Optimal Text-Based Time-Series Indices

David Ardia[a], Keven Bluteau[b,*]

[a] *CIRANO & GERAD & Department of Decision Sciences, HEC Montréal, Montréal, Canada*
[b] *Department of Finance, Université de Sherbrooke, Canada*

Abstract

We propose an approach to construct text-based time-series indices in an optimal way—typically, indices that maximize the contemporaneous relation or the predictive performance with respect to a target variable, such as inflation. Our methodology relies on binary selection matrices that, applied to the vocabulary of tokens, select the relevant texts in the corpus. Various widely-known text-based indices, such as the Economic Policy Uncertainty (EPU) index, can be formulated in terms of selection matrices. We design a genetic algorithm with domain-specific knowledge featuring tailor-made crossover and mutation operations to perform the complex optimization. We illustrate our methodology with a corpus of news articles from the Wall Street Journal by optimizing text-based indices that forecast inflation at various horizons.

Keywords: Genetic algorithm; text-based indices; NLP; text-mining; inflation; Sentometrics.

*Corresponding author.
Email addresses: david.ardia@hec.ca (David Ardia), keven.bluteau@usherbrooke.ca (Keven Bluteau)

1. Introduction

In economic and financial research, there is a growing trend of integrating textual data such as news articles into econometrics analysis (see Gentzkow et al., 2019, for a review). This integration is typically done by (i) selecting, (ii) transforming, and (iii) aggregating textual content into a time-series representation (see Ardia et al., 2019; Algaba et al., 2020, for a general overview of these steps). While many studies have focused on steps (ii) and (iii)—transforming and aggregating textual data into a quantitative measure such as sentiment (see *e.g.*, Loughran and McDonald, 2014; Jegadeesh and Wu, 2013; Manela and Moreira, 2017; Consoli et al., 2022; Barbaglia et al., 2023)—the essential selection step (i), which usually relies on subjective ad-hoc rules, has not received much attention yet.

We aim to fill this gap in this article by proposing an approach to construct text-based time-series indices optimally. Specifically, our algorithm determines which set of texts, among a large corpus, leads to a text-based index that is optimal for a specific objective—typically, an attention-based index that maximizes the contemporaneous relation or the predictive performance with respect to a target variable, such as inflation. Our methodology relies on binary *selection matrices* that, applied to the vocabulary of tokens, select the relevant texts in the corpus. Various widely-known text-based indices, such as the Economic Policy Uncertainty (EPU) index by Baker et al. (2016), can be formulated in terms of selection matrices.

Optimizing selection matrices is challenging due to the inherent non-linearity that arises when aggregating selected texts into text-based indices. To overcome this difficulty, we design a genetic algorithm with domain-specific knowledge to explore the solution space and obtain an *optimal* selection matrix. The algorithm starts with an initial population of selection matrices. These matrices are evaluated using a fitness function that measures the resulting textual-index performance in achieving the objective of selecting the texts. At each iteration, the population of selection matrices undergoes tailor-made crossover and mutation operations, as traditional operators are not well suited to this optimization problem. We also implement additional steps to address potential overfitting issues and leverage the information in word embeddings of promising solutions to help explore good solutions more efficiently. Finally, we introduce a pruning step to avoid sub-optimal solutions.

To showcase the relevance of our methodology, we conduct two empirical applications using a collection of 837,576 Wall Street Journal news articles spanning from January 2000 to August 2024.

First, we validate our methodology with the EPU index. Specifically, we use our algorithm to see if we can recover the set of keywords proposed by Baker et al. (2016) when applying their EPU index's construction to our corpus. Given the number of tokens in our corpus vocabulary,

recovering the set of tokens for the three dictionary dimensions of Baker et al. (2016) is not trivial. We show that our approach (i) can recover the set of keywords proposed by the authors—their selection matrix—and (ii) does so in a reasonable computational time.

Next, building on Eugster and Uhl (2024), we optimize selection matrices to generate attention-based indices from negative-news articles that have a positive (negative) relation with U.S. inflation, conditional on whether inflation is above (below) the U.S. central bank 's 2% target. This empirical setting demonstrates the flexibility of our approach and presents a challenging task that underscores the utility of our proposed optimization strategy. Our inflation forecasting results indicate that incorporating optimized negative-news attention-based indices to forecasting models significantly improves the out-of-sample prediction accuracy across various forecasting horizons.

In addition, we validate our optimization strategy by demonstrating that imposing a penalty for excessive sparsity—that is, penalizing solutions where fewer than a certain proportion of news articles are selected at each point in time—improves out-of-sample performance. We also underscore the interpretability of our approach by analyzing the articles selected by the optimized selection matrices through a topic model. Our findings reveal that, for forecasting inflation, the selection matrices select news articles focusing on topics and themes—such as monetary policy, investments, oil, and the bond market—that are plausibly linked to key drivers of future inflation.

The rest of this paper is organized as follows. Section 2 introduces the notation, presents the concept of selection matrices, and outlines the optimization problem. Section 3 presents our optimization strategy, including the genetic algorithm and the proposed crossover and mutation operators. Section 4 presents our corpus and an empirical validation of our algorithm with the EPU. Section 5 presents our empirical application on inflation forecasting, and Section 6 concludes.

2. Token-Based Text Selection

We first introduce the concept of tokens and vocabulary to analyze a corpus of texts. A token, denoted by v, represents a sequence of characters, including acronyms, words, sequences of words, or even regular expressions. The vocabulary of size V is defined as a collection of such tokens.

The text corpus is represented as a matrix $\boldsymbol{C}_t$ of size $N_t \times V$, where N_t is the number of texts available at a given time t. Each element $c_{n,v,t}$ among a collection of matrices $\boldsymbol{C}_t$, where $t = 1, \ldots, T$, indicates if the token v appears in the text n published at time t. If it does, the element takes the value of one and zero otherwise. Hence, a specific text published at time t

can be represented by the row vector $\boldsymbol{c}_{n,t}$ (of size $1 \times V$).

Typically, the corpus consists of a vast collection of texts, and our objective is to select texts for further analysis. For instance, we could focus on texts related to the U.S. economy from a collection of news articles published by various newspapers. A simple way to proceed is by using a keyword-based (*i.e.*, token-based in our nomenclature) approach, which we formalize below.

We employ a selection matrix $\boldsymbol{\Omega}$ of size $V \times K$, where each column vector $\boldsymbol{\omega}^k$ (of size $V \times 1$) corresponds to a selection rule. The binary selection vector $\boldsymbol{\kappa}_t$ of size N_t contains elements $\kappa_{n,t}$, which are defined through the selection function $f_\kappa(\cdot)$ as:

$$\kappa_{n,t} \equiv f_\kappa(\boldsymbol{\Omega}, \boldsymbol{c}_{n,t}) \equiv I\left[\sum_{k=1}^{K} I\big[\boldsymbol{c}_{n,t}\boldsymbol{\omega}^k > 0\big] = \sum_{k=1}^{K} I\big[\sum_{v=1}^{V} \boldsymbol{i}_V \boldsymbol{\omega}^k > 0\big]\right], \tag{1}$$

where $I[\cdot]$ is an indicator function that takes a value of one if the condition inside the parenthese is true and zero otherwise.

In (1), $\sum_{k=1}^{K} I\big[\boldsymbol{c}_{n,t}\boldsymbol{\omega}^k > 0\big]$, counts the number of unique active tokens (*i.e.*, non-zero value) for dimension k of the selection matrix $\boldsymbol{\Omega}$ (represented by $\boldsymbol{\omega}^k$) that appear in the text n at time t (represented by row vector $\boldsymbol{c}_{n,t}$). If this count is greater than zero for all dimensions $k = 1, \ldots, K$, each element of the sum takes the value of one, and the summation takes the value K. The second part, $\sum_{k=1}^{K} I\big[\sum_{v=1}^{V} \boldsymbol{i}_V \boldsymbol{\omega}^k > 0\big]$, where $\boldsymbol{i}_V$ is a row vector of lenght V where each element is equal to one, counts the number of active tokens within each dimension of the selection matrix $\boldsymbol{\Omega}$. If this count is greater than zero for all dimensions, each element of the sum takes value one, and the sum is K. Thus, if the text n published at time t contains at least one active token within each non-zero column k of $\boldsymbol{\Omega}$, the selection condition is satisfied, and $\kappa_{n,t}$ is equal to one.

The well-known Economic Policy Uncertainty (EPU) index of Baker et al. (2016) relies on this selection function. The EPU selection condition is defined by a vocabulary of size 10 with three selection dimensions ($V = 10$ and $K = 3$) representing economic, policy, and uncertainty-related tokens. Using our notation, the selection matrix $\boldsymbol{\Omega}$ (of size 10×3) of the EPU index

is:

$$\boldsymbol{\Omega}_{\text{EPU}} \equiv \begin{array}{c} \\ \text{economic} \\ \text{economy} \\ \text{congress} \\ \text{deficit} \\ \text{federal\_reserve} \\ \text{legislation} \\ \text{regulation} \\ \text{white\_house} \\ \text{uncertain} \\ \text{uncertainty} \end{array} \begin{array}{c} \begin{array}{ccc} \text{Economy} & \text{Policy} & \text{Uncertainty} \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \end{array}.$$

The EPU selection matrix contains 2, 6, and 2 active tokens in the first, second, and third dimension, respectively. In that case, the second terms of (1): $\sum_{k=1}^{K} I\big[\sum_{v=1}^{V} \boldsymbol{i}_V \boldsymbol{\omega}_{\text{EPU}}^{k} > 0\big] = 3$. For a text n published at time t containing the tokens "economic," "congress," and "uncertainty," the first terms of (1): $\sum_{k=1}^{K} I\big[\boldsymbol{c}_{n,t}\boldsymbol{\omega}_{\text{EPU}}^{k} > 0\big] = 3$. Given the equality of both terms, for this article, $\kappa_{n,t} = 1$. Alternatively, a text n published at time t containing two times the token "economic" and one time the token "congress," the first terms of (1), $\sum_{k=1}^{K} I\big[\boldsymbol{c}_{n,t}\boldsymbol{\omega}_{\text{EPU}}^{k} > 0\big] = 2$, and thus $\kappa_{n,t} = 0$ as both the left and right terms of (1) are not equal.

