are freely accessible and implemented in a way that allows most instances to run on a personal laptop.

As introduced by Baker, Bloom and Davis (2016), the EPU index is a time series measure that gauges the perceived general uncertainty about economic policy within a specific population, geography, and time. The scope of "uncertainty" in this instance is defined broadly and may include elements of policy formation, implementation, and revocation as well as the action, or inaction, of affected agents. The key assumption to their news-based index is that the level of policy uncertainty is identifiable by the frequency with which news outlets report on the topic of interest. Thus, to calculate a measure of policy uncertainty, you simply compute the percent of all news articles in a given time and newspaper relating to

[7]Note that, in practice, the vast majority of LLM fine-tuning is a few-shot task. This is largely attributable to the sheer number of model parameters in contemporary LLMs—it takes an exceptionally large amount of data or very small learning rate to require more than a few epochs to encode new information without overfitting.

or discussing the topic. These values can then be normalized and combined across news outlets to create a single, continuous measure of policy uncertainty. When the index is greater than the normalized mean, typically set to 100, this indicates times of greater policy-induced uncertainty. This same conceptual framework can then be extended to any "category" of policy uncertainty. For example, Baker, Bloom and Davis (2016) report greater healthcare policy uncertainty around 2010 in response to the Affordable Care Act.

The influence of Baker, Bloom and Davis (2016) in extracting economically meaningful data from text cannot be overstated, as it underpins much of the contemporary work evaluating textual data in economics and has spawned hundreds of related indices across various topics, geographies, and temporal bounds. In general, scholars exploring policy uncertainty follow the same two-step classification and normalization procedure developed in Baker, Bloom and Davis (2016). BOW is the most frequently employed method of classification and assigns policy uncertainty labels to individual texts based on the presence of words from predefined term sets. The original EPU classifier looks for at least one occurrence of terms in each bag: "*economic*" or "*economy*", "*uncertain*" or "*uncertainty*", and "*Congress*", "*deficit*", "*Federal Reserve*", "*legislation*", "*regulation*", or "*White House*". Subsequently, binary labels of policy uncertainty are then aggregated to the percent of all articles for each newspaper and time that have been classified as relating to policy uncertainty. These values are then normalized and presented as a continuous time series index following the procedure detailed in Equations (1)-(5) where a_{ijt} represents the jth news article in newspaper i in time t and σ_i is the standard deviation of X_{it}.

$$X_{it} = \frac{\sum_j a_{ijt}^{EPU}}{J} \tag{3}$$

$$Y_{it} = \frac{X_{it}}{\sigma_i} \tag{4}$$

$$Z_t = \frac{1}{n} \sum_{i=1}^{n} Y_{it} \tag{5}$$

$$\overline{Z} = \frac{1}{T} \sum_{t=1}^{T} Z_t \tag{6}$$

$$EPU_t = \frac{Z_t}{\overline{Z} \cdot 100} \tag{7}$$

Despite the popularity of such indices, there has been very little methodological innovation in the space. Simple adaptations may consider the frequency with which terms occur, weight BOW classifications by text sentiment (Charemza, Makarova and Rybiski, 2023), or place constraints on the proximity of term co-occurrence (Benguria et al., 2022). Going one step further, Azqueta-Gavaldn et al. (2023) suggests a simple strategy which selects a corpus of articles that

contain any words related to "*economy*" or "*uncertainty*" and then uses a simple unsupervised learning algorithm, Latent Dirichlet Allocation (LDA), to assign a topic to each article. Charemza, Makarova and Rybiski (2022) also employ LDA along with various simple NLP techniques like *word2vec* to construct a Russian index of policy uncertainty. Qureshi et al. (2022) develop the *EPU-NLP* index which uses a number of preprocessing strategies for article filtration (Rapid Automatic Keyword Extraction, GrapeNLP, PyLucene, RoBERTa, and SBERT) before defaulting to the standard EPU BOW classifier. While all of these approaches attempt to impose further structure and improve the modeling process, they fail to address the fundamental cruxes of dictionary-based methods: they are often minimally empirically driven, fundamentally correlational in nature, and in no way consider the underlying semantics or context of the text that is being classified.

Tobback et al. (2018) are among the earliest to use machine learning (ML) on a small labeled dataset of newspaper articles to classify EPU. Their SVM achieves an accuracy of 88% on a test set of 100 Belgian news articles. Noailly, Nowzohour and van den Heuvel (2022) further explore ML classifiers using a dataset of 2,500 labeled articles and fit a SVM with a precision of 56% and recall of 70%. Both studies point to significant improvements in classifier performance over the BOW method, yet SVMs are often disregarded in favor of the more easily implemented BOW alternative. One such study that thoroughly explores the potential weaknesses of the BOW methodlogy is Keith et al. (2020). The authors provide a number of general advice on the applications of text as data while, more notably for the context of this study, show that resulting indices vary significantly when adjusting both BOW lexicons and when extending to simple methods such as logistic regression. Trust, Zahran and Minghim (2023) is, to the authors knowledge, the only other example of fine-tuning language models to classify policy uncertainty. Despite the authors recovering an F1 score of 63.98% with their fine-tuned BERT model, these results are subject to a critical weakness. In absence of labeled training data, the authors make use of labeling functions—simple rules or heuristics like patterns, keywords, or domain specific strategies that aid in assigning a true label. As such, the trained classifier is only as good as the underlying labeling functions which, in a sense, are no more powerful or causal than the BOW classifier we are hoping to supplant. In contrast, this study exclusively uses human-designated labels of policy uncertainty which are widely considered the gold standard method of labeling textual data.

To address the shortcomings of the status quo bag-of-words (BOW) classifier, and provide insights into more contemporary approaches, I develop a method of generating indices of policy uncertainty from textual data using LLMs fine-tuned on human-designated labels. This case study evaluates three open-source LLMs for classifying newspaper articles as relating to EPU: BERT, Longformer, and Llama 3.[8] Of these three LLMs, Longformer is selected as the optimal architecture

[8]Meta has developed 8, 70, and 405 billion parameter versions of Llama 3. This work selects Llama

to balance classifier performance and computational overhead; however, all LLMs fine-tuned for EPU classification, regardless of the proportion of article used by the model, exhibit superior performance when compared to BOW.[9] As such, all results reported from this point are generated from a fine-tuned Longformer architecture. A comparison of all three LLMs can be found in Appendix Section A.A4. Note that, the fundamental innovation of this work lies solely in a more robust underlying data generating process that has no impact on downstream econometric analyses.

A. Data

A key element of Baker, Bloom and Davis (2016) was a human audit study which employed teams of researchers to review and label more than 12,000 news articles according to their relevance to uncertainty and policy matters.[10] The resulting data consist of article texts spanning from 1900 to 2012 with labels of economic uncertainty (EU) as well as EPU.[11] In addition to binary labels of EU and EPU, auditors noted their degree of certainty for each article coding, as well as any policy category particularly relevant to the text. Within the scope of Baker, Bloom and Davis (2016), these data are used to develop the term sets of their BOW classifier; similarly, all data used for model development and evaluation in this work come directly from the audit study to maintain consistency and ensure fair comparisons of model performance. It should be emphasized that any BOW performance metric reported herein is a theoretical upper bound, as the method (term sets) was developed to minimize the error rate on the same data we are now evaluating.

I utilize 10,393 newspaper article texts with accompanying human-designated labels—widely considered the gold standard for topic assignment within the NLP community—of EPU. A subset of 4,596 articles have sufficient information to be linked to the most relevant policy category: auditors originally selected from a list of 16 policy categories whereas I focus on seven with established bag-of-words term sets for comparison.[12] Labeled articles are randomly split into training (70%), validation (20%), and testing (10%) sets. Model weights are updated based on the training set, incremental performance evaluation (for model hyperparameter optimization) is done on the validation set and final performance evaluations are reported on the held-out testing set. Reporting results on the held-out testing set reduces the risk of overfitting and inflating reported model performance metrics.

3 8B as it is the most computationally feasible and efficient model for the majority of applications.

[9]BERT and Llama 3 models fine-tuned on a context length of 32 tokens technically yield a comparable F1 score. However, given that these methods are utilizing considerably less data, this is viewed as an improvement in performance.

[10]Auditors were thoroughly vetted and closely supervised while also adhering to a standardized auditing guide. Currently available at policyuncertainty.com.

