Interpretable Machine Learning for the Social Sciences: Applications in Political Science and Labor Economics

Keyon Vafa

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2023

Abstract

Interpretable Machine Learning for the Social Sciences:
Applications in Political Science and Labor Economics

Keyon Vafa

Recent advances in machine learning offer social scientists a unique opportunity to use data-driven methods to uncover insights into human behavior. However, current machine learning methods are opaque, ineffective on small social science datasets, and tailored for predicting unseen values rather than estimating parameters from data. In this thesis, we develop interpretable machine learning techniques designed to uncover latent patterns and estimate critical quantities in the social sciences. We focus on two aspects of interpretability: explaining individual model predictions and discovering latent patterns from data. We describe a method for explaining the predictions of general, black-box sequence models. This method approximates a combinatorial objective to elucidate the decision-making processes of sequence models. Next, we narrow our focus to domain-specific applications. In political science, we develop the text-based ideal point model, a model that quantifies political positions from text. This model marries a classical idea from political science with a Bayesian matrix factorization technique to infer meaningful structure from text. In labor economics, we adapt a model from natural language processing to analyze career trajectories. We describe a transfer learning method that can overcome the constraints posed by small survey datasets. Finally, we adapt this predictive model to estimate an important quantity in labor economics: the history-adjusted gender wage gap.

Table of Contents

List of Figures

List of Tables

Acknowledgements

Foremost, I would like to thank my advisor, David Blei, whose commitment to my academic growth has made this thesis possible. David has been a constant source of encouragement since the first day of my PhD. His enthusiasm for research is contagious and inspiring. I could not ask for a better advisor.

I am also incredibly grateful to Susan Athey for her mentorship during my PhD. Her expertise and extensive knowledge played a crucial role in shaping my pursuit of inter-disciplinary research. This thesis would not be possible without Susan's generous investment of time and support.

I would also like to thank Suresh Naidu. I have benefited from countless thought-provoking conversations with Suresh throughout my PhD. His enthusiasm has encouraged me to pursue captivating research problems.

Sasha Rush has also been incredibly supportive of my academic development ever since I was an undergraduate who asked him what a neural network was. Sasha's wide-range of expertise, coupled with his hands-on mentorship, have been instrumental to my growth as a researcher.

I am fortunate to have collaborated with many other colleagues and mentors who have helped shape this work. Jaan Altosaar, Elliott Ash, Joe Blitzstein, Peter Brown, Yuntian Deng, David Duvenaud, Amir Feder, Alessandro Grande, Alp Kucukelbir, Scott Linderman, Michael Luca, Xiao-Li Meng, Gemma Moran, David Parkes, Rajesh Ranganath, Matt Salganik, Aaron Schein, Claudia Shi, Dhanya Sridhar, Richard Zemel, and Carolina Zheng have all provided invaluable insights, feedback, and support during my PhD.

Additionally, I am grateful for the financial support I received from the NSF GRFP fellowship and the Cheung-Kong Innovation Doctoral Fellowship during my PhD. I also thank Facebook Artificial Intelligence Research and Google Brain for hosting me for summer internships.

Finally, I would like to thank my friends and family who have continuously provided me with love and support. Thank you to my mom and dad for their love and guidance. Thank you to my brothers, Farzan and Neekon, for being there both when I wanted to discuss specific research ideas and when I wanted to take a break and win a round of *Monopoly Deal*. Thank you to my grandparents for always bringing joy into my life. Thank you to my uncles, Ali and Amir, for providing me with a home away from home. And thank you to Katherine, for your unwavering support throughout my academic journey, from meeting at statistics office hours to helping proofread these acknowledgements.

Dedication

To my parents and grandparents.

Chapter 1: Introduction

Recent advances in machine learning have delivered methods that can mimic human behavior with remarkable fluency. These methods can write *Harry Potter* fan-fiction, compete in programming competitions, and produce photographs of the Big Bang in the style of Ansel Adams. Modern machine learning methods are adept at emulating a wide range of human activity. But can they uncover insights into human behavior?

Economists, political scientists, and other social scientists are increasingly turning to machine learning for methods that analyze human behavior from data. Frequently, their goals are to reveal latent structure or estimate distributional quantities: What is the causal effect of the writing style of a news article on a reader's perception of the issues? What are the common features of career trajectories for which the racial wage gap is most pronounced? However, adapting machine learning techniques to answer these questions involves significant challenges: modern methods are opaque; the most successful models require large datasets while social science datasets are small; machine learning methods are tailored for predicting unseen values rather than estimating underlying parameters.

This thesis presents work aimed at harnessing the rich predictive abilities of recent advances in machine learning into methods that uncover insight into human behavior. A recurring theme throughout this thesis is the importance of *interpretability* for social science research. Interpretability takes many flavors (Lipton, 2018). We focus on two: explaining individual model predictions and discovering latent patterns from data that can help social scientists study human behavior.

Explainability is important in myriad social science settings: a labor economist relies on model explanations to identify career trajectories linked to unemployment. A policy-maker modeling educational outcomes requires justifications to design targeted interventions. This thesis presents a method for explaining the predictions of modern, black-box machine learning models.

In addition to explainability, this thesis develops methods for estimating latent values from data. These include causal effects and latent variables that correspond to real-world phenomena. In general, social science researchers model data primarily to estimate latent values, while machine learning researchers focus on modeling data to make predictions for unseen data points. The tension between estimation and prediction—between $\hat{\beta}$ versus $\hat{y}$ (Mullainathan & Spiess, 2017)—poses challenges for translating human-like predictive performance into novel insights into human behavior. However, these two goals are not incompatible; advances in prediction can enhance estimation techniques and expand the scope of possible analyses (Hofman et al., 2021). This thesis demonstrates how gains in predictive modeling can be used to estimate important quantities in the social sciences.

This thesis develops not only general machine learning methods for social science research but also techniques tailored to specific applications in political science and labor economics. These applications are bolstered by marrying domain-specific ideas with general approaches from machine learning. The remainder of the thesis is organized as follows:

- In Chapter 2, we develop a method for explaining the predictions of general, black-box sequence models. We describe a combinatorial optimization to find rationales, or subsequences of inputs, that can explain individual model predictions. Focusing on applications in natural language processing (NLP), we demonstrate that our method can efficiently summarize the decision-making processes of large language models and uncover explanations that are similar to human annotators. The method in this chapter is based on work developed with Yuntian Deng, David Blei, and Sasha Rush.

- In Chapter 3, we narrow our scope to a single domain: political text. We propose the text-based ideal point model (TBIP), a Bayesian model that can quantify political positions from text. Underlying Bayesian methods are generative models where latent variables correspond to real-world phenomena and intuitions. When these methods are combined with expressive machine learning models, complex phenomena are distilled into meaningful variables. Our model combines the ideal point model—a technique developed in the political science

literature to model roll-call data—with a matrix factorization method from machine learning to infer interpretable structure from text. The method in this chapter is based on work developed with Suresh Naidu and David Blei.

- In Chapter 4, we focus on another domain: labor economics. We adapt transformer models from NLP to model career histories. These models typically require large-scale data to make good predictions, yet the survey datasets economists fit occupation models to are small. We leverage passively-collected labor sequence data via transfer learning, enabling these large models to make effective predictions on small survey datasets. The method in this chapter is based on work developed with Emil Palikot, Tianyu Du, Ayush Kanodia, Susan Athey, and David Blei.

- In Chapter 5, we use the model developed in Chapter 4 to estimate an important quantity in labor economics: the history-adjusted gender wage gap. We propose a constrained optimization technique to adapt transformers to estimate the adjusted gender wage gap. We then analyze the extent to which history explains the wage gap across subpopulations. The method in this chapter is based on work developed with David Blei and Susan Athey.

- We conclude by summarizing the contributions of this thesis and proposing future directions.

Chapter 2: Rationales for Sequential Predictions

Many important datasets in the social sciences are comprised of sequences. A legal opinion, for example, is a sequence of words. A work history is a sequence of jobs. An online educational curriculum is a sequence of lessons and assignments. Machine learning methods designed for sequential data offer promise for social scientists seeking to analyze and model this information. However, while machine learning models can be trained to make accurate predictions on sequential data, their decision making processes are hard to explain. Explaining model predictions is crucial for social scientists wishing to validate their models and gain insights from the models fit to their data.

In this chapter, we develop a method for interpreting the predictions of general sequence models. Here, we focus on interpreting language models, i.e. sequential models of text. Later, in Chapter 4, we show how this method can also be used for labor economics applications to interpret models of job sequences.

2.1 Introduction

Sequence models are a critical component of natural language processing (NLP) applications ranging from language modeling (Radford et al., 2019) to machine translation (Brown et al., 1993; Vaswani et al., 2017) to summarization (Rush et al., 2015). These tasks are dominated by complex neural networks, whose decision making processes are opaque and difficult to interpret. Interpreting a model's predictions is important in a variety of settings: a researcher needs to understand a model to debug it; a doctor using a diagnostic model requires justifications to validate a decision; a company deploying a language model relies on model explanations to detect biases appropriated from training data.

Interpretation takes many flavors (Lipton, 2018). In this chapter, we focus on *rationales*, i.e.

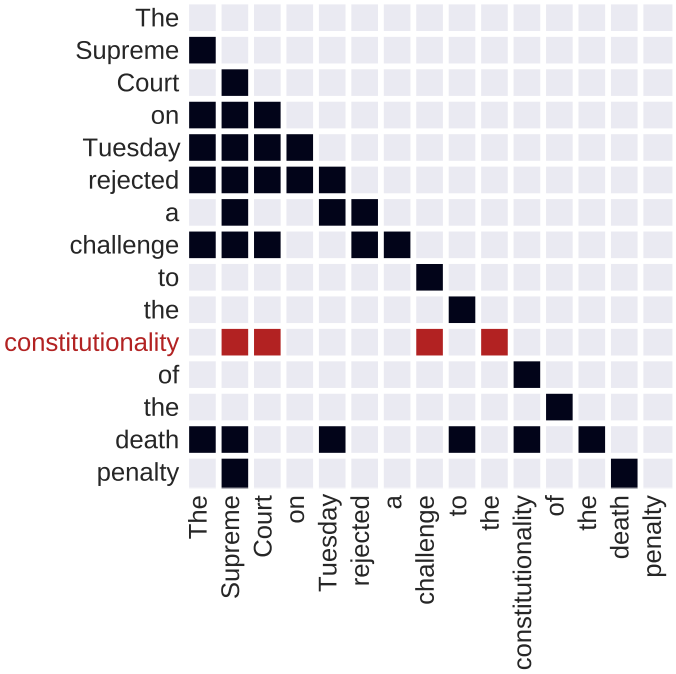


Figure 2.1: Rationales for sequential prediction on GPT-2 (Radford et al., 2019). Each row is a predicted word. The dark cells correspond to the context words found by greedy rationalization. To predict "constitutionality", the model only needs "Supreme", "Court", "challenge", and "the".

identifying the most important subset of inputs that leads to a model's prediction. For example, assume a language model generated the sentence: "The Supreme Court on Tuesday rejected a challenge to the constitutionality of the death penalty." Suppose we would like to explain the decision of the model to generate "constitutionality". While the model mathematically conditions on all the previous words, only some are necessary for its predictions. In this case, the rationale produced by the method in this chapter includes "the", "challenge", and notably "Supreme Court", but not phrases that add no information like "on Tuesday" (Figure 2.1).

Various rationale methods have been proposed for sequence classification (Lei et al., 2016; Chen et al., 2018; Jain et al., 2020). These methods were developed for classification models, where the goal is to model a single outcome rather than entire sequences. Thus, explaining model decisions requires only rationalizing a single prediction for each sequence. However, these methods cannot scale to sequence models, where each token in a sequence requires a different rationale.

This chapter frames the problem of finding rationales for a sequence model as a combinatorial

optimization: given a model, the best rationale is the smallest subset of input tokens that would generate the same token as the full sequence. Finding the global optimum in this setting is intractable, so we propose **greedy rationalization**, a greedy algorithm that iteratively builds longer rationales. This approach is efficient for many NLP models such as transformers (Vaswani et al., 2017). However, the method is general and can be applied to any sequence model.

Underlying this approach is an assumption that the model we wish to understand forms sensible predictions for incomplete subsets of the input. Although we can pass in incomplete subsets to models, there is no guarantee that their predictions on these subsets will be compatible with their predictions on full contexts (Arnold & Press, 1989). We show that compatibility can be learned by conditioning on randomly sampled context subsets while training a model. For large pretrained language models like GPT-2 (Radford et al., 2019), fine-tuning is sufficient.

In an empirical study, we compare greedy rationalization to various gradient- and attention-based explanation methods on language modeling and machine translation. Greedy rationalization best optimizes the objective, and its rationales are most faithful to the inner workings of the model. We additionally create a new dataset of annotated rationales based on the Lambada corpus (Paperno et al., 2016). We find that greedy rationales are most similar to human annotations, both on our dataset and on a labeled dataset of translation alignments. Our code and annotated dataset are available.[1]

2.2 Sequential Rationales

Consider a sequence of tokens, $y_{1:T}$, generated by some unknown process $y_{1:T} \sim F$. Each term in the sequence represents a different item; for example, y_t can represent an occupation an individual has at time t, or a discrete object like a product added to a shopping cart. In this chapter, we focus on language, where each y_t is a word, although our paradigm can extend to arbitrary discrete sequences. Each word can take one of V values.

The goal of sequence modeling is to learn a probabilistic model p_θ, governed by parameters

[1] `https://github.com/keyonvafa/sequential-rationales`

θ, that approximates F from samples. Maximum-likelihood estimation is an effective way to train these models, where the parameters θ are fit according to

$$\arg\max_{\theta} \mathbb{E}_{y_{1:T}\sim F}[\log p_\theta(y_{1:T})]. \tag{2.1}$$

For NLP applications, sequence models are typically factored into conditional distributions:

$$p_\theta(y_{1:T}) = f_\theta(y_1)\prod_{t=2}^{T} f_\theta(y_t|y_{<t}), \tag{2.2}$$

where $y_{<t} = \{y_1, \ldots, y_{t-1}\}$ represents the sequence up to position t, and f_θ is the specific model parameterizing p_θ, such as a transformer (Vaswani et al., 2017). Going forward, we drop the dependence on θ in the notation.

More specifically, f is a function that takes as input an ordered set of tokens and outputs a vector of probabilities. For an index set $\mathcal{S}$ containing t tokens,

$$f : (y_{\mathcal{S}_1}, \ldots, y_{\mathcal{S}_t}) \to \Delta_V, \tag{2.3}$$

where Δ_V denotes the V-simplex. To denote the probability of a specific token, y_t, given all tokens beforehand, we use the shorthand $f(y_t|y_{<t})$. Since f is a set function, it can also take as input the empty set; in this case, we denote the probability of a specific token y_1 as $f(y_1) = f(y_1|\emptyset)$.[2]

Word-level explanations are a natural way to interpret a sequence model: which words were instrumental for predicting a particular word? Would the same word have been predicted if some of the words had been missing? Explanations may be straightforward for simpler models; for example, a bigram Markov model uses only the previously generated word to form predictions. However, the most effective sequence models have been based on neural networks, whose predictions are challenging to interpret (Lipton, 2018).

Motivated by this goal, we consider a sequence $y_{1:T}$ generated by a sequence model p. At

[2]The method in this chapter is not specific to any model, so detailed knowledge of transformers is not required. However, see Chapter 4 for a full description.

each position t, the model takes the inputs in the context $y_{<t}$ and uses them to predict y_t. We are interested in forming *rationales*: subsets of the contexts that explain the model's prediction of y_t.[3]

What are the properties of a good rationale? Any of the contextual words $y_{<t}$ can contribute to y_t. However, if a model makes the same prediction with only a subset of the context, that subset contains explanatory power on its own. A rationale is *sufficient* if the model would produce the same y_t having seen only the rationale (DeYoung et al., 2020). While rationales consisting of the full context would always be sufficient, they would be ineffective for explaining longer sequences. Intuitively, the smaller the rationale, the easier it is to interpret, so we also prioritize *brevity*.

We combine these desiderata and frame finding rationales as a combinatorial optimization: the best rationale of a word y_t is the smallest subset of inputs that would lead to the same prediction as the full history. Each candidate rationale S is an index set, and y_S denotes the subset of tokens indexed by S. Denote by $\mathcal{S} = 2^{[t-1]}$ the set of all possible context subsets. An optimal rationale is given by

$$\arg\min_{S \in \mathcal{S}} |S| \quad \text{s.t.} \quad \arg\max_{y'_t} p(y'_t | y_S) = y_t. \tag{2.4}$$

The constraint guarantees sufficiency, and the objective targets brevity. Although the objective may have multiple solutions, we only require one.

Optimizing Equation 2.4 is hindered by a pair of computational challenges. The first challenge is that solving this combinatorial objective is intractable; framed as a decision problem, it is NP-hard. We discuss this challenge in Section 2.3. The second challenge is that evaluating distributions conditioned on incomplete context subsets $p(y'_t | y_S)$ involves an intractable marginalization over missing tokens. For now we assume that $f(y'_t | y_S) \approx p(y'_t | y_S)$; we discuss how to enforce this condition in Section 2.4.

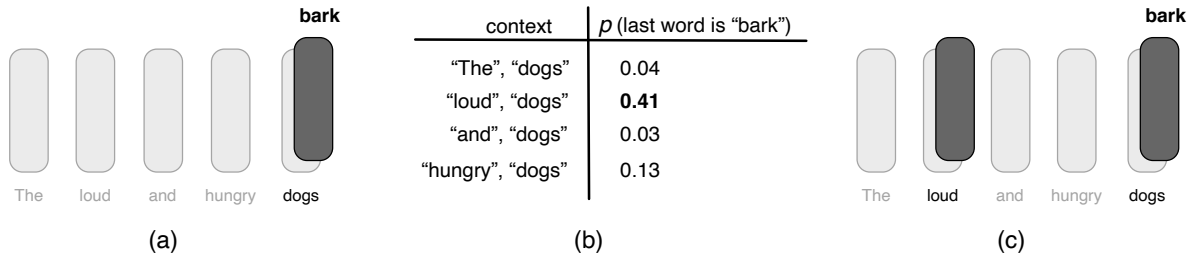


Figure 2.2: One step of greedy rationalization for a language model that generated the word "bark" from the context "The loud and hungry dogs". In (a), the rationale so far is a single word, "dogs". In (b), each candidate token is considered and "loud" results in the best probability for "bark". In (c), the token "loud" is added to the rationale. This process repeats until the most likely word is the model's original prediction.

2.3 Greedy Rationalization

We propose a simple greedy algorithm, **greedy rationalization**, to approximate the solution to Equation 2.4. The algorithm starts with an empty rationale. At each step, it considers adding each possible token, and it selects the one that most increases the probability of y_t. This process is repeated until the rationale is sufficient for predicting y_t. Figure 2.2 provides an overview.

Here is the algorithm. Begin with a rationale $S^{(0)} = \emptyset$. Denoting by $[t-1] = \{1, \ldots, t-1\}$, the first rationale set is

$$S^{(1)} = \arg\max_{k \in [t-1]} p(y_t | y_k). \tag{2.5}$$

At each step, we iteratively add a single word to the rationale, choosing the one that maximizes the probability of the word y_t:

$$S^{(n+1)} = S^{(n)} \cup \arg\max_{k \in [t-1] \setminus S^{(n)}} p(y_t | y_{S^{(n)} \cup k}). \tag{2.6}$$

We continue iterating Equation 2.6 until $\arg\max_{y'_t} p(y'_t | y_{S^{(n)}}) = y_t$. The procedure will always converge, since in the worst case, $S^{(t-1)}$ contains the full context.

The greedy approach is motivated by approximations to the set cover problem (Chvatal, 1979).

[3] Our paradigm and method extend easily to conditional sequence models, such as those used for machine translation. For full details, refer to Section A.1.

In our setting, each set is a single context token, and a rationale "covers" a sequence if it results in predicting the same token.

This procedure is simple to implement, and it is black-box: it does not require access to the inner workings of a model, like gradients or attention.

While greedy rationalization can be applied to any model, the algorithm is particularly effective for set-based models such as transformers. This is because the computational complexity of set-based models scales with the size of inputs. Since greedy rationalization only requires evaluations of input subsets, the full procedure can be performed efficiently. In fact, if the final rationale size $m = |S|$ is significantly shorter than the size of the context t, greedy rationalization requires no extra asymptotic complexity beyond the cost of a single evaluation.

For example, for transformers, the complexity of each evaluation $f(y_t|y_{<t})$ is quadratic in the input set $O(t^2)$. Each step of greedy rationalization requires evaluating $f(y_t|y_S)$, but y_S can be significantly smaller than $y_{<t}$. A rationale of size m will require m steps of $O(t)$ evaluations to terminate, resulting in a total complexity of $O(m^3t)$. As long as $m = O(t^{1/3})$, greedy rationalization can be performed with the same asymptotic complexity as evaluating a transformer on the full input, $O(t^2)$. In Section A.3, we verify the real-world efficiency of greedy rationalization.

2.4 Model Compatibility

Greedy rationalization requires computing conditional distributions $p(y_t|y_S)$ for arbitrary subsets S. Using an autoregressive model, this calculation requires marginalizing over unseen positions. For example, rationalizing a sequence $y_{1:3}$ requires evaluating the candidate rationale $p(y_3|y_1)$, which marginalizes over the model's predictions:

$$p(y_3|y_1) = \sum_k f(y_3|y_1, y_2 = k) f(y_2 = k|y_1).$$

Given the capacity of modern neural networks, it is tempting to pass in incomplete subsets y_S to f and evaluate this instead as $f(y_t|y_S) \approx p(y_t|y_S)$. However, since f is trained only on

complete feature subsets $y_{<t}$, incomplete feature subsets y_S are out-of-distribution (Hooker et al., 2019). Evaluating $f(y_3|y_1)$ may be far from the true conditional $p(y_3|y_1)$. In Figure 2.4, we show that indeed language models like GPT-2 produce poor predictions on incomplete subsets.