Several news-media-based indices follow similar selection rules: the Climate Policy Uncertainty index of Gavriilidis (2021), the Equity Market Volatility index of Baker et al. (2019), the Monetary Policy Uncertainty index of Husted et al. (2020), and the Trade Policy Uncertainty index of Handley and Limão (2022).[1]

2.1. Downstream Transformation

After identifying relevant texts in the corpus with the selection function $f_{\kappa}(\cdot)$, we typically transform and aggregate the selected texts into a quantitative time-series measure. We present below the attention-based measure.

[1]Keyword matching can sometimes lead to false positives, where an article is included in the index despite only briefly mentioning the relevant topic without being primarily about it. Our data-driven approach can serve as a first step in systematically quantifying the concept of interest and could be refined further in two ways. First, the selection function in (1) could take into account more complex selection mechanisms, such as the addition of a proximity condition between the tokens of two or more dimensions of the selection matrix $\boldsymbol{\Omega}$, as the one used to construct the Geopolitical Risk index of Caldara and Iacoviello (2022). For instance, in the case of the EPU index, one might require at least one token from the "Economy" dimension detected in the neighborhood (*e.g.*, within a two-token distance) of one of the "Policy" tokens in the same news article. Such additional complexity in the selection rule could be integrated using the information from a token distance matrix similar to the one used in Ardia et al. (2021). Second, topic modeling could better ensure that selected articles align with the intended concept, and human validation through random sample auditing could help assess and refine the accuracy of the index. We do not consider these additional steps in this paper to keep the exposition simple.

An attention measure aims to measure the level of importance attributed to the selected texts across all texts in the corpus at a given time. For instance, it could be to measure the level of importance given to the topic of climate change by the news media (see, *e.g.*, Ardia et al., 2023). Given our prior notation and definition, a typically used measure of attention is:

$$f_{\text{att}}(\boldsymbol{\Omega}, \boldsymbol{C}_t) \equiv \frac{1}{N_t} \sum_{n=1}^{N_t} f_\kappa(\boldsymbol{\Omega}, \boldsymbol{c}_{n,t}) \,. \tag{2}$$

At each time t, we measure how many texts are selected and normalize this number by the total number of texts in the corpus at that time. If the corpus is composed of several sources (*e.g.*, various newspapers), it may be relevant to standardize each source before the aggregation in (2), as in Baker et al. (2016). An alternative form of transformation is the content transformation, which requires more information than the output of the selection step. The content of selected texts is processed to derive a "score," such as polarity or sentiment (see, *e.g.* Algaba et al., 2020; Consoli et al., 2022; Barbaglia et al., 2023).[2] A practical advantage of employing the attention transformation over the content transformation lies in its reduced reliance on processing the entire corpus of texts through an additional layer of models to derive the scores (see, *e.g.*, Algaba et al., 2020; Consoli et al., 2022). For instance, when utilizing a news articles aggregator that consolidates the desired news sources, the data required for computing attention solely invoke queries derived from a selection matrix (to determine the count of selected articles) and statistical information on the number of published news articles from the desired sources. This streamlined approach facilitates the timely computation and updating of the attention index. See, for instance, `https://www.policyuncertainty.com/`, where multiple indices similar to the EPU are updated monthly.

2.2. Objective-Based Tokens Optimization

The objective of deriving information in texts by a selection function and a transformation process is to capture a (contemporaneous or predictive) relation between that information and

[2]Specifically, the content transformation function can be written as:

$$f_{\text{con}}(\boldsymbol{\Omega}, \boldsymbol{C}_t, \boldsymbol{\zeta}) \equiv \frac{1}{\sum_{n=1}^{N_t} f_\kappa(\boldsymbol{\Omega}, \boldsymbol{c}_{n,t})} \sum_{n=1}^{N_t} \left(f_\kappa(\boldsymbol{\Omega}, \boldsymbol{c}_{n,t}) \sum_{v=1}^{V} c_{n,v,t}\, \zeta_v \right) , \tag{3}$$

where ζ_v are token weights (integer or real numbers) used to score the selected texts. The average score of selected texts at time t is then used to obtain an index. Token weights can be measured from manually-composed lexicons (*e.g.*, Loughran and McDonald, 2014) or in a data-driven way (*e.g.*, Jegadeesh and Wu, 2013; Manela and Moreira, 2017; Kelly et al., 2021; Ardia et al., 2022; Consoli et al., 2022; Barbaglia et al., 2023). Much of the literature has focused on estimating $\boldsymbol{\zeta} \equiv (\zeta_1, \ldots, \zeta_V)$ given a predefined selection of texts. When using a data-driven method for content transformation, the joint estimation of the selection matrix $\boldsymbol{\Omega}$, and content transformation weights $\boldsymbol{\zeta}$, is preferable, albeit computationally intensive. Because of the added complexity of the content transformation, which is a challenge in this study, to isolate the effect of news selection from content transformation, we focus on the attention transformation measure.

a variable of interest y_t $(t = 1, \ldots, T)$. A selection matrix is typically based on a subjective but guided assessment of the tokens necessary to capture the information we want to extract from the texts. Formally, given a relevant attention-based transformation function $f_{\text{att}}(\boldsymbol{\Omega}, \boldsymbol{C}_t)$, the aim is to minimize:

$$\min_{\alpha,\beta,\boldsymbol{\gamma}} \frac{1}{T} \sum_{t=1}^{T} \left(y_t - (\alpha + \beta f_{\text{att}}(\boldsymbol{\Omega}, \boldsymbol{C}_{t-h}) + \boldsymbol{\gamma}' \boldsymbol{x}_{t-h}) \right)^2 , \tag{4}$$

with $h \geq 0$ and where $\boldsymbol{x}_{t-h}$ is a vector containing control variables, which may include lagged values of y_t. Note that we limit the presentation to linear specifications for simplicity, but our approach also extends to nonlinear specifications. Once estimated, one would typically make inference about parameter β or test whether the (out-of-sample) root-mean-squared error is significantly lower than the root-mean-squared error of a model where β is constrained to be zero (*i.e.*, nested model); see Ardia et al. (2019). For instance, several studies analyze the relationship between the EPU and economic variables using the following regression framework:[3]

$$\min_{\alpha,\beta,\boldsymbol{\gamma}} \frac{1}{T} \sum_{t=1}^{T} \left(y_t - \left(\alpha + \beta f_{\text{att}}(\boldsymbol{\Omega}_{\text{EPU}}, \boldsymbol{C}_{t-h}) + \boldsymbol{\gamma}' \boldsymbol{x}_{t-h}\right) \right)^2 . \tag{5}$$

One limitation of this approach is related to the fact that $\boldsymbol{\Omega}_{\text{EPU}}$ is given. This may be reasonable in some applications but suboptimal for others, particularly when it comes to nowcasting and forecasting. We thus introduce the following optimization problem:

$$\begin{aligned} \min_{\alpha,\beta,\boldsymbol{\gamma},\boldsymbol{\Omega}} \frac{1}{T} \sum_{t=1}^{T} \left(y_t - \left(\alpha + \beta f_{\text{att}}(\boldsymbol{\Omega}, \boldsymbol{C}_{t-h}) + \boldsymbol{\gamma}' \boldsymbol{x}_{t-h}\right) \right)^2 \\ + \frac{\lambda_1}{T} \sum_{t=1}^{T} I[f_{\text{att}}(\boldsymbol{\Omega}, \boldsymbol{C}_{t-h}) < \tau] \\ + \lambda_2 I[\beta > 0] + \lambda_3 I[\beta \leq 0] . \end{aligned} \tag{6}$$

In the optimization problem outlined in (6), we assume an unrestrictive and large vocabulary and optimize active tokens selection in addition to the regression parameters. We standardize y_t to avoid scale-dependent penalty parameters. We use three penalty terms to control the optimization process.

First, the term $\frac{\lambda_1}{T} \sum_{t=1}^{T} I[f_{\text{att}}(\boldsymbol{\Omega}, \boldsymbol{C}_{t-h}) < \tau]$ is a sparsity penalty that penalizes solutions that select fewer documents than a prescribed proportion τ of the total at any given time t. The penalty scales linearly with the frequency of this occurrence. For example, if a solution

[3] At the time of this writing, for instance, Baker et al. (2016), the research paper introducing the EPU index, has received over 9,000 citations according to Google Scholar. While many of those citations do not use the index per se, many, however, analyze how the EPU explains or interacts with other macroeconomics and financial variables.

selects fewer than $\tau\%$ of documents in half the time (*i.e.*, $0.5 \times T$ instances), then $0.5 \times \lambda_2$ is added to the objective function. This mechanism helps mitigate overfitting by discouraging the algorithm from focusing solely on narrow events that cause spikes in the target variable, rather than capturing a more robust underlying signal. Consequently, introducing this penalty fosters better out-of-sample generalization and enables more effective extraction of news-based signals in high noise-to-signal ratio environments.

The two remaining penalty terms, $\lambda_2 I[\beta > 0]$ and $\lambda_3 I[\beta \leq 0]$, serve to establish the intended association between the text-based index and the variable of interest y_t. When a negative relationship is sought, the condition $\lambda_2 > \lambda_3$ is applied. Consequently, this allows for controlling a selection condition that selects texts either positively or negatively correlated with the variable of interest. It is crucial to choose the sign of beta as selection matrices can be derived in a manner that selects texts associated negatively with the variable of interest and another set that is positively associated with it.

A LASSO penalty could also be considered in (6) (*e.g.*, $\lambda_4 ||vec(\mathbf{\Omega})||$). By varying the LASSO penalty, researchers could observe which potential tokens remain significant and which are eliminated. Such an analysis could shed light on the impact of specific tokens and their contribution to the overall forecasting performance, for instance. While very appealing conceptually, such an approach faces two challenges in practice. First, calibrating an additional LASSO penalty parameter introduces a substantial computational burden. Given the complex, multi-step nature of our optimization strategy (see Section 3), including another layer of parameter tuning would turn impractical in several real-life settings. Second, our approach involves a non-linear transformation in which the mapping from the selection matrix to the index value does not guarantee a direct, monotonic improvement in the in-sample fit when a new token is added (unlike a traditional linear model where adding a predictor typically shifts coefficients in a more predictable manner). Consequently, the expected benefits of a LASSO penalty—particularly in reducing overfitting—will not be as pronounced as in standard linear or generalized linear modeling frameworks.