[11]This study utilizes articles from the *Chicago Tribune, Dallas Morning News, Los Angeles Times, Miami Herald, New York Times, San Francisco Chronicle, Wall Street Journal,* and *Washington Post.*

[12]Policy categories include: monetary policy, fiscal policy and government spending, taxes, entitlement programs, healthcare, financial regulation, and trade policy.

TABLE 1—DATA SPLITS

	Full Article					
	Random			Temporal		
	# Articles	EPU = 0	EPU = 1	# Articles	EPU = 0	EPU = 1
Train	7,275	4,081	3,194	7,660	4,279	3,381
Validation	2,078	1,159	919	1,915	1,012	903
Test	1,040	546	494	818	495	323
	Titles					
	Random			Temporal		
	# Articles	EPU = 0	EPU = 1	# Articles	EPU = 0	EPU = 1
Train	7,230	3,988	3,242	7,612	4,182	3,430
Validation	2,065	1,154	911	1,904	1,037	867
Test	1,034	570	464	813	493	320

A temporally-based secondary split randomly assigns articles from 1985-2005 to training and validation sets and assigns all articles from 2006-2012 to the held-out test set. A final data split is performed on the policy category subset of articles using an iterative sampling procedure to ensure equal class proportions across splits, the details for which can be found in Appendix Section A.A3. Article counts and class proportions for the random and temporal splits are described in Table 1.

B. Methods

All BOW classifiers use term sets established within Baker, Bloom and Davis (2016) or, in the case of some policy categories, draw on subsequent work, the term sets for which are officially hosted on policyuncertainty.com. SVM parameters are optimized via gridsearch and are fitted and evaluated on the aforementioned training and validation splits.[13] Unless otherwise noted, both BOW and SVM classifiers use the full available text. LLMs are fine-tuned using context lengths (maximum input sequence length) of 128, 256, 512, 1024, and 2048 tokens (words). In addition, I construct SVM and LLM models with a context length of 32 tokens, utilizing titles alone rather than article text. For each context length and model, a minimum of 576 gpu training hours or 15 seperate models are trained using a structured search of the hyperparameter space to optimize performance and

[13]SVMs are implemented within scikit-learn and are evaluated for all possible combinations of: regulatization [0.1, 1, 10, 100, 500, 1000], kernel [linear, rbf, poly, sigmoid], gamma [scale, auto], and vectorizer [TfidVectorizer, CountVectorizer].

generalizability.[14] The trial (model) with the minimum validation loss is selected as optimal and will be made freely available in just a handful of lines of code within the Huggingface Hub. See Appendix Sections A.A1 - A.A3 for detailed discussion on model fitting and hyperparameter optimization and Appendix Section A.A1 and Figure A1 for discussion on the Huggingface platform and a Python example implementation of the LLM prediction pipeline.

Classifier performance is evaluated using standard machine learning metrics as defined by equations (7) - (10) where $TP, TN, FP,$ and FN represent the count of true positives, true negatives, false positives, and false negatives, respectively. While the F1 score will be the main focus of this study due to its holistic nature as the harmonic mean of precision and recall, significant emphasis will also be placed on model recall. Recall intuitively reflects the proportion of total positive instances (EPU=1) that are correctly classified. LLMs are further evaluated using the Area Under the Receiver Operating Characteristic curve (AUROC). The AUROC serves as a measure of the classifier's ability to distinguish between positive and negative classes: a value closer to 1 indicates high performance while a value of 0.5 suggests that the classifier performs no better than random chance.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{8}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{9}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{10}$$

$$\text{F1 Score} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{11}$$

C. Classifying EPU

This section outlines key findings from fine-tuning Longformer for identifying themes of policy uncertainty within news articles.[15] Although this study employs a lightweight architecture to reduce barriers to adoption, it's worth noting that models like Llama 3 405B likely offer substantial performance improvements and merit consideration in subsequent work. Initial findings from this case study can be condensed into two primary points and are summarized by Figure 2.

First, the dictionary-based methods that dominate the economics literature are highly volatile to both term inclusion and implementation strategy. The top pane of Figure 2 evaluates the consistency of a "baseline" BOW classifier which allows

[14] Hyperparameters are components dictated by the model developer which impact the model fitting procedure.

[15] Recall Appendix Section A.A4 provides comparative analyses of all three LLMs considered in this work.

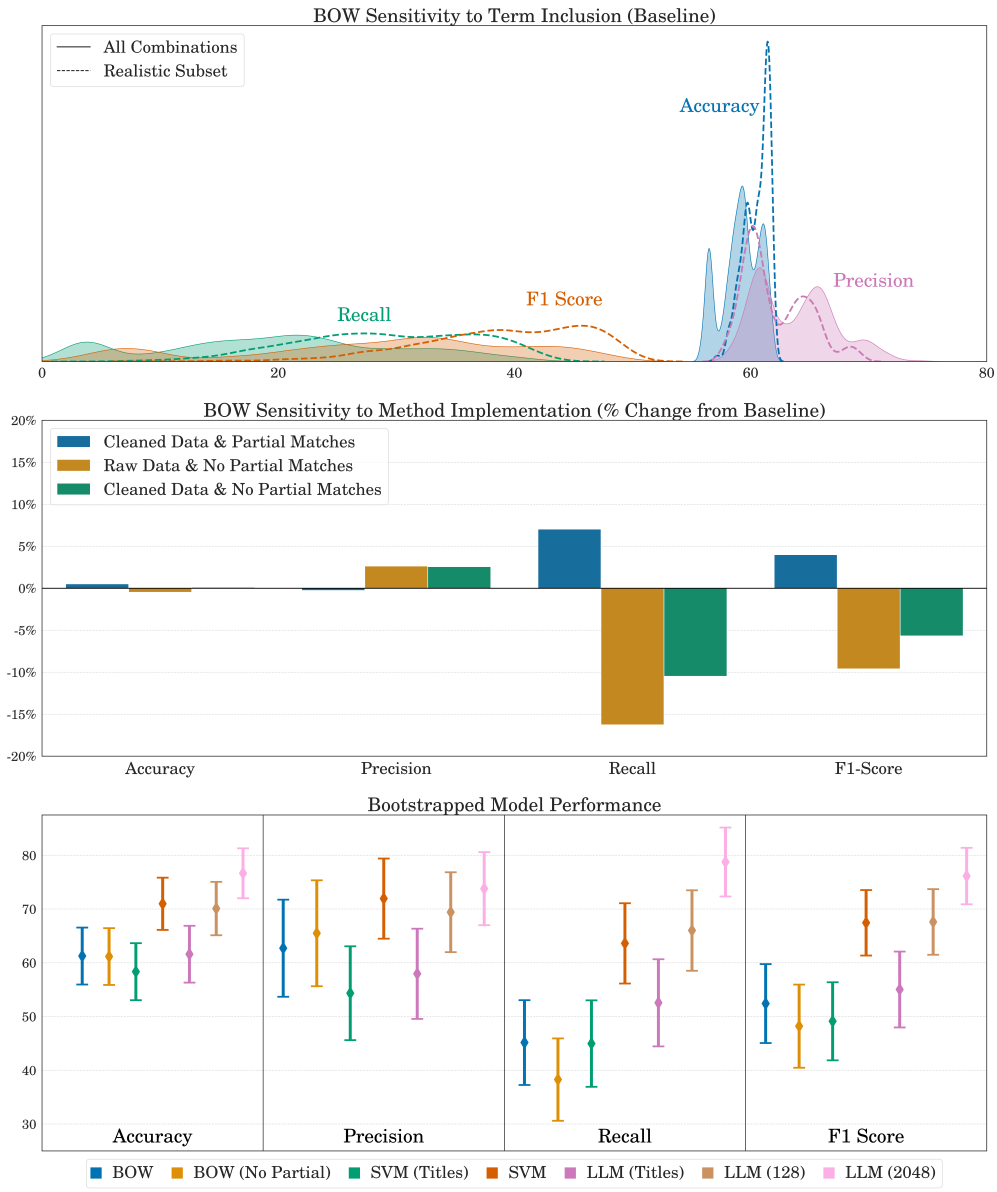


Figure 2. Classifier Performance

for partial term matching on raw (no additional cleaning) textual data. Kernel density estimates approximate the distribution of performance metrics across the 10,731 unique combinations of terms used to classify newspaper articles as relating to EPU. The realistic subset is comprised of 1,864 possibilities that use at least two terms in each of the economic and uncertainty bags and contain at least three terms in the policy bag. In both cases, the distributions of performance are both centered at a low mean while also exhibiting an exceptionally wide dispersion. Additionally, the middle pane evaluates the change in performance relative to the "baseline" model associated with various implementation strategies, including more refined data cleaning and the absence of partial term matching. While seemingly innocuous, the tangible variation in performance is highly concerning, especially for a method that is commonly deployed via API calls—binary search strings that return articles with the desired keywords—in external databases, where the exact matching strategy and data structure may be unclear.