2.4.1 Fine-tuning for Compatibility

Ideally $f(y_t|y_S)$ approximates $p(y_t|y_S)$, a property known as *compatibility* (Arnold & Press, 1989). Training with Equation 2.1 only evaluates f on complete contexts $y_{<t}$, so while the optimal $f(y_t|y_{<t})$ approximates $p(y_t|y_{<t})$ the model is never trained to make predictions $f(y_t|y_S)$ for incomplete subsets y_S. Thus, its behavior on incomplete subsets is unspecified.

However, compatibility can be obtained by training to maximize

$$\mathbb{E}_{y_{1:T}\sim F}\,\mathbb{E}_{S\sim\text{Unif}(\mathcal{S})}\left[\sum_{t=1}^{T}\log f(y_t|y_{S_{<t}})\right], \tag{2.7}$$

where $S \sim \text{Unif}(\mathcal{S})$ indicates sampling word subsets uniformly at random from the power set of all possible word subsets, and $S_{<t}$ denotes the indices in S that are less than t. Intuitively, this objective forces the model to make predictions using incomplete contexts rather than always passing in full contexts; if the model sees incomplete contexts while training, it can approximate arbitrary incomplete distributions. Jethani et al. (2021) show that the optimum of Equation 2.7 is the distribution whose conditional distributions are all equal to the ground-truth conditionals.

We approximate Equation 2.7 with word dropout. In practice, we combine this objective with standard MLE training to learn compatible distributions while maintaining the performance of the original model. The word dropout distribution in Equation 2.7 is heavily skewed towards contexts containing half the words in the sequence. To alleviate this problem, we modify the word dropout distribution to sample subsets of varying lengths; see Section A.4.

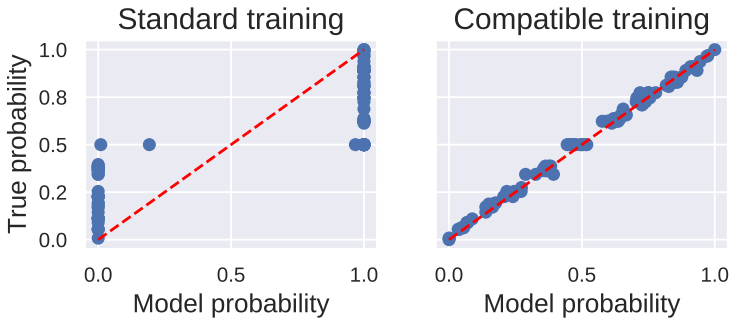


Figure 2.3: Training with word dropout (right) results in compatible predictions for the majority-class synthetic language. The optimal compatibility is the dashed line.

2.4.2 Compatibility Experiments

To demonstrate the impact of training with the compatibility objective in Equation 2.7, we consider a synthetic language where the ground truth conditional probabilities are known. Specifically, we use a majority-class language over binary strings of 19 tokens. The first 17 tokens are sampled uniformly from $\{0, 1\}$, and the 18th token is always '='. The 19th token is 0 if there are more 0's than 1's in the first 17 tokens, and 1 otherwise.

We train two models: one using the standard objective in Equation 2.1, the other using word dropout to optimize Equation 2.7. Although both models have the same heldout perplexity on the full context, training with Equation 2.7 is required to form compatible predictions on incomplete subsets. In Figure 2.3, we provide both models with random subsets S and calculate each model's probability that the last token is 1. A model that has only seen a few tokens should be less confident about the prediction of the final majority class, yet models trained without word dropout ignore this uncertainty. Meanwhile, the predictions from the model trained with word dropout are well-calibrated across contexts.

Models do not need to be trained from scratch with Equation 2.7. A model can be pre-trained with Equation 2.1, after which it can be fine-tuned for compatibility. As an example, when GPT-2 (Radford et al., 2019), a large pretrained language model, is not trained with word dropout, it makes insensible predictions for out-of-distribution sequences. For a sequence that contains only the token "the", GPT-2 is trained to give reasonable predictions for $p(y_2|y_1 =$ "the"). But when

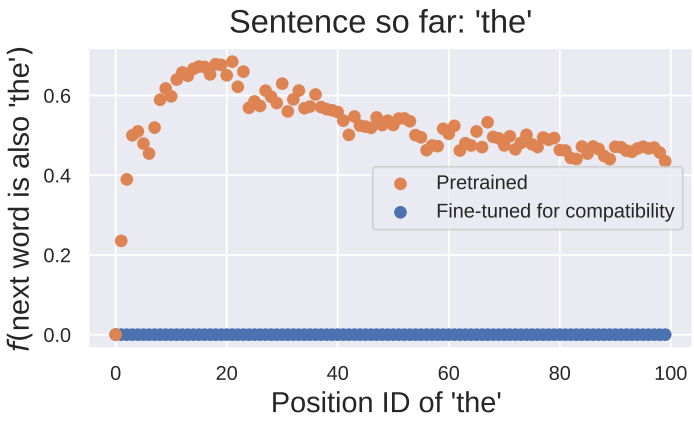


Figure 2.4: Fine-tuning GPT-2 for compatibility removes pathological repeating on incomplete contexts. For a position t, the vertical axis gives $f(y_{t+1} = \text{“the”} | y_t = \text{“the”})$.

it has only seen the token “the” somewhere besides the first position of the sequence, the top prediction for the word after “the” is also “the”.[4] Of course, following “the” with “the” is not grammatical. Fine-tuning for compatibility alleviates this problem (Figure 2.4).

Finally, we find that that fine-tuning for compatibility does not hurt the heldout performance of the complete conditional distribution of each fine-tuned model (see Section A.4).

2.5 Connection to Classification Rationales

In this section, we go over related rationalization approaches developed for classification and discuss why they cannot scale to sequence models. We also show that the combinatorial rationale objective in Equation 2.4 is a global solution to a classification rationale-style objective.

In classification problems, a sequence $x_{1:T}$ is associated with a label y. Rationale methods are commonly used in this setting (Lei et al., 2016; Chen et al., 2018; Yoon et al., 2018; Bastings et al., 2019; Jain et al., 2020; Jethani et al., 2021). The most common approach uses two models: one, a selection model $q(S|x_{1:T})$, provides a distribution over possible rationales; the other, the predictive model $p(y|x_S)$, makes predictions using only samples from the former model. Typically, p and q

[4] We represent “the” at various positions by changing the positional encoding passed into the transformer.

are both optimized to maximize

$$\mathbb{E}_{x,y\sim F}\,\mathbb{E}_{S\sim q(S|x,y)}[\log p(y|x_S) - \lambda|S|]. \tag{2.8}$$

Here, F is the ground truth, unknown data distribution, and λ is a regularizing penalty that encourages smaller rationales.

In practice, it is infeasible to adopt this objective for sequence models. Equation 2.8 is centered on providing classification models with only the words in a sequence's rationale. In sequential settings, each word has a different rationale. Since sequence models make T predictions per sequence and are trained by sharing all T word representations, each token would be indirectly exposed to words in the rationales of the words it is allowed to use. A remedy would be to train sequence models without sharing representations, but this is computationally infeasible; it requires $O(T^3)$ computations per sequence for transformer architectures.

Most classification rationale methods treat $q(S|x_{1:T})$ as a probability distribution over all possible rationales. However, the q that maximizes Equation 2.8 is deterministic for any p. To see this, note that q does not appear inside the expectation in Equation 2.8, so it can place all its mass on a single mode. We provide a formal justification in Section A.2.

Since the optimal selection model q is a point-mass, the optimal rationale can be written as

$$\arg\min_{S\in\mathcal{S}} \ \lambda|S| - \log p(y|x_S). \tag{2.9}$$

This optimization is identical to the combinatorial optimization in Equation 2.4, albeit with a soft constraint on the rationale's prediction: the true label y is not required to be the maximum of $p(y'|x_S)$. In practice, this soft constraint sometimes results in empty rationales (Jain et al., 2020). Since we view sufficiency as a key component of a good rationale, Equation 2.4 imposes a hard constraint on the rationale's prediction.

2.6 Related Work

Finding rationales is similar to feature selection. While global feature selection has been a well-studied problem in statistics (Guyon & Elisseeff, 2003; Hastie et al., 2009), instance-wise feature selection — where the goal is selecting features per-example — is a newer research area (Chen et al., 2018). We review local explanation methods used for NLP.

Gradient saliency. Gradient-based saliency methods have long been used as a measure of feature importance in machine learning (Baehrens et al., 2010; Simonyan et al., 2013; Li et al., 2016a). Some variations involve word embeddings (Denil et al., 2014); integrated gradients, to improve sensitivity (Sundararajan et al., 2017); and relevance-propagation to track each input's contribution through the network (Bach et al., 2015; Voita et al., 2021).

But there are drawbacks to using gradient-based methods as explanatory tools. Sundararajan et al. (2017) show that in practice, gradients are *saturated*: they may all be close to zero for a well-fitted function, and thus not reflect importance. Adversarial methods can also distort gradient-based saliences while keeping a model's prediction the same (Ghorbani et al., 2019; Wang et al., 2020). We compare greedy rationalization to gradient saliency methods in Section 2.8.

Attention. Recently, NLP practitioners have focused on using attention weights as explanatory tools. The literature has made a distinction between *faithfulness* and *plausibility*. An explanation is faithful if it accurately depicts how a model makes a decision (Jacovi & Goldberg, 2020); an explanation is plausible if it can be understood and interpreted by humans (Wiegreffe & Pinter, 2019). Practitioners have shown that attention-based explanations are generally not faithful (Jain & Wallace, 2019; Serrano & Smith, 2019), but that they may be plausible (Wiegreffe & Pinter, 2019; Mohankumar et al., 2020; Vashishth et al., 2019). Others show that attention weights should not be interpreted as belonging to single tokens since they mix information across tokens (Brunner et al., 2019; Kobayashi et al., 2020). Bastings & Filippova (2020) argue that general input saliency measures, such as gradients, are better suited for explainability than attention. We compare greedy

rationalization to attention-based methods in Section 2.8.

Local post-hoc interpretability. Another class of methods provides local interpretability for pretrained models. These approaches aim to explain a model's behavior for a single example or for a small subset of inputs. LIME (Ribeiro et al., 2016) trains an interpretable model that locally approximates the pretrained model. Alvarez-Melis & Jaakkola (2017) learn a causal relationship between perturbed inputs and their model outputs. These methods impose no constraints on the pretrained model. However, they are expensive – they require training separate models for each input region. In contrast, the method proposed here, greedy rationalization, can efficiently explain many predictions.

Input perturbation. Practitioners have also measured the importance of inputs by perturbing them (Zeiler & Fergus, 2014; Kádár et al., 2017). Occlusion methods (Li et al., 2016b) replace an input with a baseline (e.g. zeros), while omission methods (Kádár et al., 2017) remove words entirely. Li et al. (2016b) propose a reinforcement learning method that aims to find the minimum number of occluded words that would change a model's prediction. Feng et al. (2018) use gradients to remove unimportant words to see how long it takes for the model's prediction to change. They find that the remaining words are nonsensical and do not comport with other saliency methods. Others have shown that input perturbation performs worse than other saliency methods in practice (Poerner et al., 2018). These methods have mostly focused on subtractive techniques. For this reason, they are inefficient and do not aim to form sufficient explanations. In contrast, greedy rationalization efficiently builds up sufficient explanations.

2.7 Experimental Setup

There are two goals in our empirical studies for this chapter. The first is to compare the ability of greedy rationalization to other approaches for optimizing the combinatorial objective in Equation 2.4. The second is to assess the quality of produced rationales.

We measure the quality of rationales using two criteria: faithfulness and plausibility. An explanation is faithful if it accurately depicts how a model makes a decision (Jacovi & Goldberg, 2020); an explanation is plausible if it can be understood and interpreted by humans (Wiegreffe & Pinter, 2019). Although sufficiency is a standard way to measure faithfulness (DeYoung et al., 2020), all the rationales that satisfy the constraint of Equation 2.4 are sufficient by definition. To measure plausibility, we compare rationales to human annotations. Since there do not exist language modeling datasets with human rationales, we collected annotations based on Lambada (Paperno et al., 2016). The annotated dataset is available online, along with the code used for all experiments.[5]

We compare greedy rationalization to a variety of gradient- and attention-based baselines (see Section 2.6). To form baseline sequential rationales, we add words by the order prescribed by each approach, stopping when the model prediction is sufficient. The baselines are: l_2 gradient norms of embeddings (Li et al., 2016a), embedding gradients multiplied by the embeddings (Denil et al., 2014), integrated gradients (Sundararajan et al., 2017), attention rollout (Abnar & Zuidema, 2020), the last-layer transformer attention weights averaged-across heads, and all transformer attentions averaged across all layers and heads (Jain et al., 2020).

To compare rationale sets produced by each method to those annotated by humans, we use the set-similarity metrics described in DeYoung et al. (2020): the intersection-over-union (IOU) of each rationale and the human rationale, along with the token-level F1, treating tokens as binary predictions (either in the human rationale or out of it).

We use transformer-based models for all of the experiments. We fine-tune each model for compatibility using a single GPU. For model and fine-tuning details, refer to Section A.4.

2.8 Results and Discussion

The experiments test sequential rationales for language modeling and machine translation. Section C.9 contains full details for each experiment.

[5] `https://github.com/keyonvafa/sequential-rationales`

2.8.1 Language Modeling

Long-Range Agreement. The first study tests whether rationales for language models can capture long-range agreement. We create a template dataset using the analogies from Mikolov et al. (2013a). This dataset includes word pairs that contain either a semantic or syntactic relationship. For each type of relationship, we use a predefined template. It prompts a language model to complete the word pair after it has seen the first word.

For example, one of the fifteen categories is countries and their capitals. We can prompt a language model to generate the capital by first mentioning a country and then alluding to its capital. To test long-range agreement, we also include a distractor sentence that contains no pertinent information about the word pair. For example, our template for this category is,

> When my flight landed in **Japan**, I converted my currency and slowly fell asleep. (I had a terrifying dream about my grandmother, but that's a story for another time). I was staying in the capital, _____

Here, the parenthetical clause is a distractor sentence, since it contains no relevant information about predicting the capital of Japan. The correct capital, "Tokyo", is predicted by GPT-2 both with and without the distractor. We use this template for all of the examples in the country capital category, swapping the antecedent "Japan" for each country provided in Mikolov et al. (2013a).

We feed the prompts to GPT-2, which completes each analogy. To measure faithfulness, we calculate the percent of rationales that contain the true antecedent, and the percent of rationales that do not contain any words in the distractor. We only use examples where the prediction is the same both with and without the distractor. We also perform exhaustive rationale search on the objective in Equation 2.4 to find the global optima. This search is highly inefficient, so we only complete it for 40 examples. To measure the approximation ratio, we divide the size of the rationale found by each method by the exhaustive rationale size.

Table 2.1 contains the results on the compatible model. Although all methods contain the true antecedents in their rationales, greedy rationalization has by far the least distractors in its rationales.

	Length	Ratio	Ante	No D
Grad norms	22.5	4.1	**1.0**	0.06
Grad x emb	38.0	7.4	0.99	0.01
Integrated grads	28.1	5.2	0.99	0.00
Attention rollout	36.9	7.1	**1.0**	0.12
Last attention	16.7	2.9	0.99	0.13
All attentions	14.5	2.6	**1.0**	0.02
Greedy	**7.1**	**1.2**	**1.0**	**0.43**

Table 2.1: Language modeling faithfulness on long-range agreement with templated analogies. "Ratio" refers to the approximation ratio of each method's rationale length to the exhaustive search minimum. "Ante" refers to the percent of rationales that contain the true antecedent. "No D" refers to the percent of rationales that do not contain any tokens from the distractor.

The rationales are also universally shorter for greedy rationalization and closer to the optimal rationales, justifying our greedy assumption. To show that fine-tuning GPT-2 for compatibility is not hurting the baselines, we also perform the baseline methods on a pretrained GPT-2 without fine-tuning; see Section C.9.

Annotated Rationales. To test the plausibility of rationales for language models, we collect a dataset of human annotations. We base the collection on Lambada (Paperno et al., 2016), a corpus of narrative passages. Each passage included in Lambada is chosen so that humans need to use both local and global context to reliably predict the final word. By its construction it is guaranteed to have non-trivial rationales.

Our goal is to collect rationales that are both minimal and sufficient for humans. We run an annotation procedure with two roles: a selector and a predictor. First, the selector sees the full passage and ranks the words in order of how informative they are for predicting the final word. Next, the predictor sees one word at a time chosen by the selector, and is asked to predict the final word of the passage. The words the predictor saw before guessing the correct word form a human rationale. This rationale selection method is inspired by Rissanen Data Analysis (Rissanen, 1978; Perez et al., 2021), which uses a minimum description length metric to estimate feature importances. We rely on human annotators to estimate information gains.

Since it could be trivial for humans to predict the final word if it also appears in the context,

	Length	IOU	F1
Gradient norms	60.2	0.14	0.22
Gradient x embedding	68.3	0.12	0.21
Integrated gradients	62.8	0.12	0.21
Attention rollout	73.9	0.11	0.19
Last attention layer	54.6	0.15	0.25
All attention layers	48.7	0.20	0.28
Greedy	**17.9**	**0.25**	**0.35**

Table 2.2: Language modeling plausibility on rationale-annotated Lambada.

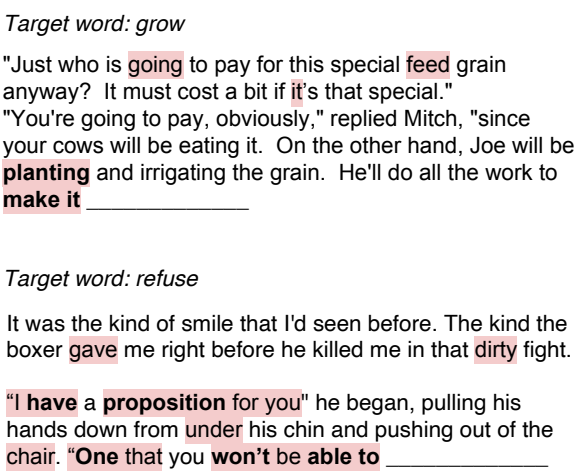


Figure 2.5: Examples from our annotated Lambada dataset. Highlighted text denotes greedy rationales, and **bolded text** denotes human-annotated rationales.

we only include examples that do not repeat a word. We collect annotations for 107 examples, which we also release publicly. We use two sets of annotators for 15% of the examples in order to compute inter-annotator agreement. On this subset, the average token-level Cohen's κ is 0.63 (Cohen, 1960), indicating substantial agreement.

We compare the rationales produced by each method to the annotated rationales. Table 2.2 shows that the greedy rationales are most similar to the human-annotated rationales. Greedy rationalization is also the most effective at minimizing the combinatorial objective in Equation 2.4, as its rationales are by far the shortest. Figure 2.5 contains examples of rationales for this dataset.

It is worth noting that the top few words added by the baselines are quite relevant; after 5 tokens, the all-attention baseline has a better F1 and IOU than greedy rationalization. However,

the baselines struggle to form sufficient rationales, which hurts their overall performance.

2.8.2 Machine Translation

Greedy rationalization can also be applied to conditional sequence models, such as those used for machine translation. Here, we demonstrate the effectiveness of greedy rationalization for machine translation in two separate empirical studies. In the first, we measure the faithfulness of the rationales produced by each method; in the latter, we measure the plausibility of the rationales by comparing to human-labeled word alignments.

Distractors. To measure faithfulness of machine translation, we use IWSLT14 De-En, a widely-used machine translation dataset from German to English. We train a transformer (and fine-tune it for compatibility), after which we generate translations for 1000 source sequences from the test set. We then create a corpus by concatenating random example pairs; for two sampled pairs of source and target sequences, (S_1, T_1) and (S_2, T_2), we create a new example $(S_1 S_2, T_1 T_2)$. Each token in T_1 is generated from S_1 alone, so its rationales shouldn't contain any tokens from S_2. Similarly, T_2 is generated from S_2 alone, so its rationales shouldn't contain any tokens from S_1 or T_1.

We evaluate each rationale by counting how many times it has crossed over: a rationale for T_1 crosses over every time it contains a token in S_2, and a rationale for T_2 crosses over every time it contains a token in S_1 or T_1 (since the model is autoregressive, T_1's rationales can never contain tokens from T_2).

Table 2.3 contains the results. Greedy rationalization has by far the fewest average number of crossovers per rationale. Although the percent of source rationales that cross over is slightly higher than the percent using gradient norms, the percentage on the target side is superior.

Annotated Alignments. To test plausibility, we compare the rationales to word alignments (Brown et al., 1993). Using a dataset containing 500 human-labeled alignments for German-English trans-

	Mean Crossovers		Crossover Rate	
	Source	Target	Source	Target
Grad norms	0.40	0.44	**0.06**	0.06
Grad x emb	6.25	5.57	0.42	0.41
Integrated grads	2.08	1.68	0.23	0.14
Last attention	0.63	2.41	0.09	0.24
All attentions	0.58	0.80	0.08	0.12
Greedy	**0.12**	**0.12**	0.09	**0.02**

Table 2.3: Translation faithfulness with distractors. "Mean crossovers" refers to the average number of crossovers per rationale, and "Crossover rate" refers to the fraction of rationales that contain at least one crossover.