In the presentation of our methodology and our empirical applications below, we consider a fixed (static) vocabulary of tokens—potentially encompassing all relevant words, bigrams, and trigrams—and, as such, fixed-size selection matrices. However, as new words and topics are discussed in the news over time, we could envision adapting the methodology to encompass a dynamic vocabulary, as suggested by a referee. This would require a mechanism to track and incorporate newly emerging terms, which implies continual updates to the vocabulary and the selection matrix. While conceptually feasible, we believe the design and implementation would be challenging. A first ad-hoc approach could be to run our current algorithm on a rolling

window basis. But already with this simple setup, we would face a massive computational burden. Thus, we prefer to leave these interesting extensions for further research.

3. Optimization Strategy

Regression (4) is trivial to estimate as $\mathbf{\Omega}$ is fixed and thus not part of the estimation process. On the contrary, the minimization problem (6) is complex as $f(\mathbf{\Omega}, \boldsymbol{C}_t)$ is a non-linear function of $\mathbf{\Omega}$, which is now considered as a parameter. Below, we present our strategy based on a genetic algorithm to perform the minimization. We first define the crossover and mutation operators specifically designed for our problem.

3.1. Crossover and Mutation Operators

Because of the distinctive characteristics of our optimization problem, we introduce specific crossover and mutation operators. To illustrate these operators, we consider a specific scenario wherein our optimization objective is to replicate the EPU index while imposing a constraint that restricts the number of active tokens to five out of a maximum of VK (*i.e.*, when all elements in the matrix $\mathbf{\Omega}$ are equal to one).

3.1.1. Token Crossover Operator

We begin with the "token crossover operator". This operator takes two parent solutions and generates two offspring solutions by exchanging a single token between the parents while maintaining the dimensionality of the replaced token. Consider the following two parents:

Parent 1:	1	2	3
economic	1	0	0
legislation	0	1	0
congress	0	1	0
house	0	0	1
risk	0	0	1
⋮	⋮	⋮	⋮

Parent 2:	1	2	3
economy	1	0	0
financial_crisis	0	1	0
congress	0	1	0
house	0	0	1
risk	0	0	1
⋮	⋮	⋮	⋮

The token crossover operator selects two tokens (in gray) and generates two offspring solu-

tions as follows:

Child 1	1	2	3
financial_crisis	1	0	0
legislation	0	1	0
congress	0	1	0
house	0	0	1
risk	0	0	1
⋮	⋮	⋮	⋮

Child 2	1	2	3
economy	1	0	0
economic	0	1	0
congress	0	1	0
house	0	0	1
risk	0	0	1
⋮	⋮	⋮	⋮

It is important to note that this crossover operator maintains a fixed number of active tokens within each dimension and operates solely on already activated tokens. In contrast, a regular crossover operator would not guarantee that each dimension remains the same size or even has active tokens. Additionally, owing to the high sparsity of our optimization problem (where only five tokens are active out of a potential VK points), a regular crossover operator could perform non-altering operations by exchanging slices of non-active tokens (*i.e.*, tokens with zero values), consequently reducing the efficiency of the search algorithm.

3.1.2. Mutation Operators

Next, we focus on three mutation operators, each serving a distinct purpose.

Switch Mutation Operator. The "switch mutation" operator allows an active token to change its dimension:

Parent	1	2	3
economy	1	0	0
economic	0	1	0
congress	0	1	0
house	0	0	1
risk	0	0	1
⋮	⋮	⋮	⋮

Child	1	2	3
economy	1	0	0
economic	1	0	0
congress	0	1	0
house	0	0	1
risk	0	0	1
⋮	⋮	⋮	⋮

This operator allows for the correction of potential dimensional misalignment. It is worth noting that in the example above, the set of texts selected by the parent, while not optimal, may be highly correlated with the set of texts of the child, which is optimal. This operation facilitates a more efficient search for misalignment compared to a random search. Similar to other operators, this mutation operator only applies to active tokens. However, it cannot change the dimensions of a token if it is the only token within its dimension, ensuring that each dimension has at least one active token.

N-Gram Mutation Operator. Next, we introduce the "n-gram mutation" operator, which involves transforming one of the selected tokens into an n-gram containing that token:

Parent	1	2	3
economy	1	0	0
economic	1	0	0
congress	0	1	0
house	0	0	1
risk	0	0	1
⋮	⋮	⋮	⋮

Child	1	2	3
economy	1	0	0
economic	1	0	0
congress	0	1	0
white_house	0	0	1
risk	0	0	1
⋮	⋮	⋮	⋮

To comprehend this operator, we must recognize that the texts containing an n-gram always include the texts containing its components. For instance, the number of texts containing the token "stock" is equal to or larger than the number of texts containing the bi-gram "stock_market." As such, if the algorithm selects the component of an n-gram, there is a high likelihood that using an n-gram containing that component would be more optimal. The n-gram mutation operator increases the possibility of testing an n-gram compared to testing any other token. Similar to the crossover operator, the n-gram mutation only applies to active tokens, but it also applies solely to tokens that are part of n-grams within the vocabulary.

Transform Mutation Operator. Finally, we have the "transform mutation" operator, which changes a token from a specific dimension to another token in that same dimension:

Parent	1	2	3
economy	1	0	0
economic	1	0	0
congress	0	1	0
white_house	0	0	1
risk	0	0	1
⋮	⋮	⋮	⋮

Child	1	2	3
economy	1	0	0
economic	1	0	0
congress	0	1	0
white_house	0	0	1
uncertainty	0	0	1
⋮	⋮	⋮	⋮

The transform mutation is the only operator that activates new tokens (not n-grams) that have not already been activated within the population. This operator thus introduces new tokens within the population by replacing an active token from a candidate selection matrix with a new one. The transform mutation ensures that the number of active tokens within each dimension remains the same. This operator is crucial in controlling the number of active tokens while allowing the introduction of new active tokens. A standard mutation operator that flips one bit

would not work in this case. Consider a vocabulary of size 4,800 with three dimensions, leading to $2,448 \times 3 = 7,344$ potential number of active points. If a parent has ten active tokens out of 7,344, the standard mutation operator has a $\frac{7,344-10}{7,344}) = 99.86\%$ chance of generating a child with 11 active tokens and a 0.14% chance of generating a child with nine active tokens. Over a large number of iterations, this leads to a search space with a large number of active tokens, ultimately leading to overfitting.

By design, many of these operators only apply to tokens that are already active within the corpus to ensure an efficient search. Due to the extreme sparsity of our optimization problem, without these operators, the likelihood of performing non-altering operations is extremely high. We also note that these operations ensure that the number of active tokens in the child will always be the same as the number of active tokens in the parent. This is crucial for our search strategy, which is presented below.

3.2. Calibration

The calibration strategy is centered around (i) efficient search and (ii) avoiding overfitting. We start the optimization process by defining the initial population of $q = 1, \ldots, Q$ candidate selection matrices $\mathbf{\Omega}_q$ of size $V \times K$. Each member of the initial population starts with K active tokens (non-zero value in the selection matrix), where K is the number of dimensions of the selection matrix. Each dimension is set to have one active token. For instance, for $K = 3$, these could be members of the initial population (displaying only active tokens):

$$\begin{array}{c} \\ \text{democrats} \\ \text{european\_union} \\ \text{policy} \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ \text{religion} \\ \text{market} \\ \text{presidency} \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \end{pmatrix} \end{array}$$

while these would not:

$$\begin{array}{c} \\ \text{democrats} \\ \text{european\_union} \\ \text{policy} \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \end{pmatrix} \end{array} \qquad \begin{array}{c} \\ \text{religion} \\ \text{market} \\ \text{presidency} \\ \text{coffee} \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \end{pmatrix} \end{array}$$

Inspired from Scrucca (2013, 2017), we perform genetic optimization. In Figure 1, we display

a flowchart of the optimization algorithm.

[Insert Figure 1 about here.]

Each step is defined as follows:

1. Begin an epoch with an initial population. For the first epoch, this is randomly initialized, while for subsequent epochs, the initial population depends on the results from the previous epoch. Each epoch contains H iterations along these steps:
   (a) For each candidate $q = 1, \ldots, Q$ in the population, compute $f(\boldsymbol{\Omega}_q, \boldsymbol{C}_{t-h})$.
   (b) For each candidate $q = 1, \ldots, Q$ in the population, estimate parameters α, β, and $\boldsymbol{\gamma}$ by ordinary least squares with the computed $f(\boldsymbol{\Omega}_q, \boldsymbol{C}_{t-h})$ as input.
   (c) For each candidate $q = 1, \ldots, Q$ in the population, compute the objective function (6) using the estimated parameters obtained in step (b).
   (d) Elitism step. Store the 5% lowest objective value selection matrices as well as the best solution. 5% is a common choice for elitism steps in genetic algorithms. For the best solution, perform repeated ten-fold cross-validation (Kim, 2009) and store the average root mean square error.
   (e) Tournament selection. Select at random three members of the population, then select the one with the best objective function. Do this Q times to generate an interim population. Selecting a single candidate out of three is a commonly used compromise between exploration (allowing diversity) and exploitation (leveraging high-quality solutions).
   (f) Token crossover: For pairs of two randomly chosen selection matrices in the interim population, perform a token crossover operation with probability S to update the interim population, where $S = \frac{Q_{\text{unique}}}{Q}$ is the ratio of the unique selection matrix in the interim population over the population size.[4] Since doing crossover on two populations that are exacty the same does not do anything, the probability of doing a crossover operation scale. If all matrices are unique ($S = 1$), we would perform crossover on all population members (100% probability); if all matrices are the same, $S \approx 0$, no crossover is performed.