Second, empirically driven machine learning algorithms far outperform dictionary-based classifiers. The bottom pane of Figure 2 depicts mean model performance with 95% confidence intervals calculated using a bootstrapping procedural resampling of 250 articles from the held-out test set with 5,000 independent draws. Full test set performance metrics can be found in Appendix Table A4. When comparing SVMs fit on full article texts to the BOW baseline, we see 36.07% and 49.24% relative improvements in F1 score and recall. These results suggest that, at a minimum, researchers should be employing simple machine learning strategies when data permits. Improvements in performance are even more remarkable when considering the top performing LLM (Longformer with a maximum context length of 2048) exhibiting a relative increase of 53.53% in F1 score and an 85% relative improvement in recall. Intuitively, the improvement in recall demonstrates a significant boost in the models ability to accurately identify articles related to EPU, while also highlighting the limitations of the BOW classifier, which struggles to capture the nuanced concepts that humans associate with EPU. Looking past the first step of generating the EPU index, a comparison of resulting quarterly EPU indices using temporally-trained LLMs with the audit (human labels) and BOW baselines can be found in Appendix Figure A4. Indices exhibit Pearson's R correlations with the baseline audit index over the held-out testing set of 0.24 and 0.74 for BOW and LLMs, respectively. Not only do LLMs far improve the underlying data generating process, but by doing so, the resulting index more closely represents the "ground truth" perceptions of auditors.

A potential limitation of the fine-tuned LLM approach is the need for full article texts; however, acquiring concise abstracts or titles is usually more feasible.[16] To address potential methodological limitations spawning from data constraints, I construct models with maximum context lengths of 128 and 32 tokens, akin to the use of a brief summary or article titles alone. In general, there is a positive

[16]Databases like ProQuest or NexisUni typically allow users to export large quantities of article citations which can easily be used in this modeling paradigm.

relationship between maximum context length and model performance, with inferior performance often driven by articles exceeding the maximum length. See Appendix Section A.A4 and Table A2. Even so, I find that in these extremely constrained environments, LLMs using title alone and brief summaries provide 11.10% and 36.43% relative increases in F1 score, respectively. In the case of short summaries, we are vastly outperforming the status quo BOW while simultaneously matching the performance of SVMs making use of the full article text. This evidence suggests that, when data can be obtained, a dictionary-based classifier is almost never advisable given the accessibility of superior alternatives.

D. Emulating Human Perception

When integrating AI into the data generating process underlying economic analyses, as is the case with our EPU example, it is critical that our chosen model not only adequately predicts the object of interest, but also emulates the perceptions of economic agents. To test the ability of LLMs to learn relationships akin to human perceptions of EPU, I select the set of articles within the external test set with labels of auditor certainty and stratify into *certain* and *uncertain* label

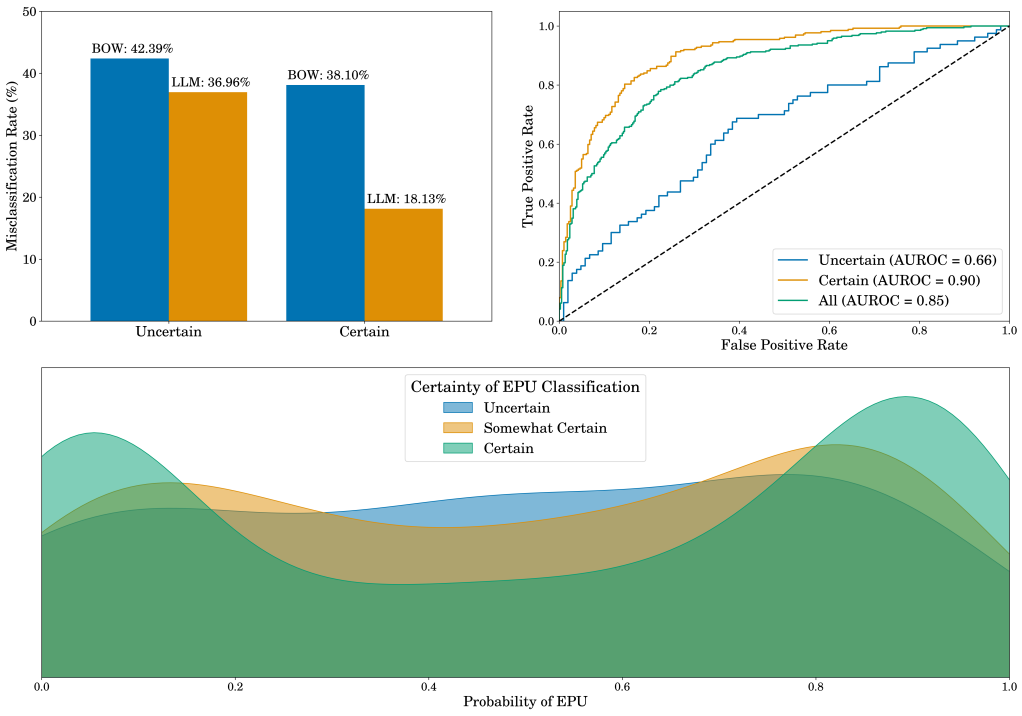


Figure 3. Performance Under Uncertainty

assignment classifications.[17] The top panes of Figure 3 illustrate the performance of fine-tuned language models by human auditor certainty. In contrast to BOW, the error rate of the fine-tuned LLM is reduced by more than half when evaluating articles labeled with a high degree of certainty. Additionally, we see AUROC curves that perform significantly better on *certain* articles than *uncertain*—recall the dashed line of identity represents the equivalent of random guessing while an AUROC of one represents a perfect classifier.

Reflecting broadly on the probabilistic-NLP movement, it is widely recognized that indicators drawn from textual data are rarely binary in nature. The indicators of perceived uncertainty discussed herein are no different. As mentioned previously, a key benefit of the proposed deep learning approach is that the estimator is probabilistic in nature. Not only does this improve the richness of information extracted from the article but also offers the potential to refine the index normalization procedure. If economic agents might interpret the text on a spectrum, then the optimal modeling procedure would encode that information for downstream econometric estimation. Ideally, and as observed in the bottom pane of Figure 3, there is an outward shift in density as the certainty of label assignment increases. In other words, as the certainty of human auditors increases, so too does the models confidence in classification. The fact that we observe this relationship suggests that a probabilistic index of EPU may better reflect human perceptions of the information we hope to quantify. This result also represents a significant improvement in the spectrum of information retrieved in contrast to the degenerate distribution resulting from the BOW classifier. Appendix Figure A5 compares binary and probabilistic LLM indices with the probabilistic EPU index correlating at 0.74 rather than 0.69.

E. Policy Uncertainty Categories

In the spirit of facilitating broader methodological adoption of LLMs, I explore the ability of classifiers in the prediction of various policy uncertainty categories. Functionally, deep learning paradigms provide a significant advantage over BOW and SVM counterparts in the simplicity of multi-label classification implementation. Whereas traditional dictionary based methods like BOW require individual term sets for each category and SVMs are typically structured in a one-vs-rest fashion, neural network classification heads allow for simple multi-label outputs of N arbitrary dimensions. The data used in this work provide a secondary implicit experiment on performance in data scarce environments: as detailed in Table 2, category-specific labeled articles account for between 1.5% to 13.5% of the total 3,973 articles used for model development. To mitigate issues of mode collapse—classifier convergence to predicting only the most populous label, common in the

[17] Auditors were instructed to assign articles on a range from one to three on the certainty of label assignment. Given the lack of sufficient representation for the least certain category (one), the lower two certainties are aggregated to create a group of uncertain (184 unique articles) and certain (546 unique articles) when evaluating the rate of misclassification.

face of heavy class imbalance—I undertake an iterative sampling procedure for data splitting to ensure balance across all steps of model development and weight loss functions inverse proportionately to category prevalence. Both procedures are described in Appendix Section A.A3.