	Length	AER	IOU	F1	Top1
Grad norms	10.2	0.82	0.30	0.16	0.63
Grad x emb	13.2	0.90	0.16	0.12	0.40
Integrated grads	11.3	0.85	0.24	0.14	0.42
Last attention	10.8	0.84	0.27	0.15	0.59
All attentions	10.7	0.82	0.32	0.15	**0.66**
Greedy	**4.9**	**0.78**	**0.40**	**0.24**	0.64

Table 2.4: Translation plausibility with annotated alignments. The first four columns correspond to using the full source rationale found by each method; the last column "Top1" refers to the accuracy of the first source token added by each method. AER refers to alignment error rate.

lation,[6] we compute rationales for each method using the ground truth targets. We measure similarity to the labeled rationales by computing alignment error rate (AER) (Och & Ney, 2000), along with computing the IOU and F1 between sets. To separate the requirement that the rationale be sufficient from each method's global ordering of tokens, we also compare top-1 accuracies, which measure whether the top token identified by each baseline is present in the labeled alignment set.

Table 2.4 contains the results. The rationales learned by greedy rationalization are more similar to human-labeled alignments than those provided by gradient and attention methods. Many methods have similar top-1 accuracies — indeed, the best top-1 accuracy comes from averaging all attention layers. This reinforces the notion that although the baselines may be able to capture first-order information, they struggle to form sufficient rationales. Figure 2.6 contains an example of greedy rationalization applied to machine translation, along with the human-labeled alignments.

[6] `https://www-i6.informatik.rwth-aachen.de/goldAlignment/`

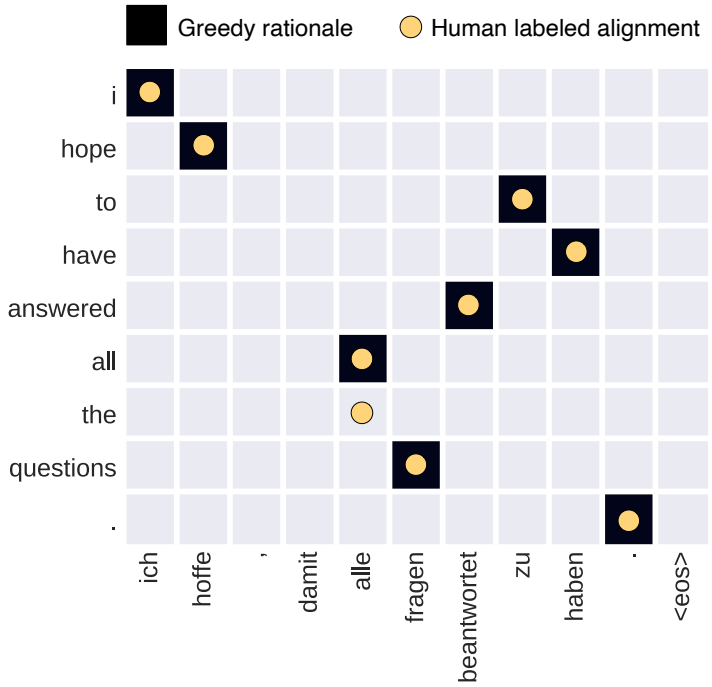


Figure 2.6: Greedy rationalization for machine translation. Each row depicts the source words contained in a rationale. Although each rationale includes both source and target words, here we only show source-side rationales so they can be compared to annotated alignments.

2.9 Summary

We proposed an optimization-based algorithm for rationalizing sequence predictions. Although exact optimization is intractable, we developed a greedy approach that efficiently finds good rationales. Moreover, we showed that models can be fine-tuned to form compatible distributions, thereby circumventing an intractable marginalization step. In experiments, we showed that the greedy algorithm is effective at optimization, and that its rationales are more faithful and plausible than those of gradient- and attention-based methods.

While this chapter has focused on applications to text, the method is general and can be applied to any model trained on any type of sequential data. In Chapter 4, we will apply this method to rationalize the predictions of a model trained to model distributions of career histories.

Chapter 3: Text-Based Ideal Points

A primary goal for computational social scientists is developing models that can uncover patterns and behaviors embedded in data. An important aspect of this process is interpreting the predictions of models fit to data; Chapter 2 described one general method for interpreting the predictions of sequence models. However, models would ideally reveal underlying patterns in data without extra machinery.

Bayesian machine learning offers a framework for these problems. Underlying Bayesian methods are generative models where latent variables correspond to real-world phenomena and intuitions. When these methods are combined with expressive machine learning models, complex phenomena are distilled into meaningful variables.

In this chapter, we introduce the text-based ideal point model (TBIP) model, a Bayesian model that can quantify political positions from text, such as tweets and Senate speeches. The model marries a classical technique from political science—the ideal point model—with a Bayesian matrix factorization technique from probabilistic machine learning. Though the model does not analyze votes or political affiliations, it can separate lawmakers by party and learn patterns in political language that vary across topics.

3.1 Introduction

Ideal point models analyze lawmakers' votes and quantify their political behavior with numerical summaries, known as ideal points (Poole & Rosenthal, 1985). These models are widely used to help characterize modern democracies, analyzing lawmakers' votes to estimate their positions on a political spectrum. But votes aren't the only way that lawmakers express political preferences—press releases, tweets, and speeches all help convey political positions. Like votes, these signals

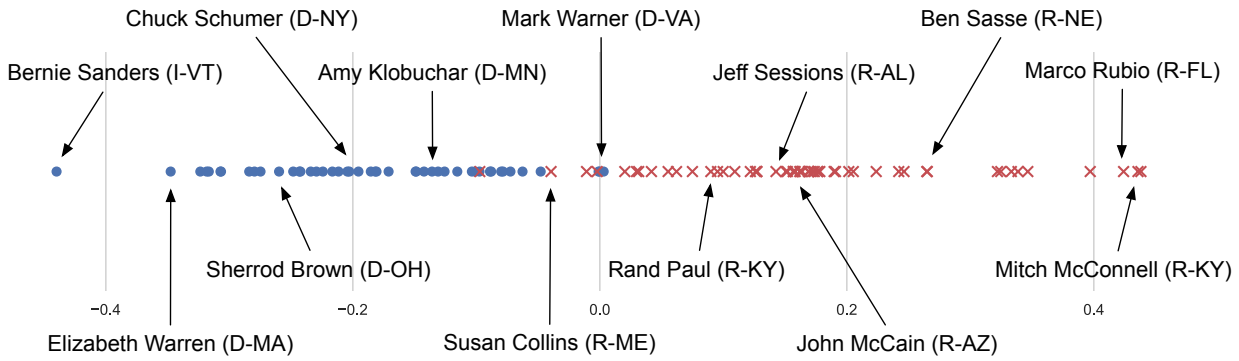


Figure 3.1: The TBIP separates senators by political party using only speeches. The algorithm does not have access to party information, but senators are coded by their political party for clarity (Democrats in blue circles, Republicans in red x's). The speeches are from the 114th U.S. Senate.

are recorded and easily collected.

This chapter develops the text-based ideal point model (TBIP), a probabilistic topic model for analyzing unstructured political text to quantify the political positions of their authors. While classical ideal point models analyze how different people vote on a shared set of bills, the TBIP analyzes how different authors write about a shared set of latent topics. The TBIP is inspired by the idea of political framing: the specific words and phrases used when discussing a topic can convey political messages (Entman, 1993). Given a corpus of political texts, the TBIP estimates the latent topics under discussion, the latent political positions of the authors of texts, and how per-topic word choice changes as a function of the political position of the author.

A key feature of the TBIP is that it is unsupervised. It can be applied to any political text, regardless of whether the authors belong to known political parties. It can also be used to analyze non-voting actors, such as political candidates.

Figure 3.1 shows a TBIP analysis of the speeches of the 114th U.S. Senate. The model lays the senators out on the real line and accurately separates them by party. (It does not use party labels in its analysis.) Based only on speeches, it has found an interpretable spectrum—Senator Bernie Sanders is liberal, Senator Mitch McConnell is conservative, and Senator Susan Collins is moderate. For comparison, Figure 3.2 also shows ideal points estimated from the voting record of the same senators; their language and their votes are closely correlated.

The TBIP also finds latent topics, each one a vocabulary-length vector of intensities, that describe the issues discussed in the speeches. For each topic, the TBIP involves both a neutral vector of intensities and a vector of ideological adjustments that describe how the intensities change as a function of the political position of the author. Illustrated in Table 3.1 are discovered topics about immigration, health care, and gun control. In the gun control topic, the neutral intensities focus on words like "gun" and "firearms." As the author's ideal point becomes more negative, terms like "gun violence" and "background checks" increase in intensity. As the author's ideal point becomes more positive, terms like "constitutional rights" increase.

The TBIP is a bag-of-words model that combines ideas from ideal point models and Poisson factorization topic models (Canny, 2004; Gopalan et al., 2015). The latent variables are the ideal points of the authors, the topics discussed in the corpus, and how those topics change as a function of ideal point. To approximate the posterior, we use an efficient black box variational inference algorithm with stochastic optimization. It scales to large corpora.

We develop the details of the TBIP and its variational inference algorithm. We study its performance on three sessions of U.S. Senate speeches, and we compare the TBIP to other methods for scaling political texts (Slapin & Proksch, 2008; Lauderdale & Herzog, 2016a). The TBIP performs best, recovering ideal points closest to the vote-based ideal points. We also study its performance on tweets by U.S. senators, again finding that it closely recovers their vote-based ideal points. (In both speeches and tweets, the differences from vote-based ideal points are also qualitatively interesting.) Finally, we study the TBIP on tweets by the 2020 Democratic candidates for President, for which there are no votes for comparison. It lays out the candidates along an interpretable progressive-to-moderate spectrum.

3.2 The text-based ideal point model

We develop the text-based ideal point model (TBIP), a probabilistic model that infers political positions from political texts. We first review Bayesian ideal points and Poisson factorization topic models, two probabilistic models on which the TBIP is built.

3.2.1 Background: Bayesian ideal points

Ideal points quantify a lawmaker's political preferences based on their roll-call votes (Poole & Rosenthal, 1985; Jackman, 2001; Clinton et al., 2004). Consider a group of lawmakers voting "yea" or "nay" on a shared set of bills. Denote the vote of lawmaker i on bill j by the binary variable v_{ij}. The Bayesian ideal point model posits scalar per-lawmaker latent variables x_i and scalar per-bill latent variables (α_j, η_j). It assumes the votes come from a factor model,

$$\begin{aligned} x_i &\sim \mathcal{N}(0, 1) \\ \alpha_j, \eta_j &\sim \mathcal{N}(0, 1) \\ v_{ij} &\sim \text{Bern}(\sigma(\alpha_j + x_i \eta_j)). \end{aligned} \tag{3.1}$$

where $\sigma(t) = \frac{1}{1+e^{-t}}$.

The latent variable x_i is called the lawmaker's *ideal point*; the latent variable η_j is the bill's *polarity*. When x_i and η_j have the same sign, lawmaker i is more likely to vote for bill j; when they have opposite sign, the lawmaker is more likely to vote against it. The per-bill intercept term α_j is called the *popularity*. It captures that some bills are uncontroversial, where all lawmakers are likely to vote for them (or against them) regardless of their political positions.

Using data of lawmakers voting on bills, political scientists approximate the posterior of the Bayesian ideal point model with an approximate inference method such as Markov Chain Monte Carlo (MCMC) (Jackman, 2001; Clinton et al., 2004) or expectation-maximization (EM) (Imai et al., 2016). Empirically, the posterior ideal points of the lawmakers separate political parties and capture the spectrum of political preferences in American politics (Poole & Rosenthal, 2000).

3.2.2 Background: Poisson factorization

Poisson factorization is a class of non-negative matrix factorization methods often employed as a topic model for bag-of-words text data (Canny, 2004; Cemgil, 2009; Gopalan et al., 2014). Poisson factorization factorizes a matrix of document/word counts into two positive matrices: a

matrix $\boldsymbol{\theta}$ that contains per-document topic intensities, and a matrix $\boldsymbol{\beta}$ that contains the topics. Denote the count of word v in document d by y_{dv}. Poisson factorization posits the following probabilistic model over word counts, where a and b are hyperparameters:

$$\begin{aligned}
\theta_{dk} &\sim \text{Gamma}(a, b) \quad \text{for all } d, k \\
\beta_{kv} &\sim \text{Gamma}(a, b) \quad \text{for all } k, v \\
y_{dv} &\sim \text{Pois}\left(\textstyle\sum_k \theta_{dk}\beta_{kv}\right).
\end{aligned} \tag{3.2}$$

Given a matrix $\boldsymbol{y}$, practitioners approximate the posterior factorization with variational inference (Gopalan et al., 2015) or MCMC (Cemgil, 2009). Note that Poisson factorization can be interpreted as a Bayesian variant of nonnegative matrix factorization, with the so-called "KL loss function" (Lee & Seung, 1999).

When the shape parameter a is less than 1, the latent vectors θ_d and β_k tend to be sparse. Consequently, the marginal likelihood of each count places a high mass around zero and has heavy tails (Ranganath et al., 2015). The posterior components are interpretable as topics (Gopalan et al., 2015).

3.2.3 The text-based ideal point model

The text-based ideal point model (TBIP) is a probabilistic model that is designed to infer political preferences from political texts.

There are important differences between a dataset of votes and a corpus of authored political language. A vote is one of two choices, "yea" or "nay." But political language is high dimensional—a lawmaker's speech involves a vocabulary of thousands. A vote sends a clear signal about a lawmaker's opinion about a bill. But political speech is noisy—the use of a word might be irrelevant to politics, provide only a weak signal about political positions, or change signal depending on context. Finally, votes are organized in a matrix, where each one is unambiguously attached to a specific bill and nearly all lawmakers vote on all bills. But political language is unstruc-

tured and sparse. A corpus of political language can discuss any number of issues—with speeches possibly involving several issues—and the issues are unlabeled and possibly unknown in advance.

The TBIP is based on the concept of political framing. Framing is the idea that a communicator will emphasize certain aspects of a message — implicitly or explicitly — to promote a perspective or agenda (Entman, 1993; Chong & Druckman, 2007). In politics, an author's word choice for a particular issue is affected by the ideological message they are trying to convey. A conservative discussing abortion is more likely to use terms such as "life" and "unborn," while a liberal discussing abortion is more likely to use terms like "choice" and "body." In this example, a conservative is framing the issue in terms of morality, while a liberal is framing the issue in terms of personal liberty.

The TBIP casts political framing in a probabilistic model of language. While the classical ideal point model infers political positions from the differences in votes on a shared set of bills, the TBIP infers political positions from the differences in word choice on a shared set of topics.

The TBIP is a probabilistic model that builds on Poisson factorization. The observed data are word counts and authors: y_{dv} is the word count for term v in document d, and a_d is the author of the document. Some of the latent variables in the TBIP are inherited from Poisson factorization: the non-negative K-vector of per-document topic intensities is θ_d and the topics themselves are non-negative V-vectors β_k, where K is the number of topics and V is the vocabulary size. We refer to $\boldsymbol{\beta}$ as the *neutral topics*. Two additional latent variables capture the politics: the ideal point of an author s is a real-valued scalar x_s, and the *ideological topic* is a real-valued V-vector η_k.

The TBIP uses its latent variables in a generative model of authored political text, where the ideological topic adjusts the neutral topic—and thus the word choice—as a function of the ideal point of the author. Place sparse Gamma priors on $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, and normal priors on $\boldsymbol{\eta}$ and $\boldsymbol{x}$, so for all documents d, words v, topics k, and authors s,

$$\theta_{dk} \sim \text{Gamma}(a, b) \quad \text{for all } d, k \qquad\qquad \eta_{kv} \sim \mathcal{N}(0, 1) \quad \text{for all } k, v$$

$$\beta_{kv} \sim \text{Gamma}(a, b) \quad \text{for all } k, v \qquad\qquad x_s \sim \mathcal{N}(0, 1) \quad \text{for all } s.$$

These latent variables interact to draw the count of term v in document d,

$$y_{dv} \sim \text{Pois}\left(\sum_k \theta_{dk}\beta_{kv} \exp\{x_{a_d}\eta_{kv}\}\right). \tag{3.3}$$

For a topic k and term v, a non-zero η_{kv} will increase the Poisson rate of the word count if it shares the same sign as the ideal point of the author x_{a_d}, and decrease the Poisson rate if they are of opposite signs. Consider a topic about gun control and suppose $\eta_{kv} > 0$ for the term "constitution." An author with an ideal point $x_s > 0$, say a conservative author, will be more likely to use the term "constitution" when discussing gun control; an author with an ideal point $x_s < 0$, a liberal author, will be less likely to use the term. Suppose $\eta_{kv} < 0$ for the term "violence." Now the liberal author will be more likely than the conservative to use this term. Finally suppose $\eta_{kv} = 0$ for the term "gun." This term will be equally likely to be used by the authors, regardless of their ideal points.

To build more intuition, examine the elements of the sum in the Poisson rate of Equation 3.3 and rewrite slightly to $\theta_{dk} \exp(\log \beta_{kv} + x_{a_d}\eta_{kv})$. Each of these elements mimics the classical ideal point model in Equation 3.1, where η_{kv} now measures the "polarity" of term v in topic k and $\log \beta_{kv}$ is the intercept or "popularity." When η_{kv} and x_{a_d} have the same sign, term v is more likely to be used when discussing topic k. If η_{kv} is near zero, then the term is not politicized, and its count comes from a Poisson factorization. For each document d, the elements of the sum that contribute to the overall rate are those for which θ_{dk} is positive; that is, those for the topics that are being discussed in the document.

The posterior distribution of the latent variables provides estimates of the ideal points, neutral topics, and ideological topics. For example, we estimate this posterior distribution using a dataset of senator speeches from the 114th United States Senate session. The fitted ideal points in Figure 3.1 show that the TBIP largely separates lawmakers by political party, despite not having access to these labels or votes. Table 3.1 depicts neutral topics (fixing the fitted $\hat{\eta}_{kv}$ to be 0) and the corresponding ideological topics by varying the sign of $\hat{\eta}_{kv}$. The topic for immigration shows that

Ideology	Top Words
Liberal	dreamers, dream, undocumented, daca, comprehensive immigration reform, deport, young, deportation
Neutral	immigration, united states, homeland security, department, executive, presidents, law, country
Conservative	laws, homeland security, law, department, amnesty, referred, enforce, injunction
Liberal	affordable care act, seniors, medicare, medicaid, sick, prescription drugs, health insurance, million americans
Neutral	health care, obamacare, affordable care act, health insurance, insurance, americans, coverage, percent
Conservative	health care law, obamacare, obama, democrats, obamacares, deductibles, broken promises, presidents health care
Liberal	gun violence, gun, guns, killed, hands, loophole, background checks, close
Neutral	gun, guns, second, orlando, question, firearms, shooting, background checks
Conservative	second, constitutional rights, rights, due process, gun control, mental health, list, mental illness

Table 3.1: The TBIP learns topics from Senate speeches that vary as a function of the senator's political positions. The neutral topics are for an ideal point of 0; the ideological topics fix ideal points at -1 and $+1$. We interpret one extreme as liberal and the other as conservative. Data is from the 114th U.S. Senate.

a liberal framing emphasizes "Dreamers" and "DACA", while the conservative frame emphasizes "laws" and "homeland security." We provide more details and empirical studies in Section 5.4.

3.3 Related work

Most ideal point models focus on legislative roll-call votes. These are typically latent-space factor models (Poole & Rosenthal, 1985; McCarty et al., 1997; Poole & Rosenthal, 2000), which relate closely to item-response models (Bock & Aitkin, 1981; Bailey, 2001). Researchers have also developed Bayesian analogues (Jackman, 2001; Clinton et al., 2004) and extensions to time series, particularly for analyzing the Supreme Court (Martin & Quinn, 2002).

Some recent models combine text with votes or party information to estimate ideal points of legislators. Gerrish & Blei (2011) analyze votes and the text of bills to learn political language. Gerrish & Blei (2012) and Lauderdale & Clark (2014) use text and vote data to learn ideal points adjusted for topic. The models in Nguyen et al. (2015) and Kim et al. (2018) analyze votes and floor speeches together. With labeled political party affiliations, machine learning methods can also help map language to party membership. Iyyer et al. (2014) use neural networks to learn partisan phrases, while the models in Tsur et al. (2015) and Gentzkow et al. (2019) use political party labels to analyze differences in speech patterns. Since the TBIP does not use votes or party information, it is applicable to all political texts, even when votes and party labels are not present. Moreover, party

labels can be restrictive because they force hard membership in one of two groups (in American politics). The TBIP can infer how topics change smoothly across the political spectrum, rather than simply learning topics for each political party.

Annotated text data has also been used to predict political positions. Wordscores (Laver et al., 2003; Lowe, 2008) uses texts that are hand-labeled by political position to measure the conveyed positions of unlabeled texts; it has been used to measure the political landscape of Ireland (Benoit & Laver, 2003; Herzog & Benoit, 2015). Ho et al. (2008) analyze hand-labeled editorials to estimate ideal points for newspapers. The ideological topics learned by the TBIP are also related to political frames (Entman, 1993; Chong & Druckman, 2007). Historically, these frames have either been hand-labeled by annotators (Baumgartner et al., 2008; Card et al., 2015) or used annotated data for supervised prediction (Johnson et al., 2017; Baumer et al., 2015). In contrast to these methods, the TBIP is completely unsupervised. It learns ideological topics that do not need to conform to pre-defined frames. Moreover, it does not depend on the subjectivity of coders.