[4] When computing the number of unique selection matrices, we treat permutations of the filtering matrix dimensions as duplicates. For example, suppose a selection matrix has one active word in each dimension: "economic" in dimension one, "policy" in dimension two, and "uncertainty" in dimension three. If we swap "economic" and "policy" between dimensions one and two (while "uncertainty" remains in dimension three), the resulting matrix differs only in the ordering of the dimensions. Consequently, it is considered a duplicate rather than a distinct configuration. This approach prevents dimension permutations from artificially inflating the count of unique selection matrices.

(g) Switch mutation. For each selection matrix in the interim population, perform a switch mutation operation with 5% probability to update the interim population. 5% is chosen arbitrarily.

(h) N-grams mutation. For each selection matrix in the interim population, perform n-grams mutation operation with 5% probability to update the interim population. 5% is chosen arbitrarily.

(i) Transform mutation. For each selection matrix in the interim population, apply the transform mutation operator with probability $1 - S\%$. This operator replaces one active word with a non-active word, introducing new variation. If all matrices in the interim population are identical (S $\approx$ 0) then transform mutation is applied to each one. Conversely, if all matrices are unique, no transform mutation is performed. This design balances the benefits of crossover — when it is valuable to recombine information — with purely random search—when crossover provides little advantage. In addition, any duplicated member of the interim population is mutated until only unique members remain (before Elitism), ensuring a more efficient search across iterations.

(j) Elitism step. Replace 5% instance of the interim population with the 5% lowest objective value solution stored in step (d). We keep the best members of the populations before crossover and mutation. This strategy usually speeds up the convergence of the algorithm.

(k) Start a new iteration with the new population until we reach the last iteration.

2. From H best solutions, each one stored at step (d), select the one with the lowest repeated ten-fold cross-validation error.

3. Refine the vocabulary for the next epoch. To achieve this, we employ word embeddings to reduce the vocabulary size while retaining information from the active tokens of the best solution in the preceding step. A word embedding space maps tokens into high-dimensional vectors, where similar tokens exhibit shorter distances between their vectors. First, we compute the average token vector for each dimension of the best solution in step 2. Then, for each dimension, we calculate the cosine similarity between the average token embedding of that dimension and all the tokens in the original vocabulary of size V. We retain tokens with a cosine similarity larger than 0.2.[5] The set of unique, similar tokens across all dimensions, constitutes our new vocabulary. Effectively, this step narrows down

[5]Empirical tests showed that thresholds above 0.2 significantly shrunk the search space, reducing the algorithm's capacity to discover relevant tokens. Lower thresholds, on the other hand, over-inflated the vocabulary, increasing computational costs without meaningful performance gains.

the search space to a more relevant vocabulary, considering that the optimally selected tokens are indicative of the relevant tokens concerning the dependent variable.

4. Subsequently, we form a new population of size Q using the new vocabulary, where each member starts with $K + 1$ active tokens starting from the optimal solution from the old population. Additionally, we insert the optimal solution from step 2 (which has K active tokens) within the new population. If we find a better solution with more active tokens, it will disappear from the population. Further, each epoch encourage the algorithm to select the most important tokens first. Additionally, it reduces overfitting issues where solutions with many low-impact active tokens (approximating high-impact tokens) would be selected early in the optimization process.
5. Start a new epoch until we achieve the maximum number of desired active tokens (from K up to 15 in our applications).

The solution with the repeated ten-fold cross validation error from step 2 of the last epoch is considered as the interim optimal solution. To obtain the final solution, we perform a pruning step. This step aims to eliminate tokens that may have adversely affected the model fit but were added due to the random nature of the optimization process (*e.g.*, uninformative tokens added before informative ones). To accomplish this, we test variations of the optimal selection matrix by deactivating one or several active tokens, encompassing all variations involving only deactivations. For instance, if the solution with the lowest objective value from the validation window has 12 active tokens, we test $2^{12} = 4{,}096$ potential new solutions (ranging from all tokens being active to all tokens being deactivated).

In our experiments, our estimation strategy often converges to the same or very similar solutions across different (stochastic) runs. This outcome reflects our structured iterative strategy—adding one active token per epoch—which naturally narrows the search space early on by focusing on the most significant tokens in the initial rounds. Consequently, the solutions identified in later epochs inherit these early selections, leading to consistency across multiple runs.

To avoid converence to a local optimum, we investigated several hyperparameter setups and ended up with the following robust choice:

- We set the population to $Q = 600$, enabling the genetic algorithm to explore a large set of candidate selection matrices;
- We find that $H = 200$ iterations offers a practical balance between running time and avoiding suboptimal local minima;
- By modifying—in an adaptive way—the probability of the mutation and crossover operator, we reduce the risk of being stuck in a local optimum.

These choices arise from a combination of (i) standard GA practices, (ii) extensive testing, and (iii) computational feasibility.

Our algorithm is designed to run sequentially, adding one active token per epoch. This controlled expansion helps avoid overfitting by gradually increasing the degrees of freedom. Although a parallel structure might speed up certain aspects of computation, it would alter the core sequential logic that underpins our algorithm's ability to balance exploration and complexity. We do use parallel computing to evaluate different selection matrices simultaneously when computing the objective function—the attention indices. For instance, in our applications (EPU validation and inflation forecasting), a population of 600 was split into batches of 100 across six parallel processes. We found this approach to be the best compromise between runtime and memory constraints, given our large corpus of textual data.

Finally, it is interesting to note that the principles in Romer and Romer (2023, Table 1)'s "Requirements for Rigorous Narrative Analysis", which are primarily designed for human-driven text-mining approaches hold within our data-driven approach.[6]

4. Data and Algorithm Validation

In this section, we describe the corpus of news articles used in the forecasting application of Section 5. We also present a validation exercise of our algorithm in a controlled environment setting—we check that our optimization strategy converges to an optimal and known solution in a reasonable time.

4.1. Textual Data, Corpus Processing, and Vocabulary

Our corpus consists of all news articles published in the printed version of the Wall Street Journal from January 2000 to August 2024. In total, the corpus is composed of 837,576 news articles. To derive candidate tokens for the vocabulary describing the corpus, we proceed as in Ardia et al. (2024):

1. To normalize the news articles, we lowercase and lemmatize each word into its root form.

[6]Specifically: Point 1: "A reliable narrative source." Our approach requires assembling a relevant and credible textual corpus, consistent with Romer and Romer's emphasis on high-quality primary sources; Point 2: "A clear idea of what one is looking for." While our method is data-driven, users can still refine the corpus to target specific domains (e.g., inflation-related or policy-related articles). Additionally, how we transform the textual data (e.g., attention indices) must be well-defined in light of the economic variable of interest; Point 3: "Approach the source dispassionately and consistently." Our systematic and unbiased algorithmic procedure applies the same selection and optimization rules across the entire corpus, fulfilling the principle of consistency in narrative analysis; Point 4: "Document the narrative evidence carefully." Although our procedure does not generate manually annotated "narratives," we can—and do in our forecasting exercise in Section 5—use topic modeling to interpret the themes embedded in the algorithm's selected articles. This step mirrors Romer and Romer's goal of clearly documenting the story derived from the data, albeit through an algorithmic lens rather than a manual narrative approach. We thank a reviewer for pointing this out to us.

2. We then detect and combine collocation in our corpus. For instance, the compound words "interest rate" or "exchange rate" provide more information about the content of an article than if these words were not considered a combination. We rely on two methods to detect collocations in our documents: (i) The RAKE (Rapid Automatic Keyword Extraction) algorithm (Rose et al., 2010) and (ii) the process described in Hansen et al. (2018, Section IV.B).[7] For each method, we select the collocations that occur at least 100 times for bigrams and 50 times for trigrams. Finally, we concatenate the individual tokens of the collocation in each text of our corpus (*e.g.*, "interest rate" becomes "interest_rate").
3. We then build a matrix representation of the text where each column is a token, and each row is a news article such that each element represents the number of times a token is observed for a corresponding news article. We eliminate tokens that appear in less (more) than 1% (20%) of the news articles.
4. Finally, we manually verify each token and remove non-informative tokens such as dates or tokens with numbers. We also remove any token related to specific events, persons, locations, companies, or products. Overall, this results in a vocabulary of size $V = 2{,}448$.

We use the pre-trained continuous bag-of-word embedding model of Rahimikia et al. (2021) available at `https://fintext.ai` to retrieve the token vector for each of the 2,448 tokens. The model is compiled from 15 years of business news archives and, as a domain-specific model, is more appropriate to capture the relationship between tokens than a pre-trained general-domain word embedding model such as Google Word2Vec. For collocations, we take the average of the token vector across the individual collocation tokens. These token vector representations of the 2,448 tokens will be used to narrow down the vocabulary after each epoch.

4.2. Algorithm Validation

As a first empirical experiment of our estimation strategy, we construct a daily EPU index with our corpus of news articles and the EPU keywords in Baker et al. (2016)—this index becomes the dependent variable in our experiment—and then apply our algorithm to the corpus to find the index that maximizes the contemporaneous correlation with the dependent variable. If properly working, the set of optimized tokens obtained with our estimation strategy should

[7] Specifically, RAKE assigns a score to each candidate keyword based on the frequency of occurrence of the keyword in the text, as well as the frequency of occurrence of each of its constituent words in the text. The algorithm then identifies sets of adjacent keywords with high scores, indicating that they are likely collocations. The approach in Hansen et al. (2018, Section IV.B) looks at part of speech patterns within the text. Following Justeson and Katz (1995) these patterns are: adjective-noun, noun-noun, adjective-adjective-noun, adjective-noun-noun, noun-adjective-noun, noun-noun-noun, and noun-preposition-noun. We find the part of speech of each word using the UDPipe methodology implemented in the R package **udpipe** (Wijffels, 2021).

be the same as the EPU keywords. Specifically, we perform the following optimization:

$$(\hat{\alpha}, \hat{\beta}, \hat{\mathbf{\Omega}}) \equiv \underset{\alpha,\beta,\mathbf{\Omega}}{\arg\min} \frac{1}{T} \sum_{t=1}^{T} \left(f_{\text{att}}(\mathbf{\Omega}_{\text{EPU}}, \boldsymbol{C}_t) - \left(\alpha + \beta f_{\text{att}}(\mathbf{\Omega}, \boldsymbol{C}_t)\right)\right)^2 , \tag{7}$$

and see if $\hat{\mathbf{\Omega}} = \mathbf{\Omega}_{\text{EPU}}$.