Table 2 provides accuracy and F1 scores for all three classifiers evaluated on policy uncertainty category prediction.[18] Notably, this table illustrates the critical role of selecting appropriate and holistic performance metrics, especially when working with large corpora and likely class imbalances. In the case of healthcare policy uncertainty, model performance appears equivalent when evaluating accuracy; however, F1 scores vary drastically across the three modeling approaches. While the SVM classifier fails to improve upon the BOW baseline resulting in differences of -11.79% and -43.95% in micro- and macro-averages respectively, the LLM counterpart exhibits performance increases of 101.09% and 67.53% in micro- and macro-averages.[19] In extreme cases, such as financial regulation, LLMs provide a relative increase in F1 score of 811% only having been trained and evaluated on 74 positively labeled articles. These increases in performance are likely lower-bounds of the potential of deep learning paradigms. All policy category predictions are generated from a single fine-tuned model, i.e. our model is forced to simultaneously learn the most important elements for predicting all seven categories of uncertainty simultaneously. Although less efficient, individual models for each category would likely further improve performance. Regardless of the chosen strategy for model development, these results illustrate the capacity of more sophisticated classifiers to significantly outperform status quo methods.

Table 2—Performance by Policy Category

Policy Category	# Positive Articles	Accuracy			F1 Score		
		BOW	SVM	LLM	BOW	SVM	LLM
Taxes	511	75.38	79.55	89.08	49.95	26.26	64.22
Healthcare	60	96.30	96.64	96.64	40.49	14.29	45.45
Trade Policy	460	87.36	79.27	87.96	29.10	15.91	59.81
Monetary Policy	350	71.76	86.27	87.39	33.21	28.99	52.63
Financial Regulation	74	31.97	96.36	94.40	4.11	0.00	37.50
Entitlement Programs	63	93.51	97.20	95.24	21.34	0.00	37.04
Fiscal Policy & Govt Spending	537	63.13	79.55	80.67	30.93	31.78	53.69

III. Conclusion

This study evaluates the potential of advanced NLP methods for extracting economically relevant data from nontraditional sources, namely text. Results from

[18] Appendix Table A6 provides the BOW keywords used for each category.

[19] Micro-averaging equally weights all observations and illustrates the average performance across all predictions. Macro-averaging takes the arithmetic mean of performance metrics calculated for each class independently, i.e. assigning equal weight to each category in the final determination of performance.

the EPU case study show a substantial performance increase using fine-tuned LLMs relative to both the dictionary-based methods that dominate the economic literature and machine learning classifiers like SVMs. Best performing LLMs exhibit 54% and 85% relative increases in F1 score and recall. A key critique of the LLM approach is the necessity of article text acquisition. I illustrate the capacity of LLMs fine-tuned on articles titles alone to outperform BOW classifiers applied to the full article text. Additionally, LLMs tuned on minimal information, akin to a brief summary or article heading, match the performance of the optimal SVM applied to full article texts, resulting in a 36% relative improvement in F1 score over the BOW baseline. The EPU case study demonstrate how LLMs better emulate human perceptions of uncertainty resulting in more robust numerical indices. I further show the potential for LLMs to predict multiple labels of policy uncertainty, unanimously outperforming BOW and SVM classifiers across all categories while providing a 101% relative improvement in micro-average F1 score.

Throughout this study, a particular emphasis is placed on democratizing AI within economics and, as such, model development and future application was designed for optimal accessibility. All models will be made freely available through the Huggingface Hub and via GitHub repository. Promising areas of future direction include the exploration of additional policy categories and temporal horizons, integrating joint prediction of sentiment alongside article classifications for richer indices, and exploring other architectures and tuning frameworks to improve classifier performance.

REFERENCES

Akiba, Takuya, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. "Optuna: A Next-generation Hyperparameter Optimization Framework." *arXiv.org*.

Arora, Abhishek, and Melissa Dell. 2024. "LinkTransformer: A Unified Package for Record Linkage with Transformer Language Models."

Azqueta-Gavaldn, Andrs, Dominik Hirschbhl, Luca Onorante, and Lorena Saiz. 2023. "Sources of Economic Policy Uncertainty in the euro area." *European economic review*, 152: 104373.

Baker, Scott R., Nicholas Bloom, and Steven J. Davis. 2016. "Measuring Economic Policy Uncertainty*." *The Quarterly Journal of Economics*, 131(4): 1593–1636.

Beltagy, Iz, Matthew E. Peters, and Arman Cohan. 2020. "Longformer: The Long-Document Transformer." *arXiv.org*.

Benguria, Felipe, Jaerim Choi, Deborah L. Swenson, and Mingzhi (Jimmy) Xu. 2022. "Anxiety or pain? The impact of tariffs and

uncertainty on Chinese firms in the trade war." *Journal of International Economics*, 137: 103608.

Bryan, Tom, Jacob Carlson, Abhishek Arora, and Melissa Dell. 2023. "EfficientOCR: An Extensible, Open-Source Package for Efficiently Digitizing World Knowledge."

Bybee, J. Leland. 2023. "The Ghost in the Machine: Generating Beliefs with Large Language Models."

Charemza, Wojciech, Svetlana Makarova, and Krzysztof Rybiski. 2022. "Economic uncertainty and natural language processing; The case of Russia." *Economic analysis and policy*, 73: 546–562.

Charemza, Wojciech, Svetlana Makarova, and Krzysztof Rybiski. 2023. "Anti-pandemic restrictions, uncertainty and sentiment in seven countries." *Economic Change and Restructuring*, 56(1): 1–27.

Dell, Melissa, Jacob Carlson, Tom Bryan, Emily Silcock, Abhishek Arora, Zejiang Shen, Luca D'Amico-Wong, Quan Le, Pablo Querubin, and Leander Heldring. 2023. "American Stories: A Large-Scale Structured Text Dataset of Historical U.S. Newspapers."

Dettmers, Tim, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. "QLoRA: Efficient Finetuning of Quantized LLMs."

Devlin, Jacob, Chang Ming-Wei, Kenton Lee, and Kristina Toutanova. 2019. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *arXiv.org*.

Du, Tianyu, Ayush Kanodia, Herman Brunborg, Keyon Vafa, and Susan Athey. 2024. "LABOR-LLM: Language-Based Occupational Representations with Large Language Models."

Gentzkow, Matthew, and Jesse M. Shapiro. 2010. "What Drives Media Slant? Evidence From U.S. Daily Newspapers." *Econometrica*, 78(1): 35–71.

Gentzkow, Matthew, Bryan Kelly, and Matt Taddy. 2019. "Text as Data." *Journal of Economic Literature*, 57(3): 53574.

Hu, Edward J., Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. "LoRA: Low-Rank Adaptation of Large Language Models."

IDC. 2019. "How IDCs Industry CloudPath SaaSPath Surveys Can Inform Your Cloud/SaaS Strategy."

Kalajdzievski, Damjan. 2023. "A Rank Stabilization Scaling Factor for Fine-Tuning with LoRA."

Keith, Katherine A., Christoph Teichmann, Brendan O'Connor, and Edgar Meij. 2020. "Uncertainty over Uncertainty: Investigating the Assumptions, Annotations, and Text Measurements of Economic Policy Uncertainty." *arXiv.org*.

Llama Team, AI @ Meta. 2024. "The Llama 3 Herd of Models."

Loughran, Tim, and Bill McDonald. 2011. "When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks." *The Journal of finance (New York)*, 66(1): 35–65.

Mangrulkar, Sourab, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. "PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods." *https://github.com/huggingface/peft*.

Noailly, Joelle, Laura Nowzohour, and Matthias van den Heuvel. 2022. "Does Environmental Policy Uncertainty Hinder Investments Towards a Low-Carbon Economy?"

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. "Scikit-learn: Machine Learning in Python." *Journal of Machine Learning Research*, 12: 2825–2830.

Qureshi, Shafiullah, Ba Chu, Fanny S. Demers, and Michel Demers. 2022. "Using Natural Language Processing to Measure COVID19-Induced Economic Policy Uncertainty for Canada and the US."

Sechidis, Konstantinos, Grigorios Tsoumakas, and Ioannis Vlahavas. 2011. "On the Stratification of Multi-label Data." 145–158. Berlin, Heidelberg:Springer Berlin Heidelberg.