WORDFISH (Slapin & Proksch, 2008) is a model of authored political texts about a single issue, similar to a single-topic version of TBIP. WORDFISH has been applied to party manifestos (Proksch & Slapin, 2009; Lo et al., 2016) and single-issue dialogue (Schwarz et al., 2017). WORDSHOAL (Lauderdale & Herzog, 2016a) extends WORDFISH to multiple issues by analyzing a collection of labeled texts, such as Senate speeches labeled by debate topic. WORDSHOAL fits separate WORDFISH models to the texts about each label, and combines the fitted models in a one-dimensional factor analysis to produce ideal points. In contrast to these models, the TBIP does not require a grouping of the texts into single issues. It naturally accommodates unstructured texts, such as tweets, and learns both ideal points for the authors and ideology-adjusted topics for the (latent) issues under discussion. Furthermore, by relying on stochastic optimization, the TBIP algorithm scales to large data sets. In Section 5.4 we empirically study how the TBIP ideal points compare to both of these models.

3.4 Inference

The TBIP involves several types of latent variables: neutral topics β_k, ideological topics η_k, topic intensities θ_d, and ideal points x_s. Conditional on the text, we perform inference of the latent variables through the posterior distribution $p(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{x} | \boldsymbol{y})$. But calculating this distribution is intractable. We rely on approximate inference.

We use mean-field variational inference to fit an approximate posterior distribution (Jordan et al., 1999; Wainwright et al., 2008; Blei et al., 2017). Variational inference frames the inference problem as an optimization problem. Set $q_\phi(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{x})$ to be a variational family of approximate posterior distributions, indexed by variational parameters ϕ. Variational inference aims to find the setting of ϕ that minimizes the KL divergence between q_ϕ and the posterior.

Minimizing this KL divergence is equivalent to maximizing the *evidence lower bound* (ELBO),

$$\mathbb{E}_{q_\phi}[\log p(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{x}) + \log p(\boldsymbol{y} | \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{x}) - \log q_\phi(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{x})]. \tag{3.4}$$

The ELBO sums the expectation of the log joint—here broken up into the log prior and log likelihood—and the entropy of the variational distribution.

To approximate the TBIP posterior we set the variational family to be the mean-field family. The mean-field family factorizes over the latent variables, where d indexes documents, k indexes topics, and s indexes authors:

$$q_\phi(\boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{\eta}, \boldsymbol{x}) = \prod_{d,k,s} q(\theta_d) q(\beta_k) q(\eta_k) q(x_s). \tag{3.5}$$

We use lognormal factors for the positive variables and Gaussian factors for the real variables,

$$q(\theta_d) = \text{LogNormal}_K(\mu_{\theta_d}, I\sigma^2_{\theta_d}) \tag{3.6}$$

$$q(\beta_k) = \text{LogNormal}_K(\mu_{\beta_k}, I\sigma^2_{\beta_k}) \tag{3.7}$$

$$q(\eta_k) = \mathcal{N}_K(\mu_{\eta_k}, I\sigma^2_{\eta_k}) \tag{3.8}$$

$$q(x_s) = \mathcal{N}(\mu_{x_s}, \sigma^2_{x_s}). \tag{3.9}$$

Our goal is to optimize the ELBO with respect to $\phi = \{\boldsymbol{\mu}_\theta, \boldsymbol{\sigma}^2_\theta, \boldsymbol{\mu}_\beta, \boldsymbol{\sigma}^2_\beta, \boldsymbol{\mu}_\eta, \boldsymbol{\sigma}^2_\eta, \boldsymbol{\mu}_x, \boldsymbol{\sigma}^2_x\}$.

We use stochastic gradient ascent. We form noisy gradients with Monte Carlo and the "reparameterization trick" (Kingma & Welling, 2014; Rezende et al., 2014), as well as with data subsampling (Hoffman et al., 2013). To set the step size, we use Adam (Kingma & Ba, 2015).

We initialize the neutral topics and topic intensities with a pre-trained model. Specifically, we pre-train a Poisson factorization topic model using the algorithm in Gopalan et al. (2015). The TBIP algorithm uses the resulting factorization to initialize the variational parameters for θ_d and β_k. Empirically, we find that this initialization does not have a large effect on the learned ideal points, but it helps make the topics more interpretable. The full procedure is described in Section B.1.

For the corpus of Senate speeches described in Section 3.2, training takes 9 hours on a single NVIDIA Titan V GPU. We have released open source software in Tensorflow and PyTorch.[1]

3.5 Empirical studies

We study the text-based ideal point model (TBIP) on several datasets of political texts. We first use the TBIP to analyze speeches and tweets (separately) from U.S. senators. For both types of texts, the TBIP ideal points, which are estimated from text, are close to the classical ideal points, which are estimated from votes. We also compare the TBIP to existing methods for scaling political texts (Slapin & Proksch, 2008; Lauderdale & Herzog, 2016a). The TBIP performs better, finding ideal points closer to the vote-based ideal points. Finally, we use the TBIP to analyze a group

[1] `http://github.com/keyonvafa/tbip`

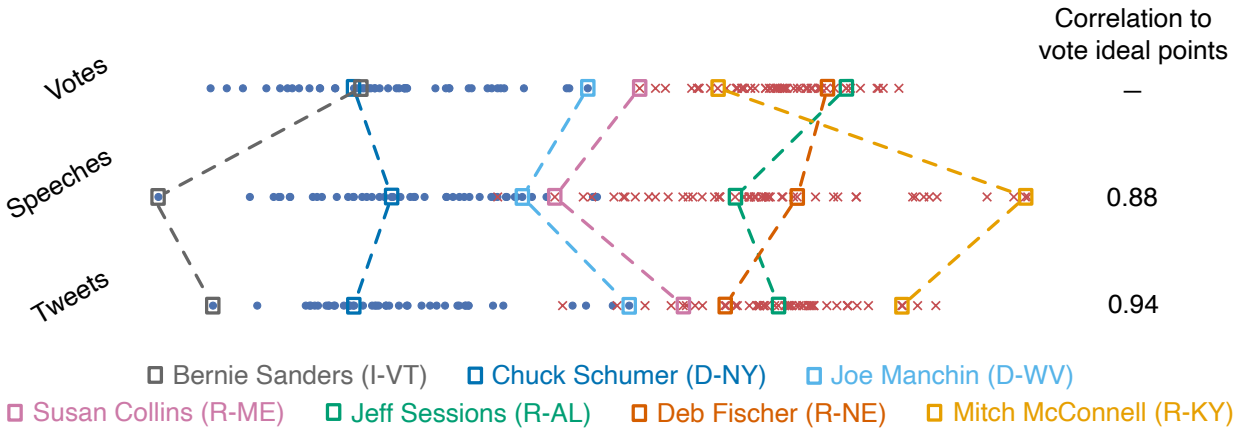


Figure 3.2: The ideal points learned by the TBIP for senator speeches and tweets are highly correlated with the classical vote ideal points. Senators are coded by their political party (Democrats in blue circles, Republicans in red x's). Although the algorithm does not have access to these labels, the TBIP almost completely separates parties.

that does not vote: 2020 Democratic presidential candidates. Using only tweets, it estimates ideal points for the candidates on an interpretable progressive-to-moderate spectrum.

3.5.1 The TBIP on U.S. Senate speeches

We analyze Senate speeches provided by Gentzkow et al. (2018), focusing on the 114th session of Congress (2015-2017). We compare ideal points found by the TBIP to the vote-based ideal point model from Equation 3.1. (Section B.2 provides details about the comparison.) We use approximate posterior means, learned with variational inference, to estimate the latent variables. The estimated ideal points are $\hat{\boldsymbol{x}}$; the estimated neutral topics are $\hat{\boldsymbol{\beta}}$; the estimated ideological topics are $\hat{\boldsymbol{\eta}}$.

Figure 3.2 compares the TBIP ideal points on speeches to the vote-based ideal points.[2] Both models largely separate Democrats and Republicans. In the TBIP estimates, progressive senator Bernie Sanders (I-VT) is on one extreme, and Mitch McConnell (R-KY) is on the other. Susan Collins (R-ME), a Republican senator often described as moderate, is near the middle. The correlation between the TBIP ideal points and vote ideal points is high, 0.88. Using only the text of the speeches, the TBIP captures meaningful information about political preferences, separating the

[2]Throughout our analysis, we appropriately rotate and standardize ideal points so they are visually comparable.

political parties and organizing the lawmakers on a meaningful political spectrum.

We next study the topics. For selected topics, Table 3.1 shows neutral terms and ideological terms. To visualize the neutral topics, we list the top words based on $\hat{\beta}_k$. To visualize the ideological topics, we calculate term intensities for two poles of the political spectrum, $x_s = -1$ and $x_s = +1$. For a fixed k, the ideological topics thus order the words by $\mathbb{E}[\beta_{kv} \exp(-\eta_{kv})]$ and $\mathbb{E}[\beta_{kv} \exp(\eta_{kv})]$.

Based on the separation of political parties in Figure 3.1, we interpret negative ideal points as liberal and positive ideal points as conservative. Table 3.1 shows that when discussing immigration, a senator with a neutral ideal point uses terms like "immigration" and "United States." As the author moves left, she will use terms like "Dreamers" and "DACA." As she moves right, she will emphasize terms like "laws" and "homeland security." The TBIP also captures that those on the left refer to health care legislation as the Affordable Care Act, while those on the right call it Obamacare. Additionally, a liberal senator discussing guns brings attention to gun control: "gun violence" and "background checks" are among the largest intensity terms. Meanwhile, conservative senators are likely to invoke gun rights, emphasizing "constitutional rights."

Comparison to Wordfish and Wordshoal. We next treat the vote-based ideal points as "ground-truth" labels and compare the TBIP ideal points to those found by WORDFISH and WORDSHOAL. WORDSHOAL requires debate labels, so we use the labeled Senate speech data provided by Lauderdale & Herzog (2016b) on the 111th–113th Senates to train each method. Because we are interested in comparing models, we use the same variational inference procedure to train all methods. See Section B.2 for more details.

We use two metrics to compare text-based ideal points to vote-based ideal points: the correlation between ideal points and Spearman's rank correlation between their orderings of the senators. With both metrics, when compared to vote ideal points from Equation 3.1, the TBIP outperforms WORDFISH and WORDSHOAL; see Table 3.2. Comparing to another vote-based method, DW-NOMINATE (Poole, 2005), produces similar results; see Section B.3.

	Speeches 111		Speeches 112		Speeches 113		Tweets 114	
	Corr.	SRC	Corr.	SRC	Corr.	SRC	Corr.	SRC
WORDFISH	0.47	0.45	0.52	0.53	0.69	0.64	0.87	0.80
WORDSHOAL	0.61	0.64	0.60	0.56	0.45	0.44	—	—
TBIP	**0.79**	**0.73**	**0.86**	**0.85**	**0.87**	**0.84**	**0.94**	**0.84**

Table 3.2: The TBIP learns ideal points most similar to the classical vote ideal points for U.S. senator speeches and tweets. It learns closer ideal points than WORDFISH and WORDSHOAL in terms of both correlation (Corr.) and Spearman's rank correlation (SRC). The numbers in the column titles refer to the Senate session of the corpus. WORDSHOAL cannot be applied to tweets because there are no debate labels.

3.5.2 The TBIP on U.S. Senate tweets

We use the TBIP to analyze tweets from U.S. senators during the 114th Senate session, using a corpus provided by VoxGovFEDERAL (2020). Tweet-based ideal points almost completely separate Democrats and Republicans; see Figure 3.2. Again, Bernie Sanders (I-VT) is the most extreme Democrat, and Mitch McConnell (R-KY) is one of the most extreme Republicans. Susan Collins (R-ME) remains near the middle; she is among the most moderate senators in vote-based, speech-based, and tweet-based models. The correlation between vote-based ideal points and tweet-based ideal points is 0.94.

We also use senator tweets to compare the TBIP to WORDFISH (we cannot apply WORDSHOAL because tweets do not have debate labels). Again, the TBIP learns closer ideal points to the classical vote ideal points; see Table 3.2.

3.5.3 Using the TBIP as a descriptive tool

As a descriptive tool, the TBIP provides hints about the different ways senators use speeches or tweets to convey political messages. We use a likelihood ratio to help identify the texts that influenced the TBIP ideal point. Consider the log likelihood of a document using a fixed ideal point $\tilde{x}$ and fitted values for the other latent variables,

$$\ell_d(\tilde{x}) = \sum_v \log p(y_{dv} | \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\eta}}, \tilde{x}).$$

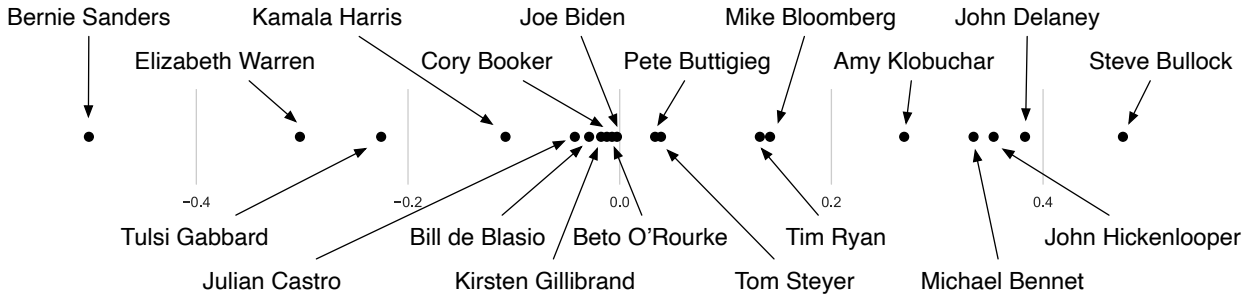


Figure 3.3: Based on tweets, the TBIP places 2020 Democratic presidential candidates along an interpretable progressive-to-moderate spectrum.

Ratios based on this likelihood can help point to why the TBIP places a lawmaker as extreme or moderate. For a document d, if $\ell_d(\hat{x}_{a_d}) - \ell_d(0)$ is high then that document was (statistically) influential in making $\hat{x}_{a_d}$ more extreme. If $\ell_d(\hat{x}_{a_d}) - \ell_d(\max_s(\hat{x}_s))$ or $\ell_d(\hat{x}_{a_d}) - \ell_d(\min_s(\hat{x}_s))$ is high then that document was influential in making $\hat{x}_{a_d}$ less extreme. We emphasize this diagnostic does not convey any causal information, but rather helps understand the relationship between the data and the TBIP inferences.

Bernie Sanders (I-VT). Bernie Sanders is an Independent senator who caucuses with the Democratic party; we refer to him as a Democrat. Among Democrats, his ideal point changes the most between one estimated from speeches and one estimated from votes. Although his vote-based ideal point is the 17th most liberal, the TBIP ideal point based on Senate speeches is the most extreme.

We use the likelihood ratio to understand this difference in his vote-based and speech-based ideal points. His speeches with the highest likelihood ratio are about income inequality and universal health care, which are both progressive issues. The following is an excerpt from one such speech:

> "The United States is the only major country on Earth that does not guarantee health care to all of our people... At a time when the rich are getting richer and the middle class is getting poorer, the Republicans take from the middle class and working families to give more to the rich and large corporations."

Sanders is considered one of the furthest left senators; his extreme speech ideal point is sensible.

That Sanders' vote-based ideal point is not more extreme appears to be a limitation of the vote-based method. Applying the likelihood ratio to votes helps illustrate the issue. (Here a bill takes the place of a document.) The ratio identifies H.R. 2048 as influential. This bill is a rollback of the Patriot Act that Sanders voted against because it did not go far enough to reduce federal surveillance capabilities (RealClearPolitics, 2015). In voting "nay", he was joined by one Democrat and 30 Republicans, almost all of whom voted against the bill because they did not want surveillance capabilities curtailed at all. Vote-based ideal points, which only model binary values, cannot capture this nuance in his opinion. As a result, Sanders' vote-based ideal point is pulled to the right.

Deb Fischer (R-NE). Turning to tweets, Deb Fischer's tweet-based ideal point is more liberal than her vote-based ideal point; her vote ideal point is the 11th most extreme among senators, while her tweet ideal point is the 43rd most extreme. The likelihood ratio identifies the following tweets as responsible for this moderation:

> "I want to empower women to be their own best advocates, secure that they have the tools to negotiate the wages they deserve. #EqualPay"

> "FACT: 1963 Equal Pay Act enables women to sue for wage discrimination. #GetitRight #EqualPayDay"

The TBIP associates terms about equal pay and women's rights with liberals. A senator with the most liberal ideal point would be expected to use the phrase "#EqualPay" 20 times as much as a senator with the most conservative ideal point and "women" 9 times as much, using the topics in Fischer's first tweet above. Fischer's focus on equal pay for women moderates her tweet ideal point.

Jeff Sessions (R-AL). The likelihood ratio can also point to model limitations. Jeff Sessions is a conservative voter, but the TBIP identifies his speeches as moderate. One of the most influential speeches for his moderate text ideal point, as identified by the likelihood ratio, criticizes Deferred

Ideology	Top Words
Progressive	class, billionaire, billionaires, walmart, wall street, corporate, executives, government
Neutral	economy, pay, trump, business, tax, corporations, americans, billion
Moderate	trade war, trump, jobs, farmers, economy, economic, tariffs, businesses, promises, job
Progressive	#medicareforall, insurance companies, profit, health care, earth, medical debt, health care system, profits
Neutral	health care, plan, medicare, americans, care, access, housing, millions
Moderate	healthcare, universal healthcare, public option, plan, universal coverage, universal health care, away, choice
Progressive	green new deal, fossil fuel industry, fossil fuel, planet, pass, #greennewdeal, climate crisis, middle ground
Neutral	climate change, climate, climate crisis, plan, planet, crisis, challenges, world
Moderate	solutions, technology, carbon tax, climate change, challenges, climate, negative, durable

Table 3.3: The TBIP learns topics from 2020 Democratic presidential candidate tweets that vary as a function of the candidate's political positions. The neutral topics are for an ideal point of 0; the ideological topics fix ideal points at -1 and $+1$. We interpret one extreme as progressive and the other as moderate.

Actions for Childhood Arrivals (DACA), an immigration policy established by President Obama that introduced employment opportunities for undocumented individuals who arrived as children:

> "The President of the United States is giving work authorizations to more than 4 million people, and for the most part they are adults. Almost all of them are adults. Even the so-called DACA proportion, many of them are in their thirties. So this is an adult job legalization program."

This is a conservative stance against DACA. So why does the TBIP identify it as moderate? As depicted in Table 3.1, liberals bring up "DACA" when discussing immigration, while conservatives emphasize "laws" and "homeland security." The fitted expected count of "DACA" using the most liberal ideal point for the topics in the above speech is 1.04, in contrast to 0.04 for the most conservative ideal point. Since conservatives do not focus on DACA, Sessions even bringing up the program sways his ideal point toward the center. Although Sessions refers to DACA disapprovingly, the bag-of-words model cannot capture this negative sentiment.

3.5.4 2020 Democratic candidates

We also analyze tweets from Democratic presidential candidates for the 2020 election. Since all of the candidates running for President do not vote on a shared set of issues, their ideal points cannot be estimated using vote-based methods.

Figure 3.3 shows tweet-based ideal points for the 2020 Democratic candidates. Elizabeth Warren and Bernie Sanders, who are often considered progressive, are on one extreme. Steve Bullock and John Delaney, often considered moderate, are on the other. The selected topics in Table 3.3 showcase this spectrum. Candidates with progressive ideal points focus on: billionaires and Wall Street when discussing the economy, Medicare for All when discussing health care, and the Green New Deal when discussing climate change. On the other extreme, candidates with moderate ideal points focus on: trade wars and farmers when discussing the economy, universal plans for health care, and technological solutions to climate change.

3.6 Summary

This chapter presented the text-based ideal point model (TBIP), an ideal point model that analyzes texts to quantify the political positions of their authors. It estimates the latent topics of the texts, the ideal points of their authors, and how each author's political position affects their choice of words within each topic. We used the TBIP to analyze U.S. Senate speeches and tweets. Without analyzing the votes themselves, the TBIP separates lawmakers by party, learns interpretable politicized topics, and infers ideal points close to the classical vote-based ideal points. Moreover, the TBIP can estimate ideal points of anyone who authors political texts, including non-voting actors. When used to study tweets from 2020 Democratic presidential candidates, the TBIP identifies them along a progressive-to-moderate spectrum.

Chapter 4: CAREER: Transfer Learning for Labor Sequence Data

Many important datasets in the social sciences are small: collecting data about humans may be expensive, populations of interest may be narrowly-defined, and ethical considerations may restrict large-scale data collection. While machine learning models have been successfully developed for applied fields like computer vision and natural language processing, their success is largely dependent on the availability of large-scale datasets. Adapting methods from these fields to the social sciences requires making modifications to effectively model smaller datasets.

In this chapter, we present CAREER, a method for modeling career trajectories collected from small-scale surveys. CAREER adapts a transformer, a successful model from natural language processing (NLP), to model career trajectories. CAREER can make effective predictions on these smaller surveys by transfer learning: it augments the survey data of interest with large-scale resume data to aids its predictions. Later, in Chapter 5, we use CAREER to estimate the adjusted gender wage gap.

4.1 Introduction

In labor economics, many analyses rely on models for predicting an individual's future occupations. These models are crucial for estimating important economic quantities, such as gender or racial differences in unemployment (Hall, 1972; Fairlie & Sundstrom, 1999); they underpin causal analyses and decompositions that rely on simulating counterfactual occupations for individuals (Brown et al., 1980; Schubert et al., 2021); and they also inform policy, by forecasting occupations with rising or declining market shares.

In this chapter we develop a novel representation-learning method—a transformer adapted for modeling jobs—for building such predictive models of occupation. Our model is pretrained on

large-scale, passively-collected resume data and fine-tuned to more curated economics datasets, which are carefully collected for unbiased generalization to the larger population. The representation it learns is effective both for predicting job trajectories and for conditioning in downstream economic analyses.