With V = 2,448 and $K = 3$, the number of selection matrices is $2^{2448\times3}$. Convergence toward the true EPU selection matrix would thus indicate massive improvement compared to a naive random search. Since it is technically possible to recover the true EPU selection matrix given the absence of noise, we do not use any penalty parameter in this test. Therefore, we test the algorithm's ability to converge and avoid local optima.

In Figure 2, we display the repeated ten-fold cross-validation loss after each epoch as well as the number of iterations before convergence—no more reduction in the objective function—and the selected words for each dimension at the end of each epoch (step 2 of the optimization process). We first note that our process, using this configuration, converges to the EPU selection matrix ($\hat{\mathbf{\Omega}}_{K=3} = \mathbf{\Omega}_{\text{EPU}}$) at the end of the optimization process (after the pruning step). Looking at the evolution of the optimal solution across epochs, the top panel shows a steady decrease in the validation loss, indicating that overfitting is unlikely.

The middle panel displays the number of iterations per epoch before convergence. When fewer tokens are allowed to be active, the solution space is more restricted, and convergence tends to occur quickly. As K grows and more tokens can be included, convergence often takes longer and yields minor improvements in validation loss—implying it becomes increasingly difficult to isolate tokens with a substantial impact on the target variable. Nevertheless, once the algorithm can include more tokens than the exact EPU solution requires—the 10 active tokens—it converges more rapidly to that solution. This behavior suggests that a small subset of key tokens (*e.g.*, "economic" and "uncertainty") drives most of the EPU dynamics, and the algorithm identifies them early. Additional policy-related tokens (*e.g.*, "federal_reserve," "congress") appear as K increases as well as "economy" and "uncertain." Eventually, lower-impact tokens (*e.g.*, "white_house," "regulation," or "deficit"), whose impact were previously approximated by other tokens (*e.g.*, "health_care" and "policy"), are included in the final stages. Finally, the pruning step removes redundant tokens such as "budget_deficit," which is an n-gram of "deficit." Interestingly, n-grams are used by the algorithm to remove undesirable tokens by replacing them with tokens that have no effect on the index. Overall, this pattern highlights the complexity of the selection function—the combination of tokens across dimensions is as important as the tokens themselves.

[Insert Figure 2 about here.]

To further validate our methodology, we re-run the algorithm from zero, limiting the active tokens to ten—the number of active keywords of the EPU selection matrix—for a single epoch of 2,000 iterations—thus ten times more then our previous 200 iterations. This scenario assumes prior knowledge of the number of active tokens. Under these constraints, the algorithm failed to converge to the EPU keywords, instead becoming trapped in a local minimum. This exercise highlights the critical aspect of our iterative approach. Moreover, in real-world applications, the optimal number of active tokens is typically unknown. This underscores the benefits of our iterative, multi-epoch structure, which strikes a balance between efficiency and flexibility, enabling a dynamic exploration of the search space until convergence.

Finally, we note that the estimation time on a standalone computer (Intel i9 3.7GHz using 6 logical processors and 48 GB RAM) using parallel processing—to compute the attention-based indices conditional on the candidate selection matrices—takes approximately nine hours and 45 minutes. Hence, this validation exercise highlights that our optimization strategy can recover pre-defined selection matrices within a reasonable computational time.

5. Forecasting Inflation

Our empirical application is inspired from Eugster and Uhl (2024), who suggest that U.S. inflation can be forecasted using news sentiment. They select news articles from predefined topics and sentiments available in Refinitiv RNA's Natural Language Processing. In particular, they focus on news items tagged with the following topic labels: N2:INFL, which includes inflation figures and forecasts; N2:PLCY, which covers events related to monetary and fiscal policymakers; and N2:FED, which includes news items about activities by or involving the U.S. Federal Reserve.

A key aspect proposed by Eugster and Uhl (2024) is the dynamic relationship between inflation and news sentiment. Specifically, to measure sentiment tied to inflation, raw sentiment scores are adjusted according to the prevailing inflationary regime. While positive sentiment in news articles typically signals higher GDP or lower unemployment, its relationship to inflation depends on whether inflation is low or high. The negative tone during low-inflation periods often suggests lower inflation, whereas during high-inflation periods, the same negative tone likely points to higher inflation.

In light of their work, our objective is to create, through our framework, an index of negative-news attention that is positively correlated with inflation rates when the five-year breakeven inflation rate is above 2% (inflation target) and negatively correlated otherwise. Thus, when negative-news attention is high during high-inflation periods, this would indicate a further increase in inflation. In contrast, when negative-news attention is high during low-inflation

periods, this would indicate a further decrease in inflation. We retrieve the five-year breakeven inflation rate from https://fred.stlouisfed.org/series/T5YIE.

5.1. Measuring Negative News

To incorporate only negative news in our pipeline, we extract the sentiment of each article in our corpus. To do so, we rely on FinBERT, an open-source large language model (LLM) developed by Huang et al. (2023). The model is available at https://huggingface.co/yiyanghkust/finbert-tone. FinBERT adapts BERT to the finance domain, outperforming simpler, more conventional sentiment analysis systems often used in finance—such as lexicon-based methods (Loughran and McDonald, 2011). Other LLMs, such as Generative Pretrained Transformers, could have been used to provide potentially more accurate sentiment estimates. Their usage's cost is, however, prohibitive with the size of our corpus.

FinBERT is adapted to compute sentiment at the sentence level. To extract the sentiment of each news article, we follow Bae et al. (2024), and compute the net positive sentence ratio of a news article as follows:

$$NetPositiveSentenceRatio = \frac{\#PosSentence - \#NegSentence}{\#TotalSentence}, \tag{8}$$

where $\#PosSentence$ ($\#NegSentence$) is the number of positive (negative) sentences, and $\#TotalSentence$ is the total number of sentences, in the news article.

Figure 3 displays the distribution of news article sentiment. The average sentiment is slightly negative at -0.0501, and the median at -0.0377. To select only negative (and not neutral) news, we filter articles whose net positive sentence ratio is below the first tercile of the distribution—new articles whose sentiment is below -0.1020. This reduces the corpus from 837,576 to 279,465 news articles. We rely on the same vocabulary of 2,448 tokens used in the EPU experiment of Section 4.2 to construct the selection matrices.

[Insert Figure 3 about here.]

5.2. Modeling Framework

To account for the regime shift in how negative-news attention and inflation interact within our optimization process, we introduce a transformation layer that switches the sign of the attention index, allowing it to better capture the presumed relationship between the target variable (inflation at various horizons) and the attention index. Let B_t be the five-year breakeven

inflation rate measured at the end of month t. We define the transformation function as:

$$f_{\text{trans}}(\boldsymbol{\Omega}, \boldsymbol{C}_t, B_t) = \begin{cases} f_{\text{att}}(\boldsymbol{\Omega}, \boldsymbol{C}_t), & \text{if } B_t \geq 2\%. \\ -f_{\text{att}}(\boldsymbol{\Omega}, \boldsymbol{C}_t), & \text{otherwise}\,. \end{cases} \tag{9}$$

Following Eugster and Uhl (2024), our measure of inflation is the annualized inflation rate from the Consumer Price Index for All Urban Consumers (CPI) in the United States. For a monthly horizon $h \geq 1$, the inflation rate from time t to time $t+h$ is given by:

$$\pi_{t,t+h} = \frac{1200}{h} \times \ln \frac{P_{t+h}}{P_t}\,, \tag{10}$$

where P_t denotes the CPI at time t. We also consider a special case where $h = 0$ as a nowcasting task, defined as the current one-month inflation rate:

$$\pi_{t,t} = 1200 \times \ln \frac{P_t}{P_{t-1}}\,. \tag{11}$$

With this setup, our objective function becomes:

$$\begin{aligned} \min_{\alpha,\beta,\boldsymbol{\gamma},\boldsymbol{\Omega}} \frac{1}{T} \sum_{t=1}^{T} \left(\pi_{t,t+h} - \left(\alpha + \beta f_{\text{trans}}(\boldsymbol{\Omega}, \boldsymbol{C}_t, B_t) + \boldsymbol{\gamma}' \boldsymbol{x}_t\right)\right)^2 \\ + \frac{\lambda_1}{T} \sum_{t=1}^{T} I[f_{\text{att}}(\boldsymbol{\Omega}, \boldsymbol{C}_t) < \tau] \\ + \lambda_2 I[\beta > 0] + \lambda_3 I[\beta \leq 0]\,. \end{aligned} \tag{12}$$

In (12), we set $\lambda_2 = 1$ and $\lambda_3 = 0$, that is, we constrain the problem such that there is a positive relationship between the transformed negative-news attention index and the inflation rate. For λ_1, we consider $\lambda_1 = 0$ and $\lambda_1 = 1$ to analyze whether this penalty can help to generalize in-sample performance to out-of-sample performance in a high noise-to-signal ratio environment. For the threshold τ, we set it to 0.1%, which ranges from one to two documents a month in terms of document number. As such, solutions that often select more than one or two documents will not be penalized. On the contrary, solutions that frequently select fewer than one or two documents per month will be heavily penalized. For the control variable $\boldsymbol{x}_t$, we use a non-overlapping autoregressive term, that is, $\pi_{t-h,t}$. We also consider a specification with no control. Finally, we consider $h \in 0, 1, 3, 6$ and use $K = 3$ in all our forecasting exercises. Overall, this leads to 16 setups, thus 16 selection matrices—four for each time horizon, where each time horizon has a set of two selection matrices optimized with control and two without control, and from those sets of two, one has a sparsity penalty while the other does not.