Shen, Zejiang, Kaixuan Zhang, and Melissa Dell. 2020. "A Large Dataset of Historical Japanese Documents with Complex Layouts."

Tang, Eric, Bangding Yang, and Xingyou Song. 2024. "Understanding LLM Embeddings for Regression."

Tobback, Ellen, Hans Naudts, Walter Daelemans, Enric Junqu de Fortuny, and David Martens. 2018. "Belgian economic policy uncertainty index: Improvement through text mining." *International journal of forecasting*, 34(2): 355–365.

Trust, Paul, Ahmed Zahran, and Rosane Minghim. 2023. "Understanding the influence of news on society decision making: application to economic policy uncertainty." *Neural computing applications*, 35(20): 14929–14945.

TURING, A. M. 1950. "I.COMPUTING MACHINERY AND INTELLIGENCE." *Mind*, LIX(236): 433–460.

Vafa, Keyon, Emil Palikot, Tianyu Du, Ayush Kanodia, Susan Athey, and David Blei. 2024. "CAREER: A Foundation Model for Labor Sequence Data." *Transactions on Machine Learning Research*.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." *arXiv.org*.

Weizenbaum, Joseph. 1966. "ELIZAa computer program for the study of natural language communication between man and machine." *Commun. ACM*, 9(1): 3645.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rmi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. "HuggingFace's Transformers: State-of-the-art Natural Language Processing."

Appendix

A1. Training & Hyperparameter Optimization

Three LLMs, each with with varying attributes and advantages, are assessed in this study, with the model offering the best trade-off between performance and computational efficiency selected for main-body results. BERT is a lightweight transformer-based model designed to capture bidirectional relationships in text and has been used in numerous contexts including fine-tuning for sentiment analysis and text classification (Devlin et al., 2019). The key limitation of BERT is the inability to model long sequences: BERT's maximum context length is 512 tokens which is typically insufficient to cover longer newspaper articles. For example, 75% of all articles used in this study exceed the maximum 512 token limit. Longformer, in contrast, was specifically designed to accommodate long sequences while ensuring computational feasibility through the implementation of attention mechanisms which scale linearly with context length (Beltagy, Peters and Cohan, 2020). Meta's Llama 3 is among the newest entrants in the LLM sphere and provides open-source architectures of 8, 70, and 405 billion parameters (Llama Team, 2024). This work adopts Llama 3 8B to minimize computational requirements for model training. A notable advantage of the latter models are their capacity to process significantly longer sequences along with the general performance increase one would expect with higher parameter models, given sufficient training data.

Both Longformer and Llama 3 are evaluated at 32, 128, 256, 512, 1024, and 2048 tokens while BERT is evaluated at identical intervals up to the maximum length of 512 tokens. In the context of newspaper articles, the relative information gain from context lengths greater than 2048 tokens is minimal with only 10% of all articles in this dataset exceeding this length. However, use cases exploring economic uncertainty from technical reports or other documents may benefit from the use of Llama 3's increased context window of 128K tokens. Model hyperparameters are optimized using Optuna's Tree-structured Parzen Estimator (TPE) (Akiba et al., 2019).[20] The TPE sampler fits a Gaussian Mixture Model (GMM) to both the set of hyperparameter values of the minimum observed loss ($f(x)$) and the remaining values ($g(x)$). For each trial, hyperparameters are selected to maximize the ratio of $f(x)$ and $g(x)$ while the model is trained to minimize the binary cross-entropy loss characterized by Equation A1 where y_i is the true label of article i and p_i is the probability of predicting $y_i = 1$.

$$BCE = -\frac{1}{N}\sum_{i=1}^{N}\big(y_i \cdot log(p_i) + (1 - y_i) \cdot log(1 - p_i)\big) \tag{A1}$$

[20]Hyperparameters are components dictated by the model developer which impact the model fitting procedure. The following hyperparameters are uniformly sampled and optimized for all model and context length combinations: learning rate [1e-8, 1e-4], weight decay [1e-5, 0.9], gradient clipping [0, 1], warmup ratio [0, 0.1], and dropout probability [0, 0.9].

In contrast to EPU classification, classifying various policy categories is a multi-label task in which articles are simultaneously and probabilistically assigned to C categories. To account for heavy class imbalance, Equation A1 is weighted by the inverse proportion of positive instances q of all categories C as defined by Equation A2. This approach effectively equally weights all classes within the loss calculation regardless of proportional representation within the data, resulting in a more stable and efficient training process. More details on multi-label classification can be found in Appendix Section A.A3.

$$BCE_w = \sum_{q=1}^{C} \left(-\frac{1}{N} \sum_{i=1}^{N} \left(q \cdot y_{i,q} \cdot log(p_{i,q}) + (1 - y_{i,q}) \cdot log(1 - p_{i,q}) \right) \right) \tag{A2}$$

For each context length and model, a minimum of 576 gpu training hours or 15 individual trials (models) are trained within the Huggingface trainer environment. Huggingface is a collaborative web-based service which allows for interfacing and hosting numerous ML architectures in personal development spaces without requiring direct acquisition of source code and in many instances model weights (Wolf et al., 2020). Once trained on a personal machine, these models can be hosted and freely accessed by any active Huggingface user within their own development environment with minimal storage and computational requirements. The hyperparameters (trial) with the minimum validation loss for each model-context length combination is selected as optimal and will be made freely available, with proper attribution, in just a handful of lines of code within the Huggingface Hub. See Figure A1 for a Python implementation example of the Huggingface prediction pipeline. Additional details on the data curation, cleaning, and splitting procedure, as well as the scripts used for model training and evaluation are available upon reasonable request and will be made publicly available via GitHub repository at the time of publication.

```python
from datasets import load_dataset, Dataset, DatasetDict
from transformers import AutoTokenizer, AutoModelForSequenceClassification,
                                        Trainer, TrainingArguments,
                                       DataCollatorWithPadding

# Step 1: Load the dataset from a JSON file
data = load_dataset("json", data_files="Data.json")

# Step 2: Create a Dataset object from the raw data
dataset = Dataset.from_dict({"text": data['text'], "label": data['label']})

# Step 3: Wrap the dataset in a DatasetDict to standardize the structure
dataset = DatasetDict({"data": dataset})

# Step 4: Load the pre-trained Longformer tokenizer
tokenizer = AutoTokenizer.from_pretrained('allenai/longformer-base-4096')

# Step 5: Define a preprocessing function that tokenizes the text
def preprocess(examples):
    """
    Tokenizes the input text and truncates to the max length.
    Args:
        examples: Input examples containing text to tokenize.
    Returns:
        dict: Tokenized output.
    """
    return tokenizer(examples['text'], truncation=True, max_length=
                                           sequence_length)

# Step 6: Apply the preprocessing function to the entire dataset
dataset = dataset.map(preprocess, batched=True)

# Step 7: Load the pre-trained Longformer model for sequence classification
                                        (this can be updated to load any
                                       trained model)
model = AutoModelForSequenceClassification.from_pretrained('allenai/
                                       longformer-base-4096')

# Step 8: Define the data collator to handle padding during training
data_collator = DataCollatorWithPadding(
        tokenizer=tokenizer,
        padding='longest',
    )

# Step 9: Define the training arguments (configuration for the Trainer)
training_args = TrainingArguments(
        output_dir = save_dir,
        per_device_train_batch_size=desired_batch_size,
        per_device_eval_batch_size=desired_batch_size,
        group_by_length=True
    )

# Step 10: Initialize the Trainer
trainer = Trainer(
        model=model,
        args=training_args,
        data_collator=data_collator
    )

# Step 11: Train the model (or run inference if you want predictions)
predictions = trainer.predict(dataset['data'])
```

Figure A1. Huggingface Python Implementation

A2. Parameter-Efficieny Fine-Tuning & Model Quantization

Larger models, such as Llama 3, require significantly more computational resources to train and can be prohibitively costly even when utilizing modern compute clusters. To limit the computational overhead of tuning such models, I use Low-Rank Adaptation (LoRA) to reduce the trainable parameters of the model with minimal performance degradation. As introduced by Hu et al. (2021), LoRA utilizes rank decomposition of the original model weights (matrices) to eliminate potential redundancies and decrease the overall number of weights that must be tuned.[21] The LoRA implementation within the PEFT library in Huggingface introduces a new set of hyperparameters which can be optimized: rank [8, 128], alpha [8, 256], and dropout [0, 0.9] (Mangrulkar et al., 2022). In reality, there are seemingly endless strategies one could conceive of tuning these models and some may outperform others for any number of reasons, however the specific implementation strategy is inconsequential to the objective and this is just one way, perhaps the most easily accessible way, of fine-tuning exceptionally powerful models.