In the past, labor-economics analyses only involved fitting predictive models to small datasets, specifically longitudinal surveys that follow a cohort of individuals during their working career (Panel Study of Income Dynamics, 2021; Bureau of Labor Statistics, 2019a). Such surveys have been carefully collected to represent national demographics, ensuring that the economic analyses can generalize, but they are also very small, usually containing only thousands of workers. As a consequence, prior models of occupation trajectories have been based on very simple sequential assumptions, such as where a worker's next occupation depends only on their most recent occupation (Hall, 1972) or a few summary statistics about their past (Blau & Riphahn, 1999).

In recent years, however, much larger datasets of online resumes have also become available. These datasets contain the occupation histories of millions of individuals, potentially revealing complex information and patterns about career trajectories. But, while one might hope these datasets can improve our economic analyses, there are fundamental difficulties to using them. First, they are passively collected and likely represent a biased sample of the population. Second, they are noisy, since the occupation sequences in the data are derived from text analysis of raw resumes. Finally, they generally omit important economic variables such as demographics and wage, which are essential for the kinds of quantities that economists would like to estimate.

To overcome these challenges, we develop CAREER, a neural sequence model of occupation trajectories. CAREER is designed to be pretrained on large-scale resume data and then fine-tuned to small and better-curated survey data for economic prediction. Its architecture is based on the transformer language model (Vaswani et al., 2017), for which pretraining and fine-tuning has proven to be an effective paradigm for many NLP tasks (Devlin et al., 2019; Lewis et al., 2019). CAREER extends this transformer-based transfer learning approach to modeling sequences of occupations, rather than text. The modifications we make to the transformer's architecture to model

occupations are straightforward yet crucial to the success of the method. We will show that CAREER's representations provide effective predictions of occupations on survey datasets used for economic analysis.

To study this model empirically, we pretrain CAREER on a dataset of 24 million passively-collected resumes. We then fine-tune CAREER's representations of job sequences to make predictions on three widely-used economic datasets: the National Longitudinal Survey of Youth 1979 (NLSY79), another cohort from the same survey (NLSY97), and the Panel Study of Income Dynamics (PSID). In contrast to resume data, these well-curated datasets are representative of the larger population. It is with these survey datasets that economists make inferences, ensuring their analyses generalize.

In this study, we find that CAREER outperforms standard econometric models for predicting and forecasting occupations, achieving state-of-the-art performance on the three widely-used survey datasets. We further find that CAREER can be used to form good predictions of other downstream variables; incorporating CAREER into a wage model provides better predictions than the econometric models currently in use. We release code so that practitioners can train CAREER for their own problems.

In summary, we demonstrate that CAREER can leverage large-scale resume data to make accurate predictions on important datasets from economics. Thus CAREER ties together economic models for understanding career trajectories with transformer-based methods for transfer learning. (See Section 4.3 for details of related work.) A flexible predictive model like CAREER expands the scope of analyses that can be performed by economists and policy-makers.

4.2 CAREER

Given an individual's career history, what is the probability distribution of their occupation in the next timestep? We go over a class of models for predicting occupations before introducing CAREER, one such model based on transformers and transfer learning.

4.2.1 Occupation Models

Consider an individual worker. This person's career can be defined as a series of timesteps. Here, we use a timestep of one year. At each timestep, this individual works in a job: it could be the same job as the previous timestep, or a different job. (Note we use the terms "occupation" and "job" synonymously.) We consider "unemployed" and "out-of-labor-force" to be special types of jobs.

Define an **occupation model** to be a probability distribution over sequences of jobs. An occupation model predicts a worker's job at each timestep as a function of all previous jobs and other observed characteristics of the worker.

More formally, define an individual's career to be a sequence $(y_1, \ldots, y_T)$, where each $y_t \in \{1, \ldots, J\}$ indexes one of J occupations at time t. Occupations are categorical; one example of a sequence could be ("cashier", "salesperson", ... , "sales manager"). At each timestep, an individual is also associated with C observed covariates $x_t = \{x_{tc}\}_{c=1}^{C}$. Covariates are also categorical, with $x_{tc} \in \{1, \ldots, N_c\}$. For example, if c corresponds to the most recent educational degree, x_{tc} could be "high school diploma" or "bachelors", and N_c is the number of types of educational degrees.[1] Define $\boldsymbol{y}_t = (y_1, \ldots, y_t)$ to index all jobs that have occurred up to time t, with the analogous definition for $\boldsymbol{x}_t$.

At each timestep, an occupation model predicts an individual's job in the next timestep, forming a probability distribution $p(y_t | \boldsymbol{y}_{t-1}, \boldsymbol{x}_t)$. This distribution conditions on covariates from the same timestep because these are "pre-transition." For example, an individual's most recent educational degree is available to the model as it predicts their next job.

Note that an occupation model is a predictive rather than structural model. The model does not incorporate unobserved characteristics, like skill, when making predictions. Instead, it implicitly marginalizes over these unobserved variables, incorporating them into its predictive distribution.

[1] Some covariates may not evolve over time. We encode them as time-varying without loss of generality.

4.2.2 Representation-Based Two-Stage Models

An occupation model's predictions are governed by an individual's career history; both whether an individual changes jobs and the specific job they may transition to depend on current and previous jobs and covariates.

We consider a class of occupation models that make predictions by conditioning on a low-dimensional representation of work history, $h_t(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t) \in \mathbb{R}^D$. This representation is assumed to be a sufficient statistic of the past; $h_t(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t)$ should contain the relevant observed information for predicting the next job.

Since individuals frequently stay in the same job between timesteps, we propose a class of models that make predictions in two stages. These models first predict whether an individual changes jobs, after which they predict the specific job to which an individual transitions. The representation is used in both stages.

In the first stage, the career representation $h_t(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t)$ is used to predict whether an individual changes jobs. Define the binary variable s_t to be 1 if a worker's job at time t is different from that at time $t-1$, and 0 otherwise. The first stage is modeled by

$$s_t | \boldsymbol{y}_{t-1}, \boldsymbol{x}_t \sim \text{Bernoulli}\left(\sigma(\eta \cdot h_t(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t))\right), \tag{4.1}$$

where $\sigma(\cdot)$ is the logistic function and $\eta \in \mathbb{R}^D$ is a vector of coefficients.

If the model predicts that an individual will transition jobs, it only considers jobs that are different from the individual's most recent job. To formulate this prediction, it combines the career representation with a vector of occupation-specific coefficients $\beta_j \in \mathbb{R}^D$:

$$p(y_t = j | \boldsymbol{y}_{t-1}, \boldsymbol{x}_t, s_t = 1) = \frac{\exp\{\beta_j \cdot h_t(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t)\}}{\sum_{j' \neq y_{t-1}} \exp\{\beta_{j'} \cdot h_t(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t)\}}. \tag{4.2}$$

Otherwise, the next job is deterministic:

$$p(y_t = j \,|\, \boldsymbol{y}_{t-1}, \boldsymbol{x}_t, s_t = 0) = \delta_{j=y_{t-1}}. \tag{4.3}$$

Two-stage prediction improves the accuracy of occupation models. Moreover, many analyses of occupational mobility focus on whether workers transition jobs rather than the specific job they transition to (Kambourov & Manovskii, 2008). By separating the mechanism by which a worker either keeps or changes jobs (η) and the specific job they may transition to (β_j), two-stage models are more interpretable for studying occupational change.

Equations 4.1 to 4.3 define a two-stage representation-based occupation model. In the next section, we introduce CAREER, one such model based on transformers.

4.2.3 CAREER Model

We develop a two-stage representation-based occupation model called **CAREER**.[2] This model uses a transformer to parameterize a representation of an individual's history. This representation is pretrained on a large resumes dataset and fine-tuned to make predictions on small survey datasets.

Transformers. A transformer is a sequence model that uses neural networks to learn representations of discrete tokens (Vaswani et al., 2017). Transformers were originally developed for natural language processing (NLP), to predict words in a sentence. Transformers are able to model complex dependencies between words, and they are a critical component of modern NLP systems including language modeling (Radford et al., 2019) and machine translation (Ott et al., 2018).

CAREER is an occupation model that uses a transformer to parameterize a low-dimensional representation of careers. While transformers were developed to model sequences of words, CAREER uses a transformer to model sequences of jobs. The transformer enables the model to represent complex career trajectories.

CAREER is similar to the transformers used in NLP, but with two modifications. First, as

[2]CAREER is short for "Contextual Attention-based Representations of Employment Encoded from Resumes."

described in Section 4.2.2, the model makes predictions in two stages, making it better-suited to model workers who stay in the same job through consecutive timesteps. (In contrast, words seldom repeat.) Second, while language models only condition on previous words, each career is also associated with covariates $\boldsymbol{x}$ that may affect transition distributions (see Equation 4.2). We adapt the transformer to these two changes. These modifications are straightforward, easy to implement, and substantially improve the model's predictions.

Parameterization. CAREER's computation graph is depicted in Figure 4.1. Note that in this section we provide a simplified description of the ideas underlying the transformer. Section C.5 contains a full description of the model.

CAREER iteratively builds a representation of career history, $h_t(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t) \in \mathbb{R}^D$, using a stack of L layers. Each layer applies a series of computations to the previous layer's output to produce its own layer-specific representation. The first layer's representation, $h_t^{(1)}(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t)$, considers only the most recent job and covariates. At each subsequent layer ℓ, the transformer forms a representation $h_t^{(\ell)}(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t)$ by combining the representation of the most recent job with those of preceding jobs. Representations become increasingly complex at each layer, and the final layer's representation, $h_t^{(L)}(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t)$, is used to make predictions following Equations 4.1 to 4.3. We drop the explicit dependence on $\boldsymbol{y}_{t-1}$ and $\boldsymbol{x}_t$ going forward, and instead denote each layer's representation as $h_t^{(\ell)}$.

The first layer's representation combines the previous job, the most recent covariates, and the position of the job in the career. It first embeds each of these variables in D-dimensional space. Define an embedding function for occupations, $e_y : [J] \to \mathbb{R}^D$. Additionally, define a separate embedding function for each covariate, $\{e_c\}_{c=1}^C$, with each $e_c : [N_c] \to \mathbb{R}^D$. Finally, define $e_t : [T] \to \mathbb{R}^D$ to embed the position of the sequence, where T denotes the number of possible sequence lengths. The first-layer representation $h_t^{(1)}$ sums these embeddings:

$$h_t^{(1)} = e_y(y_{t-1}) + \textstyle\sum_c e_c(x_{tc}) + e_t(t). \tag{4.4}$$

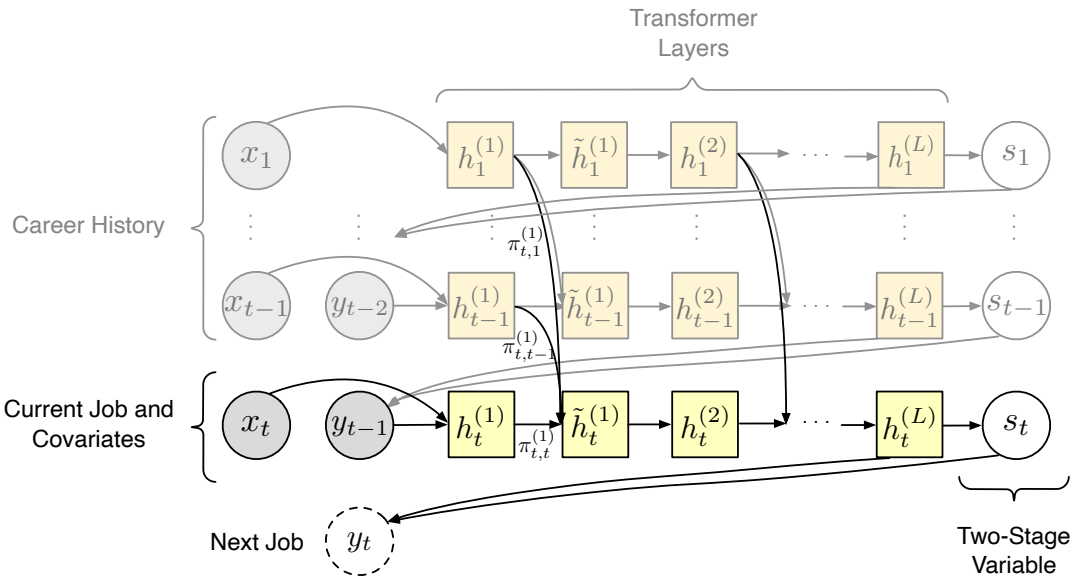


Figure 4.1: CAREER's computation graph. CAREER parameterizes a low-dimensional representation of an individual's career history with a transformer, which it uses to predict the next job.

For each subsequent layer ℓ, the transformer combines representations of the most recent job with those of the preceding jobs and passes them through a neural network:

$$\pi_{t,t'}^{(\ell)} \propto \exp\left\{\left(h_t^{(\ell)}\right)^\top W^{(\ell)} h_{t'}^{(\ell)}\right\} \quad \text{for all } t' \leq t \tag{4.5}$$

$$\tilde{h}_t^{(\ell)} = h_t^{(\ell)} + \textstyle\sum_{t'=1}^{t} \pi_{t,t'}^{(\ell)} * h_{t'}^{(\ell)} \tag{4.6}$$

$$h_t^{(\ell+1)} = \text{FFN}^{(\ell)}\left(\tilde{h}_t^{(\ell)}\right), \tag{4.7}$$

where $W^{(\ell)} \in \mathbb{R}^{D\times D}$ is a model parameter and $\text{FFN}^{(\ell)}$ is a two-layer feedforward neural network specific to layer ℓ, with $\text{FFN}^{(\ell)} : \mathbb{R}^D \to \mathbb{R}^D$.

The weights $\{\pi_{t,t'}^{(\ell)}\}$ are referred to as *attention weights*, and they are determined by the career representations and $W^{(\ell)}$. The attention weights are non-negative and normalized to sum to 1. The matrix $W^{(\ell)}$ can be interpreted as a similarity matrix; if $W^{(\ell)}$ is the identity matrix, occupations t and t' that have similar representations will have large attention weights, and thus t' would contribute more to the weighted average in Equation 4.6. Conversely, if $W^{(\ell)}$ is the negative

identity matrix, occupations that have differing representations will have large attention weights.[3] The final computation of each layer involves passing the intermediate representation $\tilde{h}_t^{(\ell)}$ through a neural network, which ensures that representations capture complex nonlinear interactions.

The computations in Equations 4.5 to 4.7 are repeated for each of the L layers. The last layer's representation is used to predict the next job:

$$p(y_t|\boldsymbol{y}_{t-1}, \boldsymbol{x}_t) = \text{two-stage-softmax}\left(h_t^{(L)}; \eta, \beta\right), \tag{4.8}$$

where "two-stage-softmax" refers to the operation in Equations 4.1 to 4.3, parameterized by the coefficients η and β.

All of CAREER's parameters – including the embedding functions, similarity matrices, feed-forward neural networks, and regression coefficients η and β – are estimated by maximizing the likelihood in Equation 4.8 with stochastic gradient descent (SGD), marginalizing out the latent variable s_t.

Transfer learning. Economists apply occupation models to survey datasets that have been carefully collected to represent national demographics. In the United States, these datasets contain a small number of individuals. While transformers have been successfully applied to large NLP datasets, they are prone to overfitting on small datasets (Kaplan et al., 2020; Dosovitskiy et al., 2021; Variš & Bojar, 2021). As such, CAREER may not learn useful representations solely from small survey datasets.

In recent years, however, much larger datasets of online resumes have also become available. Although these passively-collected datasets provide job sequences of many more individuals, they are not used for economic estimation for a few reasons. The occupation sequences from resumes are imputed from short textual descriptions, a process that inevitably introduces more noise and errors than collecting data from detailed questionnaires. Additionally, individuals may not accurately list their work experiences on resumes (Wexler, 2006), and important economic variables

[3] In practice, transformers use multiple attention weights to perform *multi-headed attention* (Section C.5).

relating to demographics and wage are not available. Finally, these datasets are not constructed to ensure that they are representative of the general population.

Between these two types of data is a tension. On the one hand, resume data is large-scale and contains valuable information about employment patterns. On the other hand, survey datasets are carefully collected, designed to help make economic inferences that are robust and generalizable.

Thus CAREER incorporates the patterns embedded in large-scale resume data into the analysis of survey datasets. It does this through transfer learning: CAREER is first *pretrained* on a large dataset of resumes to learn an initial representation of careers. When CAREER is then fit to a small survey dataset, parameters are not initialized randomly; instead, they are initialized with the representations learned from resumes. After initialization, all parameters are *fine-tuned* on the small dataset by optimizing the likelihood. Because the objective function is non-convex, learned representations depend on their initial values. Initializing with the pretrained representations ensures that the model does not need to re-learn representations on the small dataset. Instead, it only adjusts representations to account for dataset differences.

This transfer learning approach takes inspiration from similar methods in NLP, such as BERT and the GPT family of models (Devlin et al., 2019; Radford et al., 2018). These methods pretrain transformers on large corpora, such as unpublished books or Wikipedia, and fine-tune them to make predictions on small datasets such as movie reviews. Our approach is analogous. Although the resumes dataset may not be representative or carefully curated, it contains many more job sequences than most survey datasets. This volume enables CAREER to learn representations that transfer to survey datasets.

4.3 Related Work

Many economic analyses use log-linear models to predict jobs in survey datasets (Boskin, 1974; Schmidt & Strauss, 1975). These models typically use small state spaces consisting of only a few occupation categories. For example, some studies categorize occupations into broad skill groups (Keane & Wolpin, 1997; Cortes, 2016); unemployment analyses only consider employment sta-

tus (employed, unemployed, and out-of-labor-force) (Hall, 1972; Lauerova & Terrell, 2007); and researchers studying occupational mobility only consider occupational change, a binary variable indicating whether an individual changes jobs (Kambourov & Manovskii, 2008; Guvenen et al., 2020). Although transitions between occupations may depend richly on history, many of these models condition on only the most recent job and a few manually constructed summary statistics about history to make predictions (Hall, 1972; Blau & Riphahn, 1999). In contrast to these methods, CAREER is nonlinear and conditions on every job in an individual's history. The model learns complex representations of careers without relying on manually constructed features. Moreover, CAREER can effectively predict from among hundreds of occupations.

Recently, the proliferation of business networking platforms has resulted in the availability of large resume datasets. Schubert et al. (2021) use a large resume dataset to construct a first-order Markov model of job transitions; CAREER, which conditions on all jobs in a history, makes more accurate predictions than a Markov model. Models developed in the data mining community rely on resume-specific features such as stock prices (Xu et al., 2018), worker skill (Ghosh et al., 2020), network information (Meng et al., 2019; Zhang et al., 2021), and textual descriptions (He et al., 2021), and are not applicable to survey datasets, as is our goal in this chapter (other models reduce to a first-order Markov model without these features (Dave et al., 2018; Zhang et al., 2020)). The most suitable model for survey datasets from this line of work is NEMO, an LSTM-based model that is trained on large resume datasets (Li et al., 2017). Our experiments demonstrate that CAREER outperforms NEMO and other models developed for resumes when they are adapted to model survey datasets.

Recent works in econometrics have applied machine learning methods to sequences of jobs and other discrete data. Ruiz et al. (2020) develop a matrix factorization method called SHOPPER to model supermarket basket data. We consider a baseline "bag-of-jobs" model similar to SHOPPER. Like the transformer-based model, the bag-of-jobs model conditions on every job in an individual's history, but it uses relatively simple representations of careers. Our empirical studies demonstrate that CAREER learns complex representations that are better at modeling job sequences. Rajkumar

et al. (2021) build on SHOPPER and propose a Bayesian factorization method for predicting job transitions. However, their method is focused on modeling individual transitions, so it only conditions on the most recent job in an individual's history. In our empirical studies, we show that models like CAREER that condition on every job in an individual's history form more accurate predictions than Markov models.

CAREER is based on a transformer, a successful model for representing sequences of words in natural language processing (NLP). In econometrics, transformers have been applied to the text of job descriptions to predict their salaries (Bana, 2021) or authenticity (Naudé et al., 2022); rather than modeling text, we use transformers to model sequences of occupations. Transformers have also been applied successfully to sequences other than text: images (Dosovitskiy et al., 2021), music (Huang et al., 2019), and molecular chemistry (Schwaller et al., 2019). Inspired by their success in modeling a variety of complex discrete sequential distributions, CAREER adapts transformers to modeling sequences of jobs. Transformers are especially adept at learning transferrable representations of text from large corpora (Radford et al., 2018; Devlin et al., 2019). We show that CAREER learns representations of job sequences that can be transferred from noisy resume datasets to smaller, well-curated administrative datasets.

4.4 Empirical Studies

We assess CAREER's ability to predict jobs and provide useful representations of careers. We pretrain CAREER on a large dataset of resumes, and transfer these representations to small, commonly used survey datasets. With the transferred representations, the model is better than econometric baselines at both held-out prediction and forecasting.

Resume pretraining. We pretrain CAREER on a large dataset of resumes collected by Zippia Inc., a career planning company.[4] This dataset contains resumes from 23.7 million working Americans. Each job is encoded into one of 330 occupational codes, using the coding scheme of Autor &

[4]We thank Zippia for generously sharing the dataset.

Dorn (2013). We transform resumes into sequences of jobs by including an occupation's code for each year in the resume. For years with multiple jobs, we take the job the individual spent the most time in. We include three covariates: the year each job in an individual's career took place, along with the individual's state of residence and most recent educational degree. We denote missing covariates with a special token. See Section C.6 for an exploratory data analysis of this data.