We optimize the 16 selection matrices using data from January 2000 to December 2019 (training window), leaving January 2020 to August 2024 as the out-of-sample (OOS) window. Each of the optimized selection matrix $f_{\text{trans}}(\hat{\boldsymbol{\Omega}}, \boldsymbol{C}_t, B_t)$ yields a negative-news attention index value at time t, that is integrated as an exogenous variable into a (non-overlapping) AR(1) forecasting specification:

$$\pi_{t,t+h} = \rho_0 + \rho_1 \pi_{t-h,t} + \rho_2 f_{\text{trans}}(\hat{\boldsymbol{\Omega}}, \boldsymbol{C}_t, B_t) + \varepsilon_t \,. \tag{13}$$

We consider an AR(1) specification as adding lag(s) of the target variables or other exogenous variables has been shown to be useful to increase the predictive accuracy on top of text-based indicators (Banbura et al., 2010). The AR(1) parameters ρ_i $(i = 0, 1, 2)$ in (13) are estimated following an expanding-window scheme with data up to time $t-1$, while $\hat{\Omega}$ is the selection matrix estimated on the training window only. For instance, to estimate the forecast error in January 2021, we use the selection matrix optimized from data covering January 2000 to December 2019 to compute the negative-news attention index; we then estimate the AR(1) parameters using data from January 2000 to December 2020. We repeat this procedure one month at a time until we cover the full OOS window.

5.3. Results

Table 1 reports the root mean square error (RMSE) of our forecasting exercise, expressed relative to the constant-only or AR(1) benchmarks—that is, forcing $\rho_1 = \rho_2 = 0$ and $\rho_2 = 0$ in (13), respectively. The first column indicates which model is used; the second and third columns detail the configuration used to estimate the negative-news attention index in the training sample; and the last four columns report the OOS RMSE for each horizon. We also conduct pairwise Diebold–Mariano tests (Diebold and Mariano, 1995) to assess whether the improvement in RMSE is statistically significant compared to the benchmark model. Unilateral testing is implemented with the heteroscedasticity and autocorrelation robust (HAC) standard error estimators of Andrews (1991) and Andrews and Monahan (1992) with p-values computed following Clark and McCracken (2001).

First, looking at the benchmark models only, the AR(1) model performs better than the constant-only model for $h = 0$ and $h = 1$, while the constant-only performs better than the AR(1) when $h = 6$. When including the negative-news attention index, we find that the penalty controlling for the low number of selected articles yields equally or better performance over all horizons. Thus, we focus on the model with $\lambda_1 = 1$ for the analysis on how the negative-news attention indices perform relative to the benchmarks. In this case, for all horizons, the negative-news attention index brings additional performance over the constant-only and the

AR(1) models. Controlling for inflation in the construction of the negative-news attention index (CTRL Yes) does not enhance or reduce performance.

[Insert Table 1 about here.]

Next, we analyze the selection matrices. First, in Figure 4, we show the active tokens of the negative-news attention indices estimated with $\lambda_1 = 1$ and no AR(1) control (CTRL No) for each forecasting horizon. From these selection matrices, it is not easy to discern clear topical structures comparable to those arising from a manually designed EPU-type index—one drawback of such an optimized approach is that interpretability can be more challenging and is not necessarily guaranteed.

[Insert Figure 4 about here.]

To better understand which news articles are selected by each selection matrix at each forecast horizon, we employ the Correlated Topic Model (CTM) of Blei and Lafferty (2006). The CTM is an unsupervised learning approach that infers latent, correlated topics across a collection of texts; in particular, each text is modeled as a mixture of M topics, and each topic is modeled as a mixture of V words/tokens. This approach yields (i) a vector of topic prevalence for each article and (ii) a vector of word probabilities for each topic. To calibrate the CTM on our corpus, we proceed in two steps. First, we estimate the CTM at each forecast horizon for a range of $M \in \{3, 4, 5, ..., 20\}$ and select the optimal number of topics based on semantic coherence and exclusivity metrics. Second, we cluster the topics across forecast horizons (*i.e.*, across topic models) by inspecting the ten most probable words in each topic and examining the content of the new articles with the highest topic prevalence. In doing so, we can assess whether there is any topic coherence or similarity among the news articles selected for different forecast horizons.

In Table 2, we report a sample of words among the 20 most probable words of each topic, representing the topics, along with each topic's prevalence clustered into themes across forecast horizons. Several of the identified topics align closely with well-known inflation drivers. In particular, "Bonds, Interest Rates, and Monetary Policy" repeatedly features central banking terms (*e.g.*, "federal_reserve," "bond," "inflation," "treasury"), suggesting that the negative-news attention model is effectively capturing macro-financial signals commonly associated with the inflation rate. Likewise, "Oil, Energy, and Geopolitics" surfaces across forecasts, reflecting the significant role of energy shocks in shaping consumer prices. Although other themes may appear less directly linked to inflation—such as "Lifestyle, Family, Education, and Culture" or "Consumer Goods, Real Estate, and Consumer Finances"—they can still affect inflation indirectly by capturing shifts in consumer spending, credit availability, and labor market conditions.

Moreover, a set of topics revolves around "Investments, Hedge Funds, and Corporate Earnings," hinting that the model also incorporates corporate performance and broader financial-market sentiment. These factors can feed into firms' pricing power and overall cost structures—key elements in shaping inflation expectations. In short, the negative-news attention model's tendency to focus on varying sectors underscores its breadth in picking up both direct and indirect signals of future price changes.

[Insert Table 2 about here.]

6. Conclusion

In this research, we address a critical aspect often overlooked in economic and financial analysis using textual data—the text selection process. We introduce an algorithm that determines which set of texts, among a large corpus, leads to a text-based index that is optimal for a specific objective—typically, an index that maximizes the contemporaneous relation or the predictive performance with respect to a target variable, such as inflation. Our approach, based on a genetic algorithm and tailored crossover and mutation operators, offers a data-driven and systematic text selection procedure.

We illustrate the relevance of our approach using a large collection of news articles from the Wall Street Journal. First, we show that the keywords of the EPU index (Baker et al., 2019) can be recovered in an automatic data-driven manner with our algorithm, thus validating our approach. Second, we perform an inflation forecast exercise by building optimized negative-news attention indices. We find that our optimization framework significantly enhances out-of-sample forecasting performance. Also, we show that the textual indices are linked to meaningful economic topics/themes, such as monetary policy, energy prices, and financial markets.

Our approach can be extended in several directions. For instance, we could consider a nonlinear framework or incorporate more sophisticated selection rules, such as proximity-based token interactions or context-aware relationships between terms. Also, as suggested by a referee, the approach could be extended to a multivariate setting with sign restrictions (Canova and Nicoló, 2002; Uhlig, 2005). By jointly minimizing a loss function for industrial production (quantity) and inflation (price), the algorithm could identify tokens linked to supply shocks (*e.g.*, rising production, falling prices). Adjusting penalties in (6) would refine this structural framework, offering a lexicon for macroeconomic shock identification. We leave this for further research.

Supplementary Materials

Please contact the corresponding author to have access to the computer code and the data sets.

Acknowledgements

We thank the editor (Esther Ruiz Ortega), the associate editor, and two referees for their insightful comments and suggestions. We also thank Kris Boudt, Gilles Caporossi, Sébastien Laurent, and Carlos Ordas for their helpful comments. Finally, we thank IVADO and the Natural Sciences and Engineering Research Council of Canada (grant RGPIN-2022-03767) for their financial support.

References

Algaba, A., Ardia, D., Bluteau, K., Borms, S., and Boudt, K. (2020). Econometrics meets sentiment: An overview of methodology and applications. *Journal of Economic Surveys*, 34(3):512–547.

Andrews, D. W. (1991). Heteroskedasticity and autocorrelation consistent covariance matrix estimation. *Econometrica*, 59(3):817–858.

Andrews, D. W. and Monahan, J. C. (1992). An improved heteroskedasticity and autocorrelation consistent covariance matrix estimator. *Econometrica*, 60(4):953–966.

Ardia, D., Bluteau, K., and Boudt, K. (2019). Questioning the news about economic growth: Sparse forecasting using thousands of news-based sentiment values. *International Journal of Forecasting*, 35(4):1370–1386.

Ardia, D., Bluteau, K., and Boudt, K. (2022). Media abnormal tone, earnings announcements, and the stock market. *Journal of Financial Markets*, 61:100683.

Ardia, D., Bluteau, K., Boudt, K., and Inghelbrecht, K. (2023). Climate change concerns and the performance of green vs. brown stocks. *Management Science*, 69(12):7607–7632.

Ardia, D., Bluteau, K., and Kassem, A. (2021). A century of economic policy uncertainty through the French–Canadian lens. *Economics Letters*, 205:109938.

Ardia, D., Bluteau, K., and Meghani, M.-A. (2024). Thirty years of academic finance. *Journal of Economic Surveys*, 38(3):1008–1042.

Bae, J., Berger, A. N., Choi, H.-S., and Kim, H. H. (2024). Bank sentiment and loan loss provisioning. *Working paper*.

Baker, S. R., Bloom, N., and Davis, S. J. (2016). Measuring economic policy uncertainty. *Quarterly Journal of Economics*, 131(4):1593–1636.

Baker, S. R., Bloom, N., Davis, S. J., and Kost, K. J. (2019). Policy news and stock market volatility. Technical report, National Bureau of Economic Research.

Banbura, M., Giannone, D., and Reichlin, L. (2010). Nowcasting. Working paper.

Barbaglia, L., Consoli, S., and Manzan, S. (2023). Forecasting with economic news. *Journal of Business & Economic Statistics*, 41(3):708–719.

Blei, D. and Lafferty, J. (2006). Correlated topic models. *Advances in Neural Information Processing Systems*, 18:147.

Caldara, D. and Iacoviello, M. (2022). Measuring geopolitical risk. *American Economic Review*, 112(4):1194–1225.

Canova, F. and Nicoló, G. D. (2002). Monetary disturbances matter for business fluctuations in the G-7. *Journal of Monetary Economics*, 49(6):1131–1159.

Clark, T. E. and McCracken, M. W. (2001). Tests of equal forecast accuracy and encompassing for nested models. *Journal of Econometrics*, 105(1):85–110.

Consoli, S., Barbaglia, L., and Manzan, S. (2022). Fine-grained, aspect-based sentiment analysis on economic and financial lexicon. *Knowledge-Based Systems*, 247:108781.

Diebold, F. X. and Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3):134–144.

Eugster, P. and Uhl, M. W. (2024). Forecasting inflation using sentiment. *Economics Letters*, 236:111575.

Gavriilidis, K. (2021). Measuring climate policy uncertainty. Working paper.