A3. Multi-label Classification

In addition to classification of economic policy uncertainty, I train classifiers to simultaneously predict ten policy categories. Although the full range of categories includes topics like monetary policy, fiscal policy, government spending, taxes, entitlement programs, healthcare, financial regulation, energy and environment, trade policy, competition policy, and political conflict and leadership change, I focus on seven categories with well-established term sets for BOW comparisons. The related data are again sourced from Baker, Bloom and Davis (2016). If an article pertains to economic uncertainty, auditors were instructed to assign the most relevant policy topics from a list of 16 categories. I take articles labeled positively as relating to economic uncertainty and ascertain labels for the following policy categories: monetary policy, fiscal policy, taxes, entitlement programs, healthcare, financial regulation, and trade policy.[22]

These labels are only available for a subset of audit data and are heavily imbalanced. To address this, I use an iterative sampling procedure based on the algorithm introduced in Sechidis, Tsoumakas and Vlahavas (2011) which ensures identical proportional representation for each class in the training, validation, and testing set. I select all 2,973 articles with at least one positive classification for any policy category then randomly sample 1,000 articles with no positive labels to construct a dataset of 3,973 articles which is split into 70% training, 20% validation, and 10% test.

[21] Kalajdzievski (2023) suggest the inclusion of a scaling factor to mitigate performance degradation for higher-rank adapters, a suggestion which is incorporated in this work. Additional optimizations can be made using 4-bit quantization as discussed in Dettmers et al. (2023), but are left for another day.

[22] A single article may be labeled as relating to multiple policy categories.

A4. Model Selection

The "optimal" model selection criteria was based on a combination of computational efficiency and classifier performance. Table A1 provides detailed performance metrics for the best performing hyperparameters for each model and context length combination. Additionally, Figure A2 provides the Area Under the Receiver Operating Characteristic curves (AUROC) for all final models. The AUROC serves as a measure of the classifiers ability to distinguish between positive and negative classes: a value closer to 1 indicates high performance while a value of 0.5 suggests that the classifier performs no better than random chance. The fundamental takeaway from both of these exhibits is that, within the context of the experiments defined in this work, Longformer provides a comparable performance to Llama 3 while requiring considerably less compute resources. This isn't to say that there exists no set of hyperparameters and data that would result in superior performance from Llama 3 as it is likely any number of potential fine-tuned architectures could outperform the results presented in this work. Rather, the fact that this case study prioritizes accessibility and simplicity, coupled with a relatively small dataset, leads to Longformer being the most appropriate selection in this particular instance.

Following general model selection, I undergo two additional experiments regarding the overall tuning strategy and optimal context length. To test the value of attention mechanisms, i.e. learning the representations across text which are most valuable in classifying economic policy uncertainty, I evaluate two distinct model architectures. The first, and most simple strategy, takes the existing Longformer model and only trains the classification head. This method effectively estimates a linear model which takes as input the numerical representations of text constructed by Longformer and predicts the binary label of the text. In contrast, "Attention & Classification" as referenced in Figure A3 trains both the Longformer model weights and the classification head (i.e. the LLM is allowed to learn new relationships over which part of the text contributes most to text classification via tuning of attention mechanisms). Figure A3 depicts AUROC curves for both tuning strategies and reveals that the fully fine-tuned Longformer model unanimously outperforms the "Classification" only counterpart. As such, all results discussed henceforth, and all main text results, utilize a fine-tuning strategy which targets the classification head and attention mechanisms.

With regard to context length, results show that increasing context length has a diminishing marginal, but positive, effect on overall classifier performance. This is evidenced by the increasingly outward bowing of AUROC curves associated with larger context windows and can also be seen in the monotonically increasing F1 scores of Table A1. Furthermore, Table A2 provides performance metrics calculated on the held out testing set, segmented by articles that fall above and below the maximum context length. Articles exceeding the maximum length considered (i.e. those that lose information during truncation) tend to exhibit lower performance relative to all other articles. The diminishing performance

gap with longer context lengths is likely attributable to the greater volume of information considered prior to truncation—once a critical mass of information (tokens) is obtained, the marginal value of adding more decreases significantly.

Table A1—LLM Comparison

BERT					
Context Length	Accuracy	Precision	Recall	F1-Score	AUROC
32 (Title)	60.44	57.79	43.97	49.94	0.65
128	67.60	65.99	65.59	65.79	0.75
256	73.27	71.60	72.47	72.03	0.79
512	74.04	71.37	75.71	73.48	0.81
Longformer					
Context Length	Accuracy	Precision	Recall	F1-Score	AUROC
32 (Title)	61.61	57.96	52.59	55.14	0.65
128	70.10	69.51	65.99	67.71	0.77
256	70.38	68.90	68.62	68.76	0.78
512	74.23	74.35	69.84	72.03	0.82
1024	75.38	71.40	80.36	75.62	0.84
2048	76.63	73.81	78.74	76.20	0.85
Llama 3					
Context Length	Accuracy	Precision	Recall	F1-Score	AUROC
32 (Title)	63.83	66.67	38.79	49.05	0.66
128	68.75	72.18	55.67	62.86	0.75
256	71.92	73.27	64.37	68.53	0.79
512	75.10	76.29	69.03	72.48	0.82
1024	74.90	68.70	86.64	76.63	0.83
2048	76.25	71.55	83.00	76.85	0.86

Note:

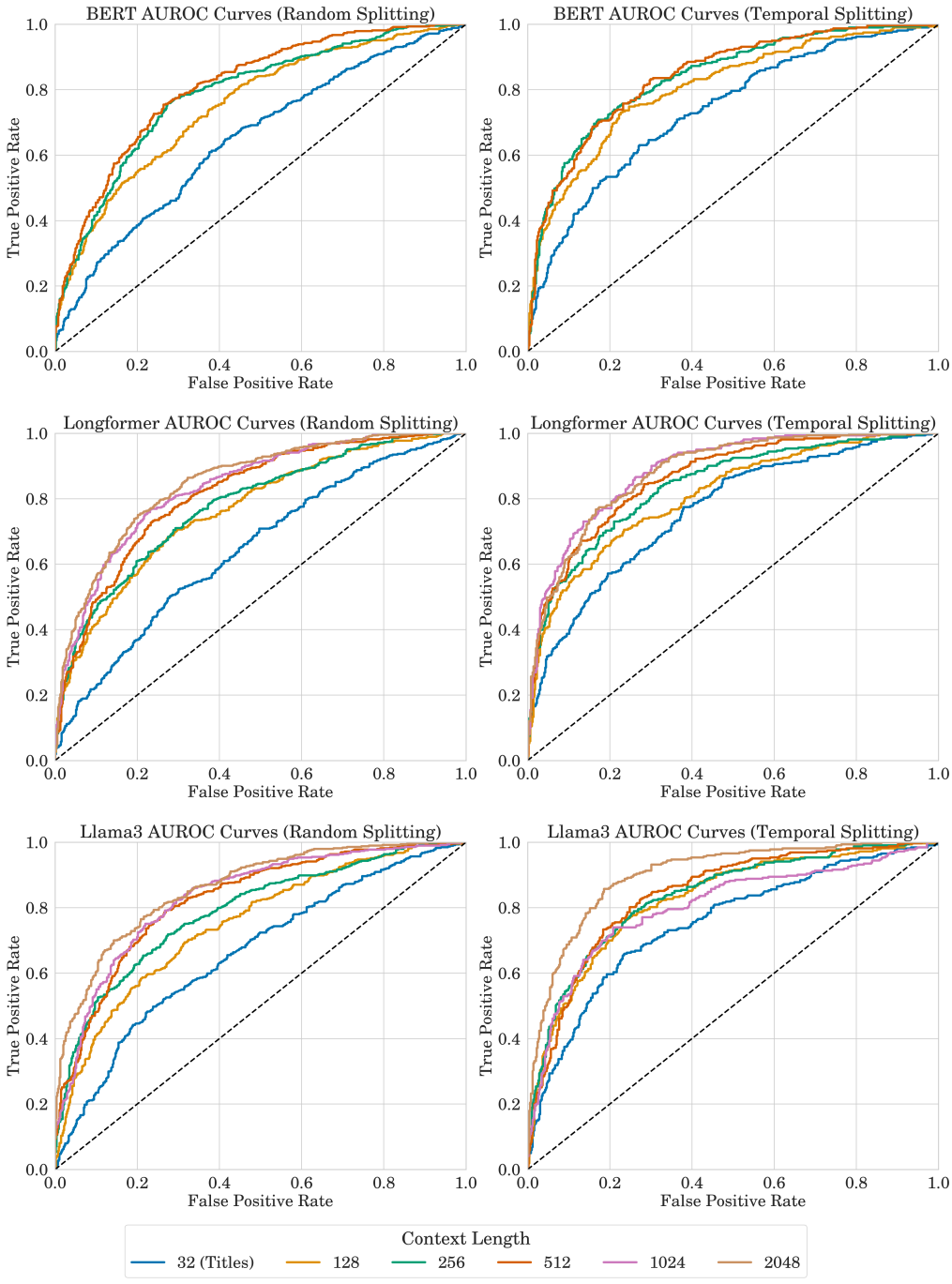


FIGURE A2. ALL AUROC CURVES FOR MODEL SELECTION

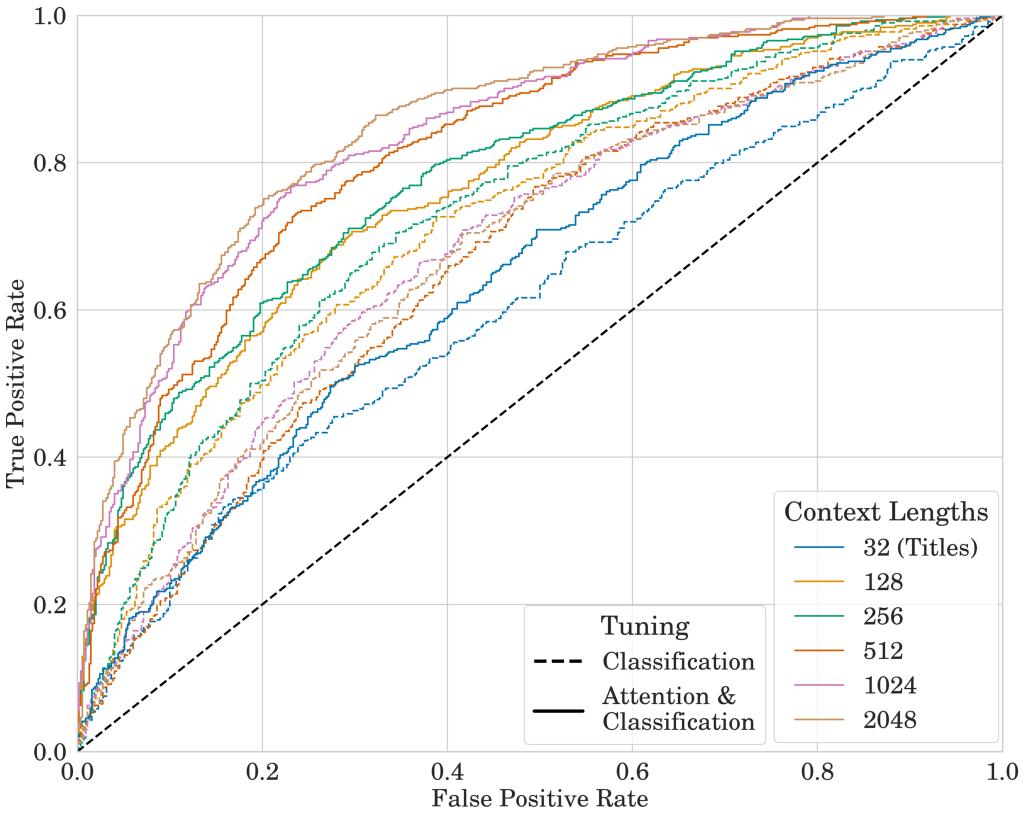

True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Context Lengths
32 (Titles)
128
256
512
1024
2048
Tuning
Classification
Attention &
Classification

FIGURE A3.

Table A2—Performance by Context Length

Max Length	Exceeds Max Length	Accuracy	Precision	Recall	F1 Score
128		100.00	100.00	100.00	100.00
	x	70.07	69.44	65.92	67.64
256		79.63	85.71	77.42	81.36
	x	69.88	67.89	68.03	67.96
512		79.50	77.46	76.39	76.92
	x	73.27	73.79	68.72	71.17
1024		76.42	70.10	84.40	76.59
	x	74.24	72.94	76.23	74.55
2048		76.91	73.73	80.00	76.74
	x	74.60	74.55	69.49	71.93

Notes: Longformer performance on the held-out test set split by within and exceeding the maximum 2048 context length. 12.12% of test set articles exceed the maximum context length.

Table A3—Longformer Performance by Splitting

Random Splitting					
	Accuracy	Precision	Recall	F1-Score	AUROC
32 (Title)	61.61	57.96	52.59	55.14	0.65
128	70.10	69.51	65.99	67.71	0.77
256	70.38	68.90	68.62	68.76	0.78
512	74.23	74.35	69.84	72.03	0.82
1024	75.38	71.40	80.36	75.62	0.84
2048	76.63	73.81	78.74	76.20	0.85
Temporal Splitting					
	Accuracy	Precision	Recall	F1-Score	AUROC
32 (Title)	70.73	64.44	57.19	60.60	0.76
128	74.33	68.17	65.63	66.88	0.81
256	74.57	65.33	75.85	70.20	0.84
512	77.26	68.27	79.26	73.35	0.86
1024	79.95	74.16	75.54	74.85	0.89
2048	77.38	66.91	84.52	74.69	0.88

Note:

Table A4—Model Performance Summary

BOW				
Context Length	Accuracy	Precision	Recall	F1-Score
Full Article	61.69	59.47	42.59	49.63
Full Article*	61.44	61.16	35.62	45.02
SVM				
Context Length	Accuracy	Precision	Recall	F1-Score
32 (Title)	58.41	54.43	45.04	49.29
Full Article	70.96	72.02	63.56	67.53
LLM				
Context Length	Accuracy	Precision	Recall	F1-Score
32 (Title)	61.61	57.96	52.59	55.14
128	70.10	69.51	65.99	67.71
256	70.38	68.90	68.62	68.76
512	74.23	74.35	69.84	72.03
1024	75.38	71.40	80.36	75.62
2048	76.63	73.81	78.74	76.20

Notes: Key performance metrics of key classifiers. All AI specification report performance on a held out test set of 1.041 articles using Longformer. Metrics for BOW and BOW Strict are computed on the full audit sample of 10,394 articles. * indicates no additional text cleaning.

Table A5—EPU Index Correlation

	Audit	BOW	LLM (Binary)	LLM (Probabilistic)
Audit	1			
BOW	0.24	1		
LLM (Binary)	0.69	0.43	1	
LLM (Probabilistic)	0.74	0.52	0.96	1

Notes:

TABLE A6—POLICY CATEGORY KEYWORDS

Category	**Keywords**
Monetary Policy	federal reserve, the fed, money supply, open market operations, quantitative easing, monetary policy, fed funds rate, overnight lending rate, bernanke, volcker, greenspan, central bank, interest rates, fed chairman, fed chair, lender of last resort, discount window, european central bank, ecb, bank of england, bank of japan, boj, bank of china, bundesbank, bank of france, bank of italy
Fiscal Policy & Gov't Spending	taxes, tax, taxation, taxed, government spending, federal budget, budget battle, balanced budget, defense spending, military spending, entitlement spending, fiscal stimulus, budget deficit, federal debt, national debt, Gramm-Rudman, debt ceiling, fiscal footing, government deficits, balance the budget
Taxes	taxes, tax, taxation, taxed
Entitlement Programs	entitlement program, entitlement spending, government entitlements, social security, medicaid, medicare, government welfare, welfare reform, unemployment insurance, unemployment benefits, food stamps, afdc, tanf, wic program, disability insurance, part d, oasdi, supplemental nutrition assistance program, earned income tax credit, eitc, head start program, public assistance, government subsidized housing
Healthcare	health care, medicaid, medicare, health insurance, malpractice tort reform, malpractice reform, prescription drugs, drug policy, food and drug administration, fda, medical malpractice, prescription drug act, medical insurance reform, medical liability, part d, affordable care act, obamacare
Financial Regulation	banking supervision, glass-steagall, tarp, bank supervision, thrift supervision, dodd-frank, financial reform, commodity futures trading commission, cftc, house financial services committee, basel, capital requirement, volcker rule, bank stress test, securities and exchange commission, sec, deposit insurance, fdic, fslic, ots, occ, firrea, truth in lending
Trade Policy	import tariffs, import duty, import barrier, government subsidies, government subsidy, wto, world trade organization, trade treaty, trade agreement, trade policy, trade act, doha round, uruguay round, gatt, dumping

Note: This table lists keywords associated with various policy categories.