CAREER uses a 12-layer transformer with 5.6 million parameters. Pretraining CAREER on the resumes data takes 18 hours on a single GPU. Although our focus is on fine-tuning CAREER to model survey datasets rather than resumes, CAREER also outperforms standard econometric baselines for modeling resumes; see Section C.2 for more details.

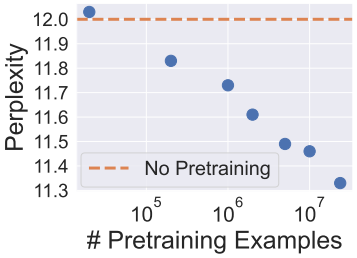
Survey datasets. We transfer CAREER to three widely-used survey datasets: two cohorts from the National Longitudinal Survey of Youth (NLSY79 and NLSY97) and the Panel Study of Income Dynamics (PSID). These datasets have been carefully constructed to be representative of the general population, and they are widely used by economists for estimating important quantities. NLSY79 is a longitudinal panel survey following a cohort of Americans who were between 14 and 22 when the survey began in 1979, while NLSY97 follows a different cohort of individuals who were between 12 and 17 when the survey began in 1997. PSID is a longitudinal survey following a sample of American families, with individuals added over the years.

Compared to the resumes dataset, these survey datasets are small: we use slices of NLSY79, NLSY97, and PSID that contain 12 thousand, 9 thousand, and 12 thousand individuals, respectively. The distribution of job sequences in resumes differs in meaningful ways from those in the survey datasets; for example, manual laborers are under-represented and college graduates are over-represented in resume data (see Section C.6 for more details). We pretrain CAREER on the large resumes dataset and fine-tune on the smaller survey datasets. The fine-tuning process is efficient; although CAREER has 5.6 million parameters, fine-tuning on one GPU takes 13 minutes on NLSY79, 7 minutes on NLSY97, and 23 minutes on PSID.

We compare CAREER to several baseline models: a second-order linear regression with co-

	PSID	NLSY79	NLSY97
Markov regression (Hall, 1972)	18.97 ±0.10	15.03 ±0.03	20.81 ±0.02
NEMO (Li et al., 2017)	17.58 ±0.04	12.82 ±0.04	18.38 ±0.08
Job rep. learning (Dave et al., 2018)	17.23 ±0.16	14.71 ±0.02	16.83 ±0.03
Job2Vec (Zhang et al., 2020)	16.48 ±0.13	14.46 ±0.01	16.20 ±0.02
Bag-of-jobs (Ruiz et al., 2020)	16.21 ±0.08	13.09 ±0.03	16.20 ±0.01
CAREER (vanilla)	15.26 ±0.08	12.20 ±0.04	16.19 ±0.04
CAREER (two-stage)	14.79 ±0.04	12.00 ±0.00	15.22 ±0.03
CAREER (two-stage + pretrain)	**13.88 ±0.01**	**11.32 ±0.00**	**14.15 ±0.03**

(a) Test perplexity on survey datasets (lower is better). Results are averaged over three random seeds. CAREER (vanilla) includes covariates but not two-stage prediction or pretraining; CAREER (two-stage) adds two-stage prediction.



(b) CAREER's scaling law on NLSY79 as a function of pretraining data volume.

Figure 4.2: Prediction results on longitudinal survey datasets and scaling law.

variates and hand-constructed summary statistics about past employment (a common econometric model used to analyze these survey datasets – see Section 4.3); a bag-of-jobs model inspired by SHOPPER (Ruiz et al., 2020) that conditions on all jobs and covariates in a history but combines representations linearly; and several baselines developed in the data-mining community for modeling worker profiles: NEMO (Li et al., 2017), job representation learning (Dave et al., 2018), and Job2Vec (Zhang et al., 2020). As described in Section 4.3, the baselines developed in the data-mining community for modeling worker profiles cannot be applied directly to economic survey datasets and thus require modifications, described in detail in Section C.9. We also compare to two additional versions of CAREER — one without pretraining or two-stage prediction, the other only without two-stage prediction — to assess the sources of CAREER's improvements. All models use the covariates we included for resume pretraining, in addition to demographic covariates (which are recorded for the survey datasets but are unavailable for resumes).

We divide all survey datasets into 70/10/20 train/validation/test splits, and train all models by optimizing the log-likelihood with Adam (Kingma & Ba, 2015). We evaluate the predictive performance of each model by computing held-out perplexity, a common metric in NLP for evaluating probabilistic sequence models. The perplexity of a sequence model p on a sequence $y_1, \ldots, y_T$ is $\exp\{-\frac{1}{T}\sum_{t=1}^{T} \log p(y_t|\boldsymbol{y}_{t-1}, \boldsymbol{x}_t)\}$. It is a monotonic transformation of log-likelihood; better predictive models have lower perplexities. We train all models to convergence and use the checkpoint with the best validation perplexity. See Section C.9 for more experimental details.

Figure 4.2(a) compares the test-set perplexity of each model. With the transferred representations, CAREER makes the best predictions on all survey datasets, achieving state-of-the-art performance. The baselines developed in the data mining literature, which were designed to model large resume datasets while relying on resume-specific features, struggle to make good predictions on these small survey datasets, performing on par with standard econometric baselines. Pretraining is the biggest source of CAREER's improvements. Although the resume data is noisy and differs in many ways from the survey datasets used for economic prediction, CAREER learns useful representations of work experiences that aid its predictive performance. While two-stage prediction is a straightforward modification to the transformer's architecture, requiring only a few additional lines of code, it is crucial to CAREER's success. In Section C.7 we show that modifying the baselines to incorporate two-stage prediction (Equations 4.1 to 4.3) improves their performance, although CAREER still makes the best predictions across datasets.

Figure 4.3 shows an example of a held-out career sequence from PSID. CAREER is much likelier than a regression and bag-of-jobs baseline to predict this individual's next job, biological technician. To understand CAREER's prediction, we show the model's rationale, the jobs in this individual's history that are sufficient for explaining the model's prediction. (We use the greedy rationalization method from Chapter 2; refer to Section C.9 for more details.) In this example, CAREER only needs three previous jobs to predict biological technician: animal caretaker, engineering technician, and student. The model can combine latent attributes of each job to predict the individual's next job. We include more qualitative analysis of CAREER's predictions in Section C.4.

To assess how the volume of resumes used for pretraining affects CAREER's predictions on survey datasets, we downsample the resume dataset and transfer to survey datasets. The scaling law for NLSY79 is depicted in Figure 4.2(b). When there are less than 20,000 examples in the resume dataset, pretraining CAREER does not offer any improvement. The relationship between pretraining volume and fine-tuned perplexity follows a power law, similar to scaling laws in NLP (Kaplan et al., 2020).

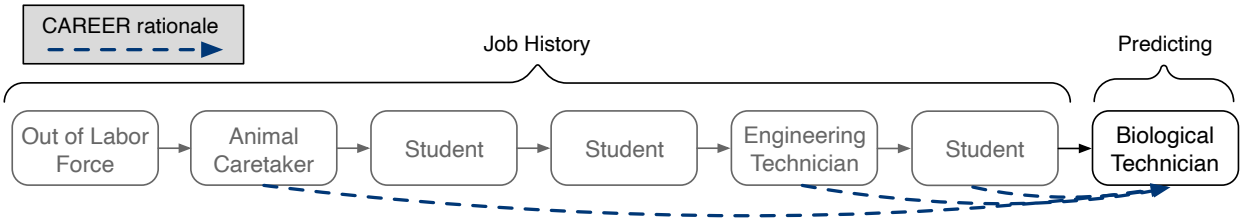


Figure 4.3: An example of a held-out job sequence on PSID along with CAREER's rationale. CAREER ranks the true next job (biological technician) as the most likely possible transition for this individual; in contrast, the regression and bag-of-jobs model rank it as 40th and 37th most likely, respectively. The rationale depicts the jobs in the history that were sufficient for CAREER's prediction.

We also assess CAREER's ability to forecast future career trajectories. In contrast to predicting held-out sequences, forecasting involves training models on all sequences before a specific year. To predict future jobs for an individual, the fitted model is used to estimate job probabilities six years into the future by sampling multi-year trajectories. This setting is useful for assessing a model's ability to make long-term predictions, especially as occupational trends change over time.

We evaluate CAREER's forecasting abilities on NLSY97 and PSID. (These datasets are more valuable for forecasting than NLSY79, which follows a cohort that is near or past retirement age.) We train models on all sequences (holding out 10% as a validation set), without including any observations after 2014. When pretraining CAREER on resumes, we also make sure to only include examples up to 2014. Table 4.1 compares the forecasting performance of all models. CAREER makes the best overall forecasts. CAREER has a significant advantage over baselines at making long-term forecasts, yielding a 17% advantage over the best baseline for 6-year forecasts on NLSY97. Again, the baselines developed for resume data mining, which had been developed to model much larger corpora, struggle to make good predictions on these smaller survey datasets.

4.5 Summary

We introduced CAREER, a method for representing job sequences from large-scale resume data and fine-tuning them on smaller datasets of interest. We took inspiration from modern language modeling to develop a transformer-based occupation model. We transferred the model from

	NLSY97				PSID			
	Overall	2-Year	4-Year	6-Year	Overall	2-Year	4-Year	6-Year
Markov regression	23.11	12.50	25.88	36.59	19.43	11.83	21.66	27.89
Bag-of-jobs	22.51	11.98	25.11	36.29	19.28	11.44	21.68	28.14
NEMO	25.26	12.59	28.35	43.01	18.58	11.08	20.67	27.29
CAREER	**19.41**	**10.78**	**21.57**	**30.19**	**16.51**	**10.35**	**18.30**	**23.18**

Table 4.1: Forecasting perplexity (lower is better) on NLSY97 and PSID. Results are averaged over three random seeds.

a large dataset of resumes to smaller survey datasets in economics, where it achieved state-of-the-art performance for predicting and forecasting career outcomes.

Chapter 5: Adjusting the Gender Wage Gap for Full Job History

Chapter 4 developed a predictive model of career trajectories. In this chapter, we adapt this predictive model to estimate an important economic quantity: the history-adjusted gender wage gap. To do so, we develop a machine learning technique that encourages both high predictive accuracy and valid estimation.

5.1 Introduction

The **gender wage gap** measures the average difference in wages between males and females. In the United States, female workers earn on average 79% of the income of male workers, a fact which has far-reaching implications for economic inequality, family dynamics, and workforce participation (Blau & Kahn, 2017). On average, a female worker would need to work more than one additional day per week to make the same income as a male worker.

A central goal for economists and policy-makers is to understand factors that contribute to the gender wage gap. To this end, researchers calculate the **adjusted gender wage gap**, which is an expected wage gap that is conditioned on a set of observed characteristics X. Denote the log wage as Y and gender as G, a variable that can take on the values "M" or "F". The adjusted gender wage gap is

$$\mathbb{E}_{X \sim p(X|G=F)} \left[\mathbb{E}[Y|G = F, X] - \mathbb{E}[Y|G = M, X] \right].^{1} \tag{5.1}$$

A related quantity is the **adjusted gender wage ratio**, which exponentiates Equation 5.1.

Why are we are interested in the adjusted wage gap? If we find characteristics X for which the adjusted wage gap is zero, then these characteristics "explain" the unadjusted gap in the sense that groups of men and women with similar characteristics will have no difference in wage. Knowing

[1] The outer expectation over covariates can be taken with respect to any distribution over covariates X; we follow Blau & Kahn (2017) in taking it with respect to the covariate distribution of females.

which factors explain the wage gap can help direct further investigations. If there is a covariate that brings the adjusted gender wage gap in Equation 5.1 closer to 0, economists and policy-makers can prioritize studying why gender differences in the covariate exist, how these differences can be reduced, and whether interventions to reduce these differences would result in more equal pay.

Economists estimate adjusted gender wage gaps using longitudinal surveys collected from a representative sample of the population. They use these survey datasets to model an individual's average wage from their gender and covariates, $\hat{E}[Y|G, X] \approx \mathbb{E}[Y|G, X]$, which is then used to calculate the adjusted wage gap as in Equation 5.1. Because constructing these survey datasets requires regularly interviewing a cohort of individuals, they are small. However, the variables economists use to adjust wage gaps for are typically simple, enabling efficient estimation of $\mathbb{E}[Y|G, X]$ from small datasets.

For example, consider adjusting the gender wage gap for industry. To estimate Equation 5.1 with X indicating the industry an individual works in, economists use survey datasets that record gender, wage, and industry for each respondent. For each individual, industry can be coded as a categorical variable taking on 15-20 categories (Blau & Kahn, 2017). The sample is then divided into male and female subsamples. For each gender, $\mathbb{E}[Y|G, X]$ is modeled as a linear regression with a single predictor: an indicator for the industry an individual works in. Because the model and predictors are simple, $\hat{E}[Y|G, X]$ can effectively model $\mathbb{E}[Y|G, X]$ without much data. The adjusted wage gap in Equation 5.1 can then be estimated by plugging in the model estimates:

$$\frac{1}{N_F} \sum_i (\hat{E}[Y|G = F, X_i] - \hat{E}[Y|G = M, X_i]) * 1(G_i = F), \tag{5.2}$$

where $1(G_i = F)$ is an indicator that the i'th sample is a female and N_F is the number of female samples. If males and females who work in the same industry have more similar pay than males and females in the overall population, gender differences in industry explain a portion of the wage gap.

One important variable to adjust wage gaps for is the full job history of an individual, their trajectory of work experience. Intuitively, job history can go a long way towards explaining why

two managers, both 15 years out of college, have different wages. Formally, assume that in addition to covariates X, each individual is associated with a job history, H. Each history is a sequence of discrete jobs, $H = (h_1, \ldots, h_T)$, where each $h_t \in \{1, \ldots, J\}$ indexes one of J occupations an individual worked in at time t.[2] We define the **full-history adjusted wage gap** as

$$\mathbb{E}_{X,H \sim p(X,H|G=F)}[\mathbb{E}[Y|G = F, X, H] - \mathbb{E}[Y|G = M, X, H]]. \tag{5.3}$$

This quantity is analogous to the adjusted wage gap in Equation 5.1, except it adjusts for both covariates and complete history.

Adjusting the gender wage gap for full work history may reveal important factors that contribute to the raw wage gap. Prior work has found that adjusting for years of actual work experience instead of potential experience (an inexact measure of experience that does not measure workforce interruptions) reduces the unexplained gender wage gap (Blau & Kahn, 2013). This finding — that workforce interruptions play an important role in the raw wage gap — could only be realized by using detailed measures of work experience. Other studies have found relatively equal pay between males and females shortly after graduation from law school (Noonan et al., 2005) and MBA programs (Bertrand et al., 2010), yet large pay disparities 10-15 years later. Incorporating detailed work history into wage gap analyses can help identify the histories that are associated with different pay for males and females in these professions. Finally, fine-grained analyses of job history may identify subpopulations for which there exist large pay gaps. These analyses can help form hypotheses for the nature of discrimination to be prioritized for future study.

However, incorporating full history into the wage models that underlie adjusted wage gap estimation is statistically untenable. The reason is that there are many more possible combinations of work history H than there are data points available for modeling $\mathbb{E}[Y|G, X, H]$; job histories are *high-dimensional*. Thus, estimating this conditional expectation by including an indicator for each possible history is impossible. In practice, researchers instead adjust for crude summaries of job

[2]We depart from the notation in Chapter 4 and represent job sequences with H rather than Y, since the label Y is reserved to indicate wage (as is standard in the econometrics literature).

history, such as years of full-time and part-time experience and an individual's current occupation and industry (Blau & Kahn, 2017). However, these summary statistics leave things unexplained; male and female managers with 15 years of experience may have differences in work history that help explain differences in their wage.

In this chapter, we develop methodology for estimating the adjusted gender wage gap for full history (Equation 5.3). Our solution adapts ideas from machine learning to enable efficient estimation of this quantity.

Our methodology is based on learning a function of job history, denoted by $\lambda_\theta(H)$, that summarizes an individual's full history. Rather than modeling wages from the full, high-dimensional job histories, this approach allows us to model wages using the low-dimensional summaries $\lambda_\theta(H)$. These summaries are then used to estimated the adjusted wage gap. We refer to $\lambda_\theta(H)$ as a *representation* of job history. Specifically, assume that $\lambda_\theta : \mathcal{H} \to \mathbb{R}^D$ maps a job history, $H \in \mathcal{H}$, to a D-dimensional vector. (This function belongs to a class of representations, with a specific function λ_θ indexed by parameters θ.) For a specific representation λ_θ, we define the **representation-adjusted gender wage gap** as

$$\mathbb{E}_{X,H\sim p(X,H|G=F)}[\mathbb{E}[Y|G=F,X,\lambda_\theta(H)] - \mathbb{E}[Y|G=M,X,\lambda_\theta(H)]]. \tag{5.4}$$

What makes a good representation λ_θ? The ideal representation is one for which the full-history adjusted wage gap (Equation 5.3) is equal to the representation-adjusted wage gap (Equation 5.4). When these quantities are not equal, λ_θ induces *bias*. We will show that there is no bias if $\lambda_\theta(H)$ contains all the aspects of job history that are predictive of both gender and wage. This insight motivates our method for fitting λ_θ.

Learning a low-dimensional representation of job history from survey data can suffer from the same problem as modeling wages from high-dimensional histories: the survey datasets used to fit these quantities are small. There are many more possible histories than there are surveys. Our solution, similar to Chapter 4, is to leverage a large-scale, passively-collected dataset of job

histories to learn the representation $\lambda_\theta(H)$. This dataset doesn't contain information about wage or gender, so we fit a representation $\lambda_\theta(H)$ by modeling individual job sequences. By learning features that are important for predicting job trajectories, the representation also learns features that are predictive of wage and gender. This representation is then adjusted on a survey dataset to estimate the history-adjusted wage gap.

We assess the effectiveness of this estimation procedure using both real and semi-synthetic data. A prerequisite for accurately estimating the adjusted wage gap is a model that makes good predictions of wage on survey data. We show that our method makes more accurate predictions of wage than econometric baselines that summarize job history with hand-constructed summary statistics, with improvements ranging from 10-15% in terms of held-out mean-squared error. However, predictive performance isn't everything; wage gap estimates may be biased if the learned low-dimensional representations discard aspects of job histories that are relevant to both gender and wage. In a suite of semi-synthetic experiments where the true adjusted wage gaps are known, we show that our method makes the most accurate wage gap estimates, and that our procedure for fitting λ_θ improves estimation accuracy.

We then estimate the full-history adjusted gender wage gap for workers in the United States. Using longitudinal survey data from the Panel Study of Income Dynamics (PSID) (Panel Study of Income Dynamics, 2021), we estimate the adjusted gender wage gap for the seven most recent years of the survey, ranging from 2006-2018. Across years, we find a consistent trend: history explains between 1/3 to 1/2 of the gap previously unexplained by other covariates and summary statistics of history. Still, across years, the remaining 1/2 to 2/3 of the gender wage gap remains unexplained even when accounting for full history. We break down the gap by occupation types along with clusters based on current and previous employment. We find groups where history consistently explains a substantial portion of the wage gap (managers), along with groups where history cannot explain much, if any, of the gap (lawyers and physicians).

5.1.1 Interpreting adjusted wage gaps

The adjusted gender wage gap in Equation 5.1 can be difficult to interpret. To gain insight into this quantity, assume that males have higher wages than females on average, but that males and females who work in the same industry have the same wage. If X indicates an individual's industry, the raw gender wage gap will be negative while the adjusted gap in Equation 5.1 will be zero. In this case, gender differences in industry explain the raw wage gap since, conditional on industry, there is no difference in expected wage between males and females. In practice, the adjusted wage gap conditions on multiple covariates, such as measures of experience, education, and occupation.

The adjusted wage gap in Equation 5.1 is mathematically similar to a natural direct effect (Robins & Greenland, 1992; Pearl, 2001), a causal quantity. It may be tempting to interpret the adjusted wage gap causally, as an estimate of the raw wage gap after intervening to set the covariates X to the same values for males and females. However, interpreting the adjusted wage gap causally requires strong assumptions — such as no unmeasured variables that are correlated with wage, gender, and the observed covariates — that may not be realistic in practice (Fortin et al., 2011; Huber, 2015). Rather, Equation 5.1 is a descriptive quantity, identifying variables for which gender differences are correlated with wage differences.

The adjusted gender wage gap is also not a measure of gender discrimination. If the adjusted wage gap for a set of covariates is zero, it does not imply that there is no wage discrimination; rather, that conditional on the same set of covariates, males and females do not have pay differences. In this case, some covariates, such as industry or occupation, may be the result of historic discrimination (Blau & Kahn, 2017). Conversely, a non-zero value of the adjusted wage gap does not imply discrimination, since there may be unmeasured variables (such as productivity) that affect wage yet vary between genders. Still, observing differences between raw and adjusted wage gaps can help researchers hypothesize about the nature of discrimination.

This chapter builds on a line of work that estimates gender wage gaps by comparing the observable characteristics of male and female workers (Altonji & Blank, 1999; Blau & Kahn, 2017).

More recent approaches have used matched firm-worker data to study the role of firms. For example, Webber (2016) finds gender differences in labor supply elasticities to be across-firms rather than within-firms, while Card et al. (2016) decompose gender differences in firm pay into sorting and bargaining effects. In this chapter, we stick with the relatively simpler setting where the role of firms is not considered. However, low-dimensional representations of work history can also be incorporated into the worker-firm effects models that underpin the studies of firm effects.

5.2 Methodology

In this section, we describe our methodology in detail. We describe how to model low-dimensional representations of job history to estimate the history-adjusted gender wage gap. We then describe how to leverage large-scale, passively-collected resume data to enhance the quality of these representations.