Gentzkow, M., Kelly, B., and Taddy, M. (2019). Text as data. *Journal of Economic Literature*, 57(3):535–574.

Handley, K. and Limão, N. (2022). Trade policy uncertainty. *Annual Review of Economics*, 14:363–395.

Hansen, S., McMahon, M., and Prat, A. (2018). Transparency and deliberation within the FOMC: A computational linguistics approach. *Quarterly Journal of Economics*, 133(2):801–870.

Huang, A. H., Wang, H., and Yang, Y. (2023). Finbert: A large language model for extracting information from financial text. *Contemporary Accounting Research*, 40(2):806–841.

Husted, L., Rogers, J., and Sun, B. (2020). Monetary policy uncertainty. *Journal of Monetary Economics*, 115:20–36.

Jegadeesh, N. and Wu, D. (2013). Word power: A new approach for content analysis. *Journal of Financial Economics*, 110(3):712–729.

Justeson, J. S. and Katz, S. M. (1995). Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27.

Kelly, B., Manela, A., and Moreira, A. (2021). Text selection. *Journal of Business & Economic Statistics*, 39(4):859–879.

Kim, J.-H. (2009). Estimating classification error rate: Repeated cross-validation, repeated hold-out and bootstrap. *Computational Statistics & Data Analysis*, 53(11):3735–3745.

Loughran, T. and McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. *Journal of Finance*, 66(1):35–65.

Loughran, T. and McDonald, B. (2014). Measuring readability in financial disclosures. *Journal of Finance*, 69(4):1643–1671.

Manela, A. and Moreira, A. (2017). News implied volatility and disaster concerns. *Journal of Financial Economics*, 123(1):137–162.

Rahimikia, E., Zohren, S., and Poon, S.-H. (2021). Realised volatility forecasting: Machine learning via financial word embedding. Working paper.

Romer, C. D. and Romer, D. H. (2023). Presidential address: Does monetary policy matter? The narrative approach after 35 years. *American Economic Review*, 113(6):1395–1423.

Rose, S., Engel, D., Cramer, N., and Cowley, W. (2010). Automatic keyword extraction from individual documents. In *Text mining: Applications and Theory*, pages 1–20. John Wiley & Sons, Ltd.

Scrucca, L. (2013). GA: A package for genetic algorithms in R. *Journal of Statistical Software*, 53(4):1–37.

Scrucca, L. (2017). On some extensions to GA package: Hybrid optimisation, parallelisation and islands evolution. *R Journal*, 9(1):187–206.

Uhlig, H. (2005). What are the effects of monetary policy on output? Results from an agnostic identification procedure. *Journal of Monetary Economics*, 52(2):381–419.

Wijffels, J. (2021). *udpipe: Tokenization, Parts of Speech Tagging, Lemmatization and Dependency Parsing with the 'UDPipe' 'NLP' Toolkit.*

Table 1: Out-of-Sample RMSE

This table reports the root mean square error (RMSE) for each model under various configurations of the negative-news attention index optimization process ($\lambda_1 = \{0, 1\}$ and with/without AR(1) control), evaluated across different forecasting horizons. The constructed negative-news attention index is integrated into a constant-only model and compared to it (top panel) or an AR(1) model and compared to it (bottom panel). For a given monthly horizon h (column), the dark gray cell reports the top-performing model, while the light gray cell reports models that are statistically indistinguishable from the best performer at the 10% level, according to the Diebold–Mariano test (Diebold and Mariano, 1995). Testing is implemented with the heteroscedasticity and autocorrelation robust (HAC) standard error estimators of Andrews (1991) and Andrews and Monahan (1992), and with p-values computed following Clark and McCracken (2001).

	Attention Index		Horizon $h =$			
Model	λ_1	CTRL	0	1	3	6
Constant-Only			5.13	5.17	4.30	3.72
Constant + Attention Index	0	No	5.75	5.91	4.29	3.88
Constant + Attention Index	0	Yes	5.51	5.38	4.47	3.38
Constant + Attention Index	1	No	4.64	4.61	3.70	3.41
Constant + Attention Index	1	Yes	4.36	4.65	3.80	3.43
AR(1)			4.19	4.19	4.29	4.28
AR(1) + Attention Index	0	No	5.03	5.03	4.40	4.39
AR(1) + Attention Index	0	Yes	4.83	4.51	4.66	4.23
AR(1) + Attention Index	1	No	4.05	4.06	3.90	4.11
AR(1) + Attention Index	1	Yes	3.92	4.06	4.07	4.24

Table 2: Topics and Themes
This table reports results from the four topic models. Each topic model is applied to the news articles selected by the selection matrix optimized with $\lambda_1 = 1$, not AR control, and for a given horizon h ($h = 0, 1, 3, 6$). The first column reports the forecasting horizon (*i.e.*, the model), the second column reports five tokens representing the topic, and the last column reports the topic's prevalence within the model. Topics are organized by themes (clusters of topics) across each model. We also report the average prevalence of each cluster across the topic models. Note that several horizons can be assigned to a theme as several topics can be related to a theme.

Horizon	Tokens	Prevalance
Theme 1	Investments, Hedge Funds, and Corporate Earnings	21.1%
$h = 0$	hedge fund, investor, portfolio, manager, yield, trading	15.3%
$h = 1$	hedge_fund, shareholder, pension, equity, stake, trader	18.0%
$h = 3$	investment_bank, merger, brokerage, ipo, equity, banker	16.3%
$h = 6$	stock_index, rally, strategist, valuation, volume, bull	18.3%
$h = 6$	hedge_fund, asset, strategy, client, capital, management	16.4%
Theme 2	Bonds, Interest Rates, and Monetary Policy	15.8%
$h = 0$	inflation, federal_reserve, euro, economy, policy, rate	15%
$h = 1$	federal_reserve, yield, treasury, bond, liquidity, banks	17.3%
$h = 3$	yield, bond_market, benchmark, rate_increase, investor, currency	14.5%
$h = 6$	treasury, debt, bond, interbank, credit, liquidity	16.3%
Theme 3	Banking, Mortgages, and the Financial Crisis	11.7%
$h = 0$	bank, loan, mortgage, borrower, regulator, liquidity	16.6%
$h = 1$	housing, property, financing, insurer, debt, credit	9.9%
$h = 3$	banks, bailout, financial_crisis, rescue, capital, risk	11.9%
$h = 3$	mortgage, foreclosure, bankruptcy, household, default, portfolio	8.4%
Theme 4	Politics, Legislation, Law Enforcement, and Regulation	14.0%
$h = 0$	democrats, republicans, senate, election, congress, legislation	10.7%
$h = 1$	vote, tax, bill, committee, lobbyist, campaign	10.2%
$h = 1$	fbi, prosecutor, lawsuit, justice, court, fraud	6.5%
$h = 3$	prime_minister, parliament, deficit, sanction, reform, governor	9.0%
$h = 3$	court, lawyer, investigation, prosecutor, sec, probe	7.6%
$h = 6$	sec, insider, compliance, settlement, disclosure, regulator	11.9%
Theme 5	Oil, Energy, and Geopolitics	7.8%
$h = 0$	oil, crude, barrel, gas, refinery, utility	7.1%
$h = 1$	export, energy, production, recession, output, fuel	10.5%
$h = 3$	commodity, infrastructure, electricity, carbon, fuel, factory	6.9%
$h = 6$	oil_price, emission, producer, metal, carbon, export	6.8%
Theme 6	Lifestyle, Family, Education, and Culture	9.7%
$h = 0$	film, actor, father, daughter, love, writer	12.0%
$h = 1$	student, school, teacher, parent, university, science	6.9%
$h = 3$	apartment, rent, family, career, town, graduate	6.2%
$h = 6$	music, tv, media, magazine, sport, entertainment	13.6%
Theme 7	Consumer Goods, Real Estate, and Consumer Finances	7.5%
$h = 0$	property, retailer, construction, consumer, payment, restaurant	9.4%
$h = 1$	broker, transaction, compliance, regulator, account, sanction	3.5%
$h = 3$	credit_card, fee, merchant, issuer, purchase, reward	7.6%
$h = 6$	brand, apparel, manufacturer, luxury, shopper, product	9.1%
Theme 8	Other Topics	12.5%
$h = 0$	ceo, acquisition, merger, ipo, shareholder, venture	13.8%
$h = 1$	app, software, internet, device, video, advertising	8.6%
$h = 1$	ceo, soldier, army, weapon, attack, rebel, war	8.7%
$h = 3$	airline, flight, passenger, airport, ticket, travel	4.1%
$h = 3$	web, e-commerce, content, music, tech, app	7.2%
$h = 6$	cancer, doctor, hospital, vaccine, drug, study	7.6%