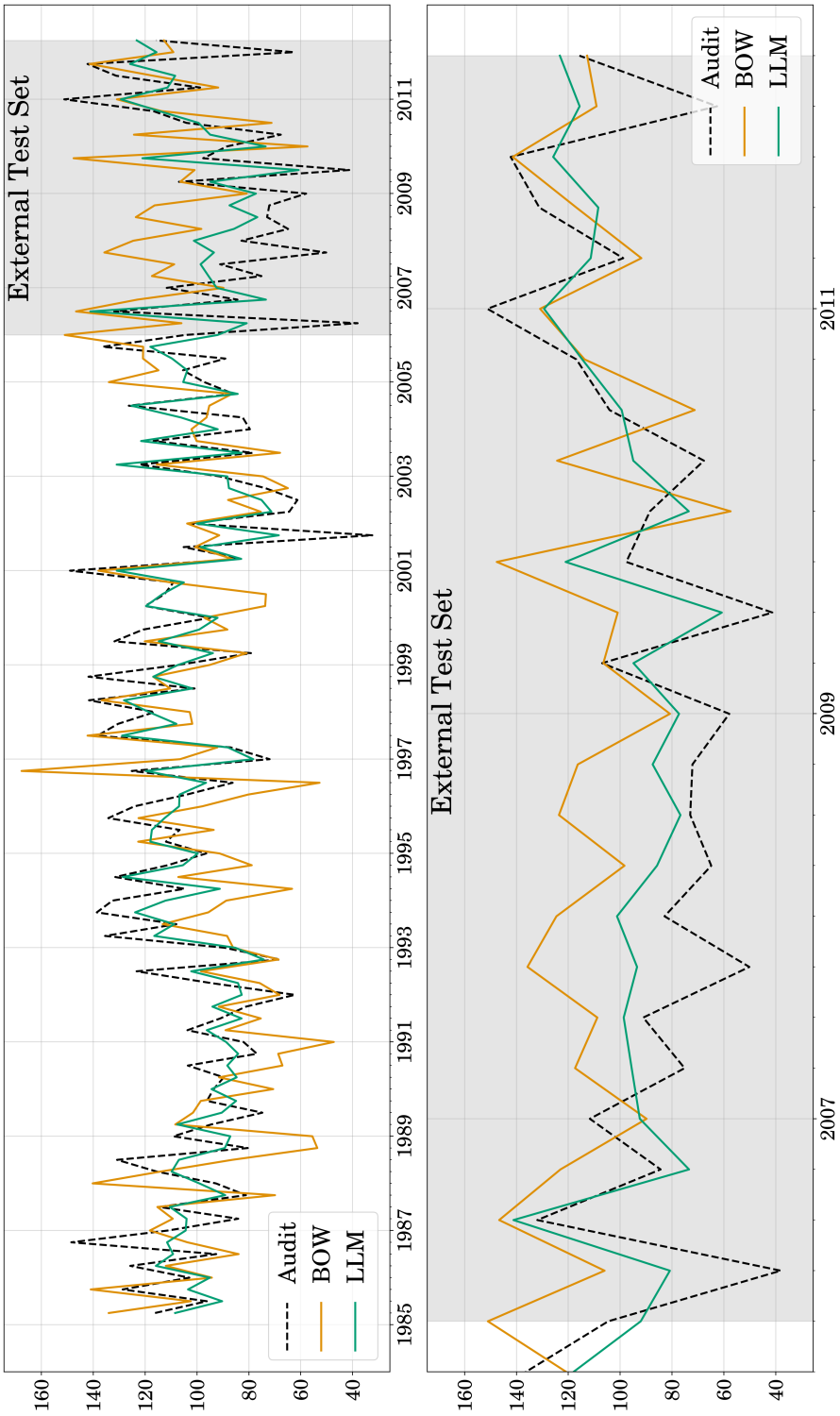


Figure A4. Temporal Index

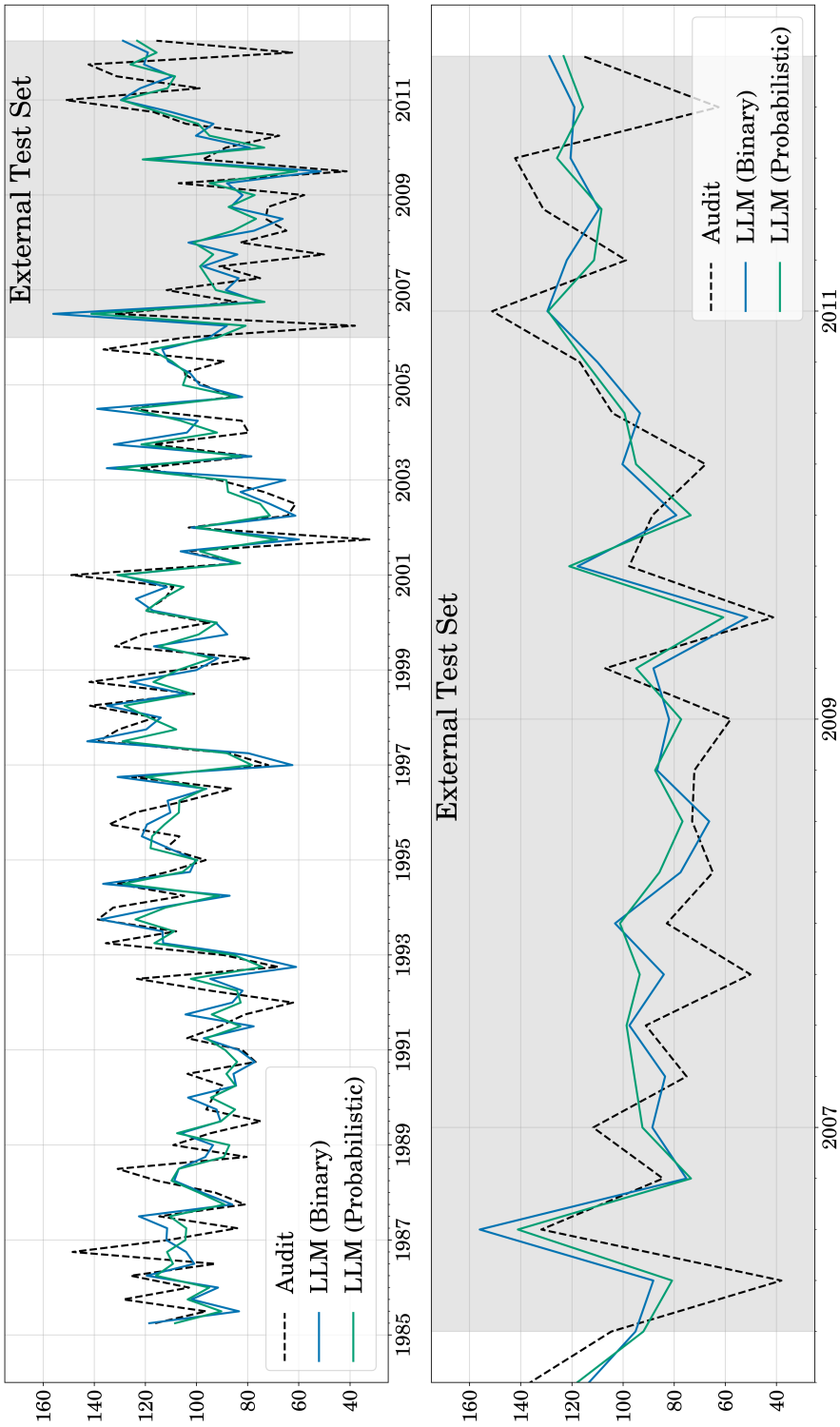


Figure A5. AI EPU Index Comparison