5.2.1 Modeling job histories with representations

Our methodology for estimating the history-adjusted gender wage gap centers on finding a low-dimensional representation of job history. But what makes a good representation? Any function of history constitutes a representation, but some are more useful than others for estimating the full-history adjusted wage gap.

For example, taking $\lambda_\theta(H) = H$, the full-history adjusted wage gap (Equation 5.3) is trivially identical to the representation adjusted wage gap (Equation 5.4). Thus estimating the conditional wage function with the representation $\mathbb{E}[Y|G, X, \lambda_\theta(H)]$ is as difficult as estimating the conditional wage function with full histories $\mathbb{E}[Y|G, X, H]$. A more effective approach is to find a representation that makes it easier to predict wages from history; in other words, modeling $\mathbb{E}[Y|G, X, \lambda_\theta(H)]$ should be easier than modeling $\mathbb{E}[Y|G, X, H]$.

Motivated by learning a representation of job history that makes accurate predictions, we adapt the CAREER model described in Chapter 4 to predict wages rather than future occupations.

CAREER uses a neural network to parameterize representations for each job in a sequence.

Each representation is contextual, meaning that the representation for each job depends on the other jobs in the history. Formally, for a history of jobs $H = \{h_1, \ldots, h_T\}$, CAREER parameterizes a D-dimensional representation of each job in the sequence, with the representations denoted as $\lambda_{\theta,1}(H), \ldots, \lambda_{\theta,T}(H)$, where $\lambda_{\theta,t} : \mathcal{H} \to \mathbb{R}^D$.[3] Although CAREER provides representations for each job in a history, modeling wage only requires making a prediction about the current job. Thus, we define the full-history representation $\lambda_\theta(H) = \lambda_{\theta,T}(H)$ to be the contextual representation of the last job in the history (the current job).

The representation-adjusted gender wage gap is estimated by building models to predict wage as a function of gender, covariates, and CAREER's representation. These models are denoted by $\hat{E}[Y|G = F, X, \lambda_\theta(H)]$ for females and $\hat{E}[Y|G = M, X, \lambda_\theta(H)]$ for males. Given covariates $X \in \mathbb{R}^P$ and a representation of history $\lambda_\theta(H) \in \mathbb{R}^D$, we use the following functional forms:

$$\hat{E}[Y|G = F, X, \lambda_\theta(H)] = \gamma_F \cdot g_F(\lambda_\theta(H)) + \beta_F \cdot X \tag{5.5}$$

$$\hat{E}[Y|G = M, X, \lambda_\theta(H)] = \gamma_M \cdot g_M(\lambda_\theta(H)) + \beta_M \cdot X, \tag{5.6}$$

where $g_F, g_M : \mathbb{R}^D \to \mathbb{R}^D$ are two-layer feedforward neural networks while $\gamma_F, \gamma_M \in \mathbb{R}^D$ and $\beta_F, \beta_M \in \mathbb{R}^P$ are regression coefficients. This model passes the representation of history through separate nonlinear functions for males and females before combining it with the other covariates to predict wage. When the terms corresponding to history are removed, this conditional expectation function is identical to the regression models used by Blau & Kahn (2017) to estimate adjusted wage gaps.

These models replace high-dimensional job histories H with low-dimensional representations $\lambda_\theta(H)$ with the goal of improving predictions of wage. However, the representation-adjusted wage gap (Equation 5.4) may not equal the quantity we are using it to estimate, the full-history adjusted wage gap (Equation 5.3). We refer to the difference between these two quantities as the *bias* of a representation λ_θ. Here we ask, for what representations λ_θ is the representation-adjusted wage

[3]For more details of the functional form of these representations, see Chapter 4.

gap in Equation 5.4 equivalent to the history-adjusted gap in Equation 5.3?

Proposition 1. *Suppose that* $(G, H, X, Y) \sim P$ *and* $\lambda_\theta : \mathcal{H} \to \mathbb{R}^D$. *Assume the following hold:*

1. *Overlap:* $0 < P(G = F|H, X) < 1$ *for all* X *and* H.

2. *Sufficiency:* $H \perp\!\!\!\perp G|\lambda_\theta(H), X$.

Then the representation-adjusted wage gap in Equation 5.4 is equivalent to the history-adjusted gap in Equation 5.3.

The first condition (overlap) is a property of the data, while the second condition (sufficiency) relates to both the data and the learned representation of history $\lambda_\theta(H)$. Intuitively, overlap is necessary for the wage gaps in Equations 5.3 and 5.4 to be well-defined; overlap in H implies overlap in $\lambda_\theta(H)$. The second condition enforces that $\lambda_\theta(H)$ carries all information from job histories that are predictive of gender. When this holds, $\mathbb{E}[Y|G, \pi(H, X)]$ is $(\lambda_\theta(H), X)$-measurable for $\pi(H, X) = P(G = F|H, X)$, and the proof follows from Theorem 3.1 of Veitch et al. (2020). Although the adjusted wage gap is not a causal quantity, the insight that $\lambda_\theta(H)$ only needs to capture parts of the history that are relevant to both wage and gender is due to the sufficiency of the propensity score for treatment effect estimation (Rosenbaum & Rubin, 1983).

Typically, parameters underlying wage models are inferred by minimizing the predictive error of wage (Blau & Kahn, 2017). However, Proposition 1 shows that predictive accuracy isn't the only important quality for a representation λ_θ. It is possible to learn representations that induce relatively low predictive errors yet large estimation biases.

Between these two goals — predictive accuracy and sufficiency — lies a tension. Representations of history that lead to accurate predictions of wage, i.e. those for which $\hat{E}[Y|G, X, \lambda_\theta(H)] \approx \mathbb{E}[Y|G, X, H]$, may result in estimates of Equation 5.3 and Equation 5.4 that are far apart when $\lambda_\theta(H)$ is not sufficient. On the other hand, if $\lambda_\theta(H)$ is sufficient for gender but not predictive of wage, the expected wage function can lead to poor predictions for small datasets; for example, keeping $\lambda_\theta(H) = H$ satisfies sufficiency, but will lead to a poor model of expected wage when histories are high-dimensional.

Instead, we cast the problem of finding a representation of history that is both predictive of wage and sufficient for gender as a constrained optimization:

$$\hat{\theta} = \arg\min_{\theta} \tfrac{1}{N} \textstyle\sum_i (Y_i - \hat{E}[Y|G, X, \lambda_\theta(H)])^2 \tag{5.7}$$

$$\text{s.t. } G \perp\!\!\!\perp H|\lambda_\theta(H), X \tag{5.8}$$

The objective in Equation 5.7 encourages low-dimensional representations that are predictive of history, while the constraint in Equation 5.8 enforces that the representations are sufficient and lead to valid adjusted estimates. It is unrealistic to assume that low-dimensional representations will capture all aspects of job history that are predictive of wage, so the sufficiency constraint enforces the model to miss in ways that lead to valid adjusted estimates.

For most representations λ_θ, this objective is intractable. We propose an iterative procedure for approximating a solution to this objective. At the beginning of the procedure, θ is randomly initialized. At a high level, each iteration of the procedure has two steps. In the first step, θ is updated with gradient descent to minimize the predictive error of wage in Equation 5.7. In the second step, θ is then projected to the space of sufficient representations, i.e. those satisfying Equation 5.8. Our procedure involves repeating these two steps; when the expected wage function is non-convex in θ, the solution of the minimization step depends on the output of the projection step. The full algorithm is presented in Algorithm 3.

Note that it is challenging to characterize the space of sufficient representations. For $\lambda_\theta(H)$ to be sufficient, it must be (with X) a balancing score for gender (Rosenbaum & Rubin, 1983). In other words, $\lambda_\theta(H)$ must carry all the information from history that is predictive of gender. For a given class of low-dimensional representations, we define the space of sufficient representations $\mathcal{S}$ to be

$$\mathcal{S} = \left\{\theta \text{ s.t. } \mathbb{E}[\log P(G|X, \lambda_\theta(H))] = \max_{\theta} \mathbb{E}[\log P(G|X, \lambda_\theta(H))]\right\}. \tag{5.9}$$

That is, $\mathcal{S}$ contains all the representations in the model class for which the expected conditional log-likelihood of gender is maximized. Note that this characterization of the space of sufficient

representations may be approximate; members of $\mathcal{S}$ are only sufficient if there exists a representation θ^* such that $P(G|X, \lambda_{\theta^*}(H)) = P(G|X, H)$. Also note that the size of $\mathcal{S}$ depends on the data distribution and the functional form of the representation, e.g. when $G \perp\!\!\!\perp H|X$, every representation in the model class is sufficient.

The best way to define the projection onto the space of sufficient representations depends on the functional form of λ_θ. We use a transformer neural network to parameterize λ_θ; since transformers are effective at retaining information from transfer learning, we use a fine-tuning approach to project onto the space of sufficient representations. For a given value of $\hat{\theta}$, we project onto the space of sufficient representations $\mathcal{S}$ by performing gradient descent on samples of the data to maximize the gender prediction score $\mathbb{E}[\log p(G|\lambda_\theta(H), X)]$, *having initialized the representation at* $\lambda_{\hat{\theta}}$. Because this objective function is non-convex, the projected representation depends on the initial value of θ; initializing at $\hat{\theta}$ encourages a projected representation λ_θ that is close to $\lambda_{\hat{\theta}}$. This approach is reminiscent of fine-tuning approaches for NLP, where a transformer is fine-tuned to a specific task by initializing gradient descent with parameters that have been pretrained on large corpora (Devlin et al., 2019; Radford et al., 2019). By initializing gradient descent at a representation that is predictive of wage, we encourage the projected representation to be not only sufficient for gender but also predictive of wage. We refer to the final representation as a *sufficient representation*.

There is a risk that projecting to the space of sufficient representations degrades the quality of wage predictions. Thus, after the final round of projections, we perform an additional round of wage error minimization. Throughout our semi-synthetic experiments, we find that including this additional round of wage error minimization results in more accurate wage predictions and more accurate estimates of the wage gap. Although this last step of the procedure does not then involve projecting the representation onto the space of sufficient representations, we find that wage gap estimates are still significantly more accurate than those from estimating the wage gap without projections. This is because this final round of wage error minimization is fine-tuned from a sufficient representation; the final representation is encouraged to be not only predictive of wage but also sufficient for gender.

Although the adjusted wage gap is not a causal quantity, the biased induced by insufficiency is mathematically equivalent to the *omitted variable bias* that can arise when estimating causal effects in the presence of high-dimensional confounding (Chernozhukov et al., 2022a). For example, if an observed confounder in a causal inference problem is weakly related to an outcome but strongly related to treatment, sparsity-inducing outcome models may omit this variable, resulting in biased estimates (Hahn et al., 2018). The problem in our setting is analogous: the representation-adjusted wage gap is biased when an aspect of job history that is related to both gender and wage is excluded from the learned low-dimensional representation.

Many remedies have been proposed to limit this bias in causal estimation problems (Belloni et al., 2014; Johansson et al., 2016; Athey et al., 2018; Assaad et al., 2021; Chernozhukov et al., 2022a), although the best approach is problem-specific (Alaa & Van Der Schaar, 2019). The constrained optimization procedure we describe is similar to the causal estimation methods proposed in Shi et al. (2019) and Chernozhukov et al. (2022b), which optimize a representation to jointly predict treatment and outcome. Unlike our procedure, these methods require a hyperparameter that tunes the tradeoff between the predictive accuracy of treatment and outcome. These methods are thus susceptible to the pitfalls that come with hyperparameter selection, and we find these methods to be sensitive to this hyperparameter in practice.

5.2.2 Leveraging large-scale data to learn representations

We have described a method that can be fit to survey datasets to estimate the history-adjusted wage gap. The survey datasets that economists typically use for wage gap estimation are small, containing only thousands of individuals. However, transformers are unlikely to learn useful representations of job histories from these small datasets (Kaplan et al., 2020). Here, we present an approach that uses transfer learning to learn effective representations.

Specifically, we leverage large-scale, passively-collected resume data to learn representations of job history. Our approach is inspired by the method in Chapter 4, where next-job predictions from resume data form the basis for next-job predictions on survey data. This method is motivated

by the fact that career trajectories in resume and survey data share common features, so using large-scale auxiliary data from resumes can help to model survey data. However, in contrast to Chapter 4, our goal in this chapter is to predict a quantity that does not appear in resume data: an individual's wage. Thus, we cannot use wage predictions from resume data to form the basis for wage predictions on survey data.

Instead, we develop a strategy that leverages resume data to learn important features without directly predicting wage. While resume datasets do not contain wages, they can contain millions of samples of job sequences. These sequences encode information about the relationships between jobs. If there are features that are relevant both to whether jobs occur in a sequence together and wage, they can be gleaned from resume data.

Our approach uses resume data to predict which jobs are in sequences together. Specifically, we begin by initializing the representation function λ_θ randomly. Then, we model:

$$p(h_t = j \,|h_S) = p(h_t = j \,|\lambda_t(h_S)) \propto \exp\left\{\alpha_j^\top \lambda_t(h_S)\right\}, \tag{5.10}$$

where S is a random index set, t indexes a timestep that is not present in the sampled index set of jobs, and $\alpha_j \in \mathbb{R}^D$ is a vector of coefficients. We maximize the likelihood in Equation 5.10 with respect to CAREER's parameters θ and the coefficients α_j by maximum likelihood estimation on sequences from resume data, re-sampling index sets S at each optimization step. Then, to model the history-adjusted wage gap with Algorithm 3, we initialize λ_θ with the fitted values.

5.3 Semi-Synthetic Experiments

Ideally, we could assess the accuracy of wage gap estimates on real world data. However, ground truth wage gaps are not available in the real world. Instead, we compare approaches on semi-synthetic data. Semi-synthetic experiments are a common method to assess causal estimation strategies in the presence of high-dimensional confounders. In order for these experiments to reflect real-world data settings, we use real job histories H.

The main idea of the experiments are as follows: we generate data with a known, ground-truth adjusted wage gap. We then compare how close different estimation methods come to finding it.

These experiments begin by forming a confounder, a function of history that is correlated with both gender and wage. Models that do not account for this confounder will not be able to estimate the true adjusted wage gap. Denoting the confounder as $\lambda_\phi(H)$, wages are then sampled as

$$Y_i = \tau * 1(G_i = F) + \gamma * (\hat{\pi}(\lambda_\phi(H)) - 0.5) + \epsilon_i,$$
$$\epsilon_i \sim \mathcal{N}(0, (0.1)^2),$$

where $\hat{\pi} : \mathbb{R}^K \to [0, 1]$ corresponds to a logistic regression fit from the confounder $\lambda_\phi(H)$ to predict gender, analogous to a propensity score. The parameter $\tau \in \mathbb{R}$ is the true parameter of interest, and $\gamma \in \mathbb{R}$ controls the confounding strength.

We use a transformer to model the confounder λ_ϕ. Specifically, we fit λ_ϕ to real data to predict wage. We use two settings: one where the confounder λ_ϕ is in the model class of the transformer used for estimation, and one where it is not. To generate a confounder λ_ϕ that is outside the model class used for estimation, we use a transformer architecture that is 20 times larger than than the one used for estimation. We overfit this model to the wage data to ensure that it is outside the model class of the smaller transformer. Then, $\hat{\pi}(\lambda_\phi(H))$ is generated by averaging the fitted propensity scores for this model and the one within the model class.

We compare five approaches for estimating the adjusted gender wage ratio (given by $\exp(\tau)$). The unadjusted gender wage ratio computes the difference in male and female average wages, while the non-history adjusted ratio models wage as a function of covariates X that includes summary statistics about history (Blau & Kahn, 2017). CAREER (no projections) uses CAREER to model wage without modeling gender. CAREER (joint optimization) follows Shi et al. (2019) and Chernozhukov et al. (2022b): it uses CAREER to jointly minimize the predictive error of wage and gender, controlled by a hyperparameter. Finally, CAREER (w/ projections) follows the sufficiency-constrained optimization procedure outlined in Algorithm 3.

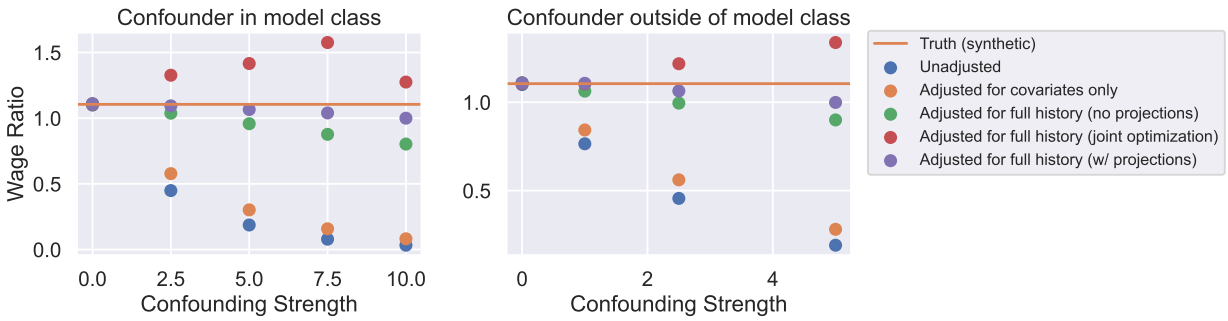


Figure 5.1: Semi-synthetic experiments comparing estimates of the history-adjusted wage gap to the true wage gap when the synthetic confounder is in and outside the model class. Five methods are compared: an unadjusted estimator; a classical estimator of the wage gap that conditions on only covariates and summary statistics about history ("non-history adjusted"); a version that uses CAREER as a wage model but does not try to enforce sufficiency ("no projections"); a version that jointly optimizes to predict wage and gender following Shi et al. (2019) ("joint optimization"); and a version that follows the sufficiency-constrained optimization approach we develop ("w/ projections").

Figure 5.1 depicts the results. Across settings, CAREER (with projections) estimates a wage ratio that is closest to the true underlying wage ratio. Although the unadjusted and non-history adjusted models perform fine when there is little confounding (as expected), they struggle as the confounding strength increases. We also find that the performance of the joint optimization technique is sensitive to the hyperparameter dictating the tradeoff between gender and wage predictive components. While this approach may yield effective estimates for the right hyperparameter, it is difficult to assess how to set it.

5.4 Empirical Studies

Here, we estimate the history-adjusted gender wage ratio on real data. We begin by demonstrating that incorporating full histories into wage predictions improves predictive accuracy. We conclude by analyzing the results and identifying populations where histories do and do not explain the wage gap.

5.4.1 Wage prediction

We begin by assessing CAREER's performance at predicting wages on the survey data used by economists for estimating the wage gap. We follow Blau & Kahn (2017) and use the Panel Study of Income Dynamics (Panel Study of Income Dynamics, 2021), or PSID. PSID is a longitudinal survey that follows a cohort of American families with biannual interviews. We consider the seven most recent years available in PSID. We restrict our sample to wage and salary workers between 25 and 64 years old who worked for at least 26 weeks in non-farm jobs, following Blau & Kahn (2017). We include job histories to this dataset by matching across previous interviews for each individual. Jobs are encoded into one of 330 `occ1990dd` categories (Autor & Dorn, 2013). Each year sample contains about 4500 individuals.

We compare our proposed method, which learns low-dimensional representations of job histories, to an econometric baseline that does not use an individual's complete job history to predict wage (Blau & Kahn, 2017). Instead, this method uses hand-constructed summary statistics about each individual's job history, such as number of years worked full-time and part-time (and their squares) along with current occupation and industry. We use the additional covariates described in Blau & Kahn (2017): years of schooling, indicators for bachelors and advanced degrees, race and ethnicity indicators, census region indicators, and an indicator for collective bargaining coverage.[4] The functional form of the model is identical to the linear model that incorporates history into the prediction of wage (Equations 5.5 and 5.6), albeit without the term relating to history.

We leverage a dataset of 24 million resumes to enhance CAREER's representations.[5] When using these resumes to learn a representation, we include a job in a sequence with 50% probability; if the job is not included in the sequence, we use the included jobs to predict it, following Equation 5.10. We then follow the sufficiency-constrained optimization procedure described in Algorithm 3 to adjust these representations for wage gap estimation. PSID provides weights for

[4]Unlike Blau & Kahn (2017), we do not use metro area indicators, since this variable is only included in the restricted PSID sample, which we do not have access to. However, when we use the data sample released by Blau & Kahn (2017), we find little difference in comparative performance when using metro area indicator.

[5]This dataset is provided by Zippia, a career planning company.

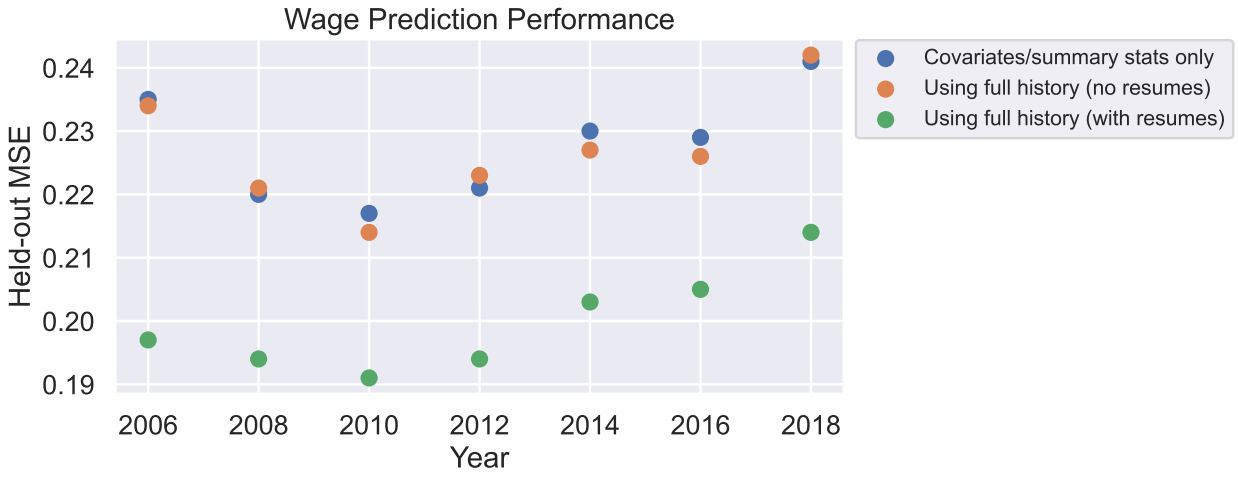


Figure 5.2: Mean-square error of wage prediction on held-out data for different years of the PSID survey. The best predictions are made by a model that learns a low-dimensional representation of job history, as opposed to the classical econometric model that summarizes history with summary statistics (Blau & Kahn, 2017). Transfer learning with a large-scale corpus of resumes is crucial to this predictive advantage.

each family that are meant to be used to construct a nationally representative sample. We use these weights to train and evaluate all wage prediction models. We split each PSID sample into 5 splits and report the average test error across splits. For more training details, see Algorithm 4 in Section D.1.