Figure 1: Flowchart of the Optimization Algorithm

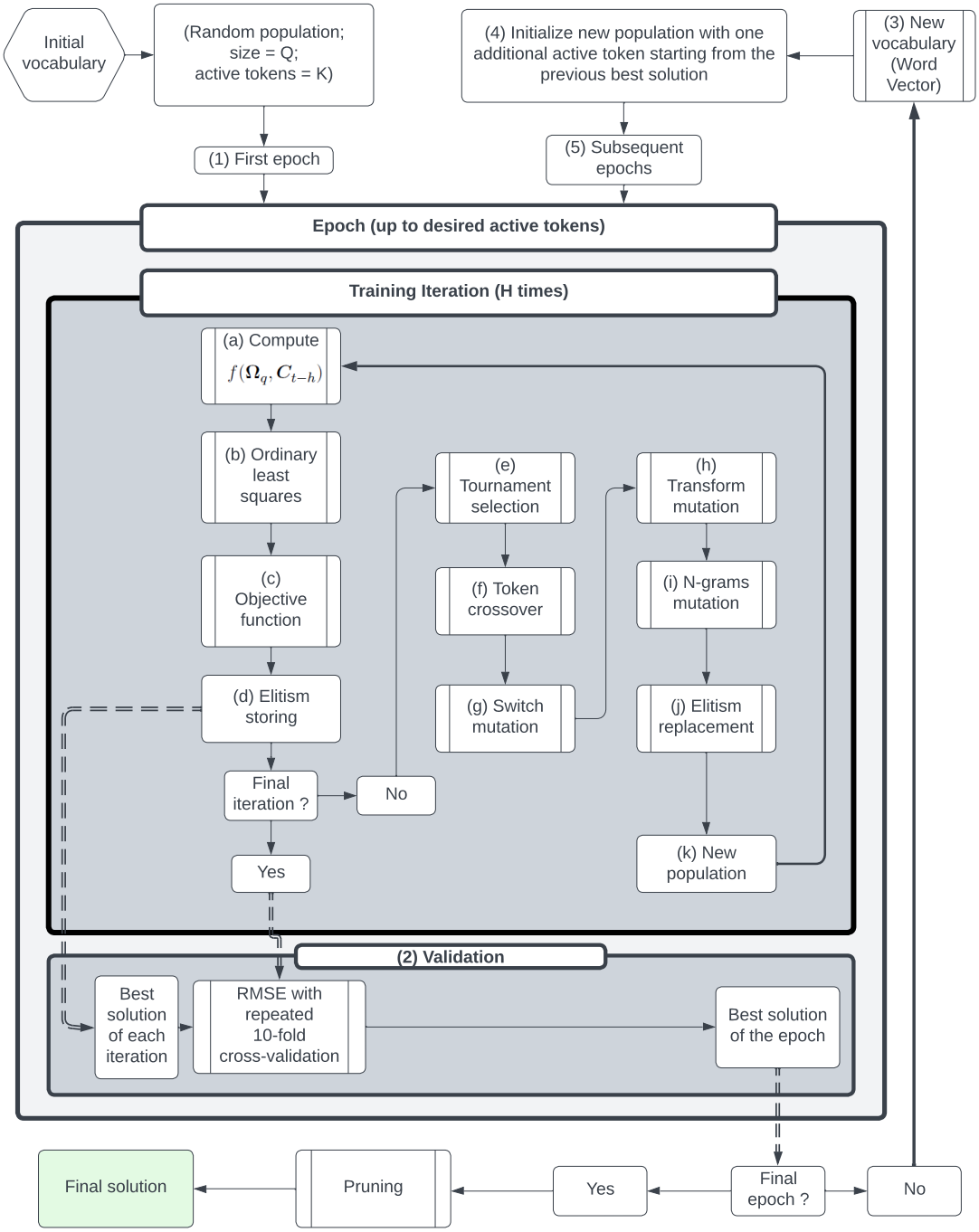

Figure 2: EPU Validation

This figure displays the optimization process for the EPU validation experiment. The first panel shows the validation loss after each epoch. The second panel shows the number of iterations within each epoch until convergence (no increase in training loss). The last three panels report the selected tokens for each dimension of the selection matrix at each epoch, that is, the solution with the lowest validation loss at each epoch. "P" denotes the pruning step.

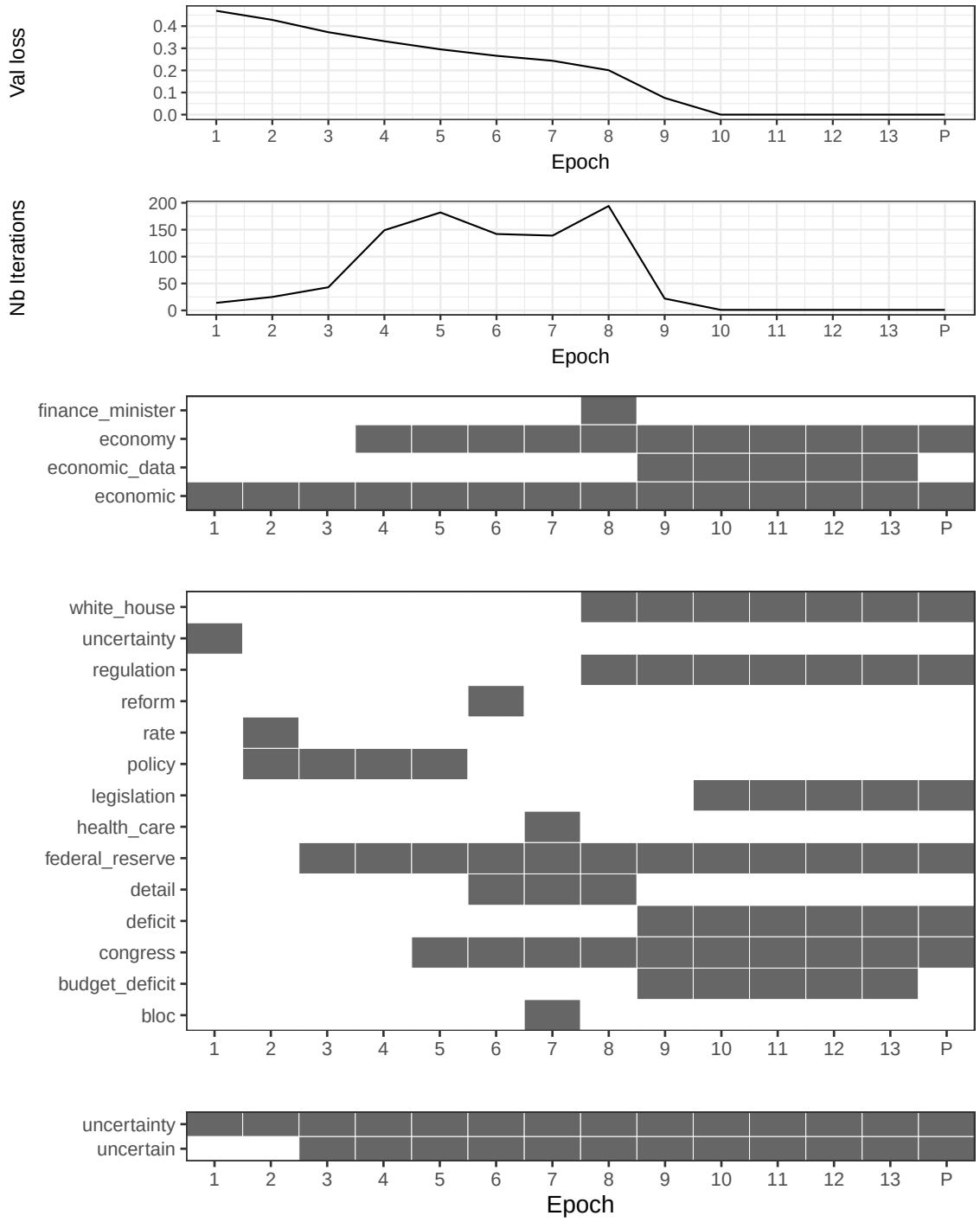

Figure 3: Distribution of News Article Sentiment
The figure shows the distribution of the net positive sentence ratio for all articles in the corpus. The red vertical line marks the first tercile cutoff at -0.1020.

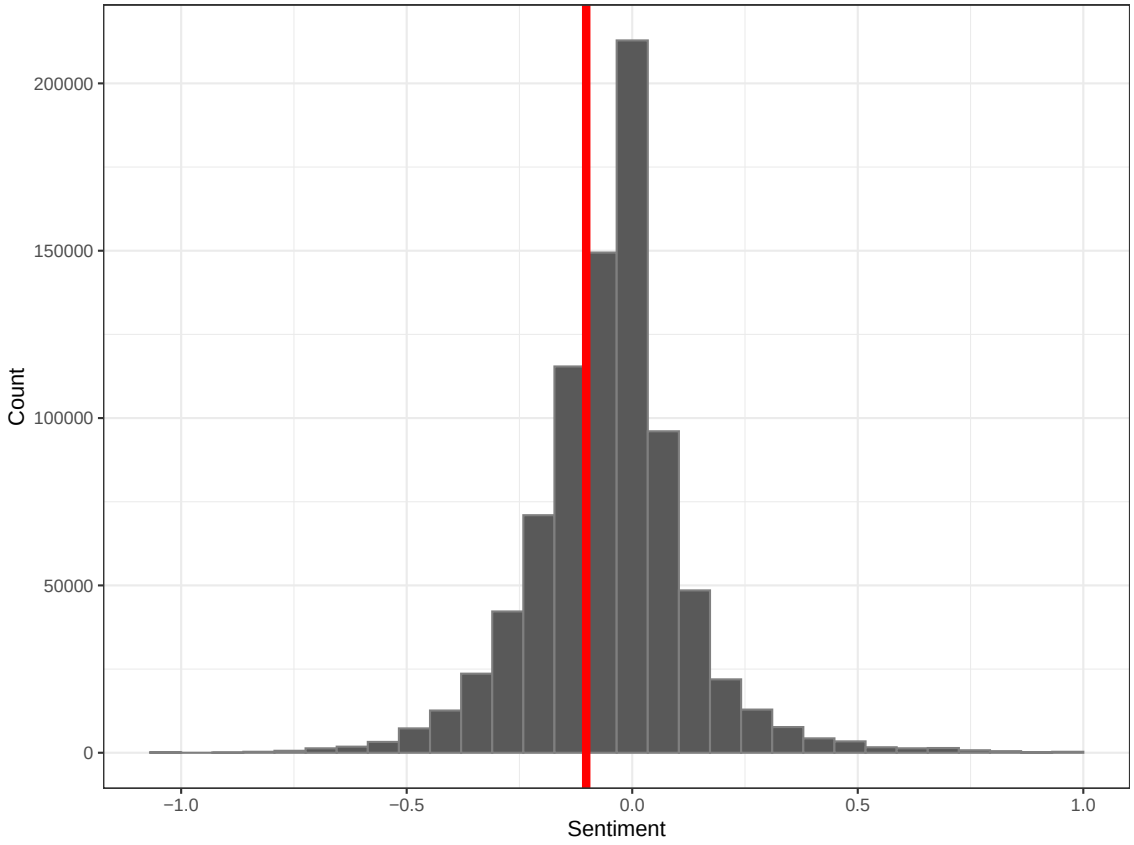

Figure 4: Optimized Selection Matrices for Inflation Forecasts
This figure displays the optimized selection matrices obtained with $\lambda_1 = 1$, not AR control, and for a given monthly forecasting horizon h ($h = 0, 1, 3, 6$). Each matrix (dimensions) highlights the tokens (in rows) that are selected for a given horizon (in columns).