Figure 5.2 depicts the wage prediction results on held-out data. Across all years, the model that uses CAREER to represent history makes the best predictions of wage. Leveraging the large-scale resume dataset is necessary for this model to make high-quality predictions. Without using these resumes, the model with history performs on par with the classical model from Blau & Kahn (2017) that only uses hand-constructed summary statistics about the past. With the resumes, the model with history consistently outperforms this model by 10-15%. Table D.1 in Section D.2 shows these same results in tabular form.

As discussed, a first-order priority of estimating the gender wage gap is a model that makes good predictions of held-out wage. While the projection-based optimization procedure encourages the model to learn representations that are sufficient for gender, there is risk that projection degrades the wage predictions. Figure D.2 in Section D.2 shows wage prediction accuracy as a

function of projection round. We find that across years, sufficiency-constrained optimization does not hurt the predictive performance of wage. In fact, in all settings, a single round of projection *improves* the predictions of wage. This improvement may seem counter-intuitive. One possible explanation is that gender prediction provides a regularizing effect; if the features from history that predict wage are also predictive of gender, fine-tuning from a representation that is sufficient for gender encourages the model to use these features rather than overfit to spurious features.

5.4.2 Estimating and analyzing the history-adjusted gender wage gap

We now use the proposed methodology to estimate the history-adjusted gender wage gap. For each year, we split PSID survey data into five folds and estimate the adjusted wage gap using cross-fitting (Chernozhukov et al., 2018). We perform sufficiency-constrained optimization for each split. We compare our estimate of the history-adjusted wage gap to the raw wage gap and the classical covariate-adjusted gap (Equation 5.1) used by Blau & Kahn (2017). We estimate standard errors by bootstrapping the estimation process 100 times, keeping the learned representation of history fixed across bootstraps (but re-estimating the wage function conditional on history using OLS). For full training details, see Algorithm 4 in Section D.1.

The history-adjusted gender wage gap can only be estimated for subpopulations where there is overlap, i.e. $0 < P(G = F|H, X) < 1$. In order to ensure overlap, we discard examples where male or female propensities are above 95% (Crump et al., 2006). We perform this clipping for all wage gap estimates to ensure all estimates are comparable. This changes the subpopulation in the adjusted wage gap estimand: We are now estimating the adjusted gender wage gap for only the individuals *whose work histories are in the middle 90% of the gender distribution*. Although the absolute wage gap estimates vary across models for different thresholding parameters, we find that the relative wage gaps estimates are consistent across thresholds. See Section D.3 for more analysis of the effects of clipping.

Figure 5.3 presents the adjusted wage ratios (recall the adjusted wage ratio is the exponentiated adjusted wage gap). For each year, history explains between one-third to one-half of the wage gap

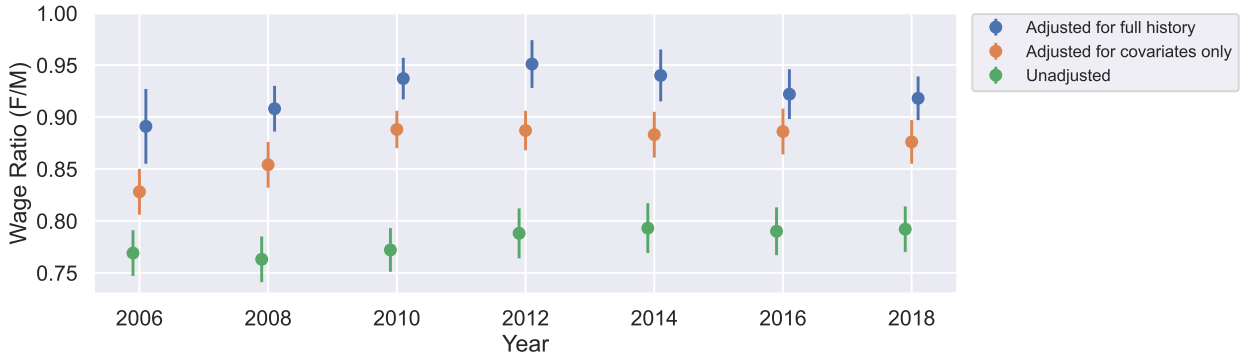


Figure 5.3: Estimates of the history-adjusted gender wage ratio on PSID, compared to the unadjusted wage ratio and the covariate-only adjusted wage ratio. Single standard errors are estimated by bootstrapping. These results are depicted in tabular form in Table D.3 in Section D.4.

left unexplained by other covariates. For example, consider workers from 2010. The unadjusted wage ratio is 0.772; this means that the (geometric) average female wage is 77.2% that of males'. The covariate-only adjusted wage ratio is 0.888, suggesting that for males and females with the same covariates (e.g. education level, industry) and history summary statistics (i.e. years of full- and part-time employment and current occupation), females earn 88.8% the wage of males. Finally, the estimate of the full-history adjusted wage gap is 0.937, suggesting that females with the same covariates and full histories as males earn 93.7% of their wage. In this case, history explains about 40% of the gender wage gap that is unexplained by other covariates, while the other 60% remains unexplained even when adjusting for full history. The trend does not appear to change meaningfully between 2006 and 2018. The results are depicted in tabular form in Table D.3 in Section D.4.

To assess how different factors contribute to the history-adjusted wage gap, we perform a Blinder-Oaxaca decomposition (Blinder, 1973; Oaxaca, 1973). The Blinder-Oaxaca relies on the fact that the raw wage gap,

$$R = \frac{1}{n_F} \sum_{i:G_i=F} Y_i - \frac{1}{n_M} \sum_{i:G_i=M} Y_i, \tag{5.11}$$

can be written as

$$R = \frac{1}{n_F} \sum_{i:G_i=F} \hat{E}[Y|G_i = F, X_i] - \frac{1}{n_M} \sum_{i:G_i=M} \hat{E}[Y|G_i = M, X_i], \tag{5.12}$$

when $\hat{E}[Y|G, X]$ has zero residual mean on the population it is being applied to (e.g. a linear regression fit with ordinary least squares has zero average residual on the sample it is fit to). By adding and subtracting the same term, we can rewrite the above as

$$R = \overbrace{\frac{1}{n_F} \sum_{i:G_i=F} (\hat{E}[Y|G_i = F, X_i] - \hat{E}[Y|G_i = M, X_i])}^{\text{unexplained wage gap}} \tag{5.13}$$

$$+ \underbrace{\frac{1}{n_F} \sum_{i:G_i=F} \hat{E}[Y|G_i = M, X_i] - \frac{1}{n_M} \sum_{i:G_i=M} \hat{E}[Y|G_i = M, X_i]}_{\text{explained wage gap}} \tag{5.14}$$

The term in Equation 5.14 is the portion of the wage gap that can be explained by covariates X; it is the difference in predicted wages for males with male covariates and males with female covariates. If this term is large, it is not because the fitted wage models for males and females differ, but rather because the distribution of covariates for males and females differ. The term in Equation 5.13 is the component of the wage gap that cannot be explained by covariates X; it is identical to an estimate of the adjusted wage gap (Equation 5.1). Together, these two terms decompose the raw wage gap into an explained component and an unexplained component. When the wage model is linear in its covariates, Equation 5.14 can further be broken down into the sum of terms corresponding to individual covariates.

Table 5.1 presents a Blinder-Oaxaca decomposition of the 2018 gender wage gap for both the classical non-history adjusted wage gap and the full-history adjusted wage gap.[6] The raw wage gap is decomposed into terms that capture how differences in specific covariates contribute to the wage gap, along with an unexplained term. For the classical model that does not adjust for history, current occupation explains 38.0% of the raw wage gap, by far explaining the most of the raw wage

[6]Although the Blinder-Oaxaca decomposition was developed for linear models, it can be used here because our model is linear in its covariates and final representation of history. The only modification that is required is that the model's residuals are not zero mean, i.e. Equation 5.12 does not hold, due to cross-fitting, nonlinearity, and clipping. Thus, we include a residual term in the decomposition.

	Percent of gender wage gap explained when adjusting for:			
	Summary statistics & covariates		Full history & covariates	
	log points	Percent of gap explained	log points	Percent of gap explained
Education variables	-0.011	-4.6%	-0.007	-3.0%
Experience summary statistics	0.021	8.9%	0.017	7.2%
Region variables	0.004	1.6%	0.005	1.9%
Race variables	0.005	2.2%	0.001	0.5%
Union status	0.000	-0.2%	0.000	0.0%
Current industry	0.005	2.3%	0.005	2.3%
Current occupation	0.089	38.0%	0.065	27.7%
Full history	—	—	0.070	30.1%
Non-zero residual	-0.012	-5.1%	-0.008	-3.4%
Total explained	0.112	48.2%	0.156	66.7%
Total unexplained	0.133	56.9%	0.086	36.7%

Table 5.1: Decomposing the 2018 gender wage gap using an Oaxaca-Blinder decomposition. The model in the first two columns follows Blau & Kahn (2017) and summarizes history with hand-constructed summary statistics; the model in the last two columns use the method developed in this chapter to adjust for full history. The sample includes wage and salary workers between 25 and 64 years old who worked for at least 26 weeks in non-farm jobs, and whose work histories are in the middle 90% of the gender distribution (to assure overlap). The non-zero residual term arises from the fact that the average wage prediction isn't necessarily equal to the empirical mean due to cross-fitting, nonlinear models, and data clipping.

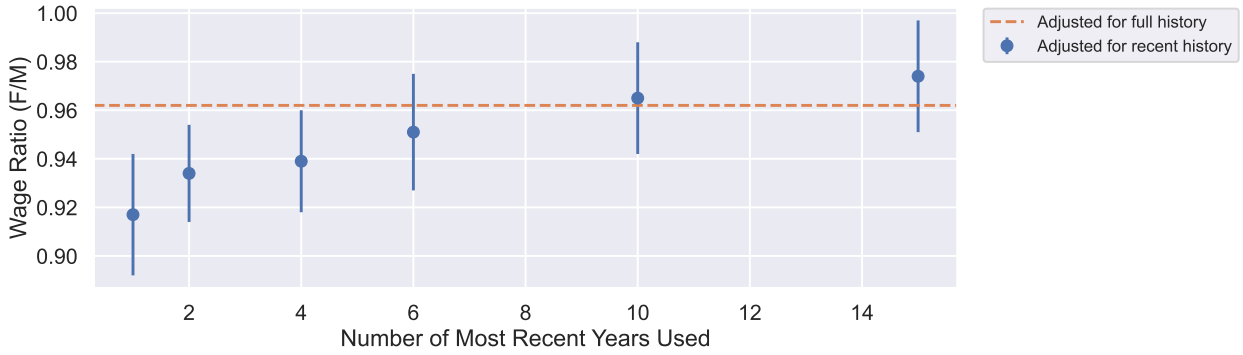


Figure 5.4: The history-adjusted gender wage ratio for 2010 as a function of the number of most recent jobs used in CAREER's representation of job history. Observations before each year cutoff are discarded and not used to estimate the conditional wage function or the adjusted wage gap.

gap. Meanwhile, occupation and job history differences account for a combined 57.8% of the raw wage gap according to the model that adjusts for full history. This number makes sense in context; if a single occupation can explain 38.0% of the raw wage gap, including more occupations via full histories should increase the explanatory power.

To understand how much history is relevant to explaining the wage gap, we estimate the adjusted wage gap for varying amounts of history. Specifically, we perform the estimation procedure above but fit a representation that only uses a fixed number of the most recent observations. For example, for estimating the adjusted wage ratio for the last five years of history, we transform each job sequence in our survey data by truncating all jobs before the last five. We then estimate the wage gap as above, using only the truncated sequences. By estimating the wage gap on truncated data, we can isolate the effect of varying lengths of job history.

Figure 5.4 depicts the adjusted wage ratios for different lengths of history. When only a few years are available, including each additional year substantially increases the adjusted wage ratio estimate. As longer histories are available, there is less of an effect when additional years are added. The effect plateaus after around 10 years. Figure D.4 in Section D.5 shows an analogous trend for predictive accuracy.

We now analyze subpopulations for which the wage gap can and cannot be explained by history. Figure D.5 in Section D.5 estimates the adjusted gender wage ratio for each of the 21 occupational

groups provided by Blau & Kahn (2017). History explains most of the wage gap for managers and production occupations. To further identify subpopulations where history explains a large portion of the wage gap, we cluster workers into subpopulations based on their representations of history. We perform a hierarchical clustering where we first divide workers into the 21 high-level occupational groups. We further divide clusters into high-paying histories and low-paying histories based on CAREER's representations. Within each remaining stratum, we project the representations to 2D using t-SNE (Van der Maaten & Hinton, 2008) and perform K-Means clustering with 3 clusters based on these 2D representations. We estimate the adjusted-gender wage ratio for each cluster in Table D.4 in Section D.5.

Here, we analyze one cluster from Table D.4. The cluster for which history explains the most of the wage gap comes from a cluster of current managers with high wage. Below are examples of career trajectories from this cluster:

- manager, manager, technician, manager, medical scientist, medical scientist, financial manager, manager
- student, student, student, student, social scientist, social scientist, manager, manager, manager
- employed, employed, employed, student, software developer, manager, manager, sales supervisors, software developer, software developer, software developer, manager, manager, manager

This cluster consists of managers whose recent jobs require specialized technical skills, such as financial or software developing skills. For this cluster, the non-history adjusted wage ratio between females and males is 0.744. Meanwhile, the full-history adjusted wage ratio is 0.846. Statistically, managers with these specialized jobs in their histories get paid more than managers without these jobs in their histories. A model that only conditions on an individual's current job cannot differentiate between managers with or without financial jobs in their recent history, so it is unable to explain this gap.

We also examine groups for which history does not explain the wage gap. We find that history does not explain the gap for physicians and lawyers. The non-history adjusted wage ratio is 0.81 for this group, compared to a full-history adjusted wage ratio of 0.80. To gain insight into why history does not explain the gap here, we include history samples for physicians in the sample below:

- student, student, physician
- student, student, lab technician, student, physician, physician, physician, physician, physician, physician, physician
- student, student, technician, child care worker, child care worker, physician, physician, physician, physician

We also include histories for a sample of lawyers:

- employed, lawyer, lawyer, lawyer, lawyer, lawyer, lawyer, lawyer
- student, student, student, student, student, student, student, employed, student, lawyer, lawyer, lawyer, lawyer, lawyer, lawyer, lawyer, lawyer, lawyer, lawyer, lawyer
- student, student, student, employed, lawyer, lawyer, lawyer, lawyer, lawyer, lawyer, lawyer, lawyer

For both physicians and lawyers, there is barely any variation across histories. Since there are no discernible differences in job histories for male and female physician or lawyers, conditioning on these histories does not explain the wage gap for the group.

5.5 Summary

This chapter introduced a method to estimate the adjusted gender wage gap for high-dimensional histories. We harnessed the rich predictive abilities of machine learning techniques to estimate this

quantity using small survey datasets. We found that history can explain roughly 1/2-1/3 of the wage gap previously unexplained by covariates. History explains a large portion of the gap for managers, while it cannot explain any of the gap for physicians and lawyers.

Conclusion

This thesis developed interpretable machine learning techniques to uncover latent patterns and estimate critical quantities in the social sciences. Chapter 2 presented a method for explaining the predictions of general, black-box sequence models. This method was able to elucidate the decision-making patterns of large language models and, later, a sequence model fit to job sequence data. Next, we focused on domain-specific applications. Chapter 3 presented the text-based ideal point model, a model that quantifies political positions from text. This model combined a classical idea from political science with a Bayesian matrix factorization technique to infer meaningful structure from text. Chapter 4 adapted a technique from natural language processing to analyze career trajectories. We relied on transfer learning to overcome the constraints posed by the small survey datasets used for labor economics research. Finally, in Chapter 5, we used this predictive model to estimate an important quantity in labor economics: the history-adjusted gender wage gap. We showed that job histories could explain a substantial portion of the wage gap, but that the magnitude varies across subpopulations.

Below, we outline general areas for future research:

- *Interpreting sequence models with causal ideas:* The greedy rationalization method in Chapter 2 produced rationales of sequence models by observing differences in the predictive distribution induced by different rationales. However, causal ideas can be used to enhance the interpretability of black-box models: are certain tokens *necessary* for individual predictions? Incorporating causal reasoning into rationalization methods is a promising direction of future research.

- *Contextual text-based ideal points:* The text-based ideal point model in Chapter 3 uses a bag-of-words assumption to model text. However, this assumption has weaknesses: the meaning of words is contextual, and the model is also sensitive to stopword curation. Combining recent advances in language modeling with Bayesian ideas can help extract more robust latent variables.

- *Fair pretraining:* The estimation method in Chapter 5 leverages passively-collected resume data in its estimate of the adjusted wage gap. As discussed, passively-collected resume datasets are not curated to represent national demographics. Pretraining CAREER on these datasets may result in representations that are affected by sampling bias. Although these representations are fine-tuned on survey datasets that are carefully constructed to represent national demographics, the biases from pretraining may propagate through fine-tuning (Ravfogel et al., 2020; Jin et al., 2021). Future work should prioritize techniques for assessing and mitigating the biases that ensue due to these distribution differences.

References

Abnar, S. and Zuidema, W. Quantifying attention flow in transformers. In *Association for Computational Linguistics*, 2020.

Alaa, A. and Van Der Schaar, M. Validating causal inference models via influence functions. In *International Conference on Machine Learning*, pp. 191–201. PMLR, 2019.

Altonji, J. G. and Blank, R. M. Race and gender in the labor market. *Handbook of labor economics*, 3:3143–3259, 1999.

Alvarez-Melis, D. and Jaakkola, T. S. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Association for Computational Linguistics*, 2017.

Arnold, B. C. and Press, S. J. Compatible conditional distributions. *Journal of the American Statistical Association*, 84(405):152–156, 1989.

Assaad, S., Zeng, S., Tao, C., Datta, S., Mehta, N., Henao, R., Li, F., and Carin, L. Counterfactual representation learning with balancing weights. In *International Conference on Artificial Intelligence and Statistics*, pp. 1972–1980. PMLR, 2021.

Athey, S., Imbens, G. W., and Wager, S. Approximate residual balancing: Debiased inference of average treatment effects in high dimensions. *Journal of the Royal Statistical Society*, 80(4): 597–623, 2018.

Autor, D. and Dorn, D. The growth of low-skill service jobs and the polarization of the U.S. labor market. *American Economic Review*, 103(5):1553–97, 2013.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv:1607.06450*, 2016.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11: 1803–1831, 2010.

Bailey, M. Ideal point estimation with a small number of votes: A random-effects approach. *Political Analysis*, 9(3):192–210, 2001.

Bana, S. H. Using language models to understand wage premia. 2021.

Bastings, J. and Filippova, K. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *ACL Workshop on BlackboxNLP*, 2020.

Bastings, J., Aziz, W., and Titov, I. Interpretable neural predictions with differentiable binary variables. In *Association for Computational Linguistics*, 2019.

Baumer, E., Elovic, E., Qin, Y., Polletta, F., and Gay, G. Testing and comparing computational approaches for identifying the language of framing in political news. In *Proceedings of ACL*, 2015.

Baumgartner, F. R., De Boef, S. L., and Boydstun, A. E. *The decline of the death penalty and the discovery of innocence*. Cambridge University Press, 2008.

Belloni, A., Chernozhukov, V., and Hansen, C. High-dimensional methods and inference on structural and treatment effects. *Journal of Economic Perspectives*, 28(2):29–50, 2014.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155, 2003.

Benoit, K. and Laver, M. Estimating Irish party policy positions using computer wordscoring: The 2002 election. *Irish Political Studies*, 18(1):97–107, 2003.

Bertrand, M., Goldin, C., and Katz, L. F. Dynamics of the gender gap for young professionals in the financial and corporate sectors. *American economic journal: applied economics*, 2(3): 228–55, 2010.

Blau, D. M. and Riphahn, R. T. Labor force transitions of older married couples in Germany. *Labour Economics*, 6(2):229–252, 1999.

Blau, F. D. and Kahn, L. M. The feasibility and importance of adding measures of actual experience to cross-sectional data collection. *Journal of Labor Economics*, 31(S1):S17–S58, 2013.

Blau, F. D. and Kahn, L. M. The gender wage gap: Extent, trends, and explanations. *Journal of Economic Literature*, 55(3):789–865, 2017.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Blinder, A. S. Wage discrimination: reduced form and structural estimates. *Journal of Human Resources*, 8(4):436–455, 1973.

Bock, R. D. and Aitkin, M. Marginal maximum likelihood estimation of item parameters: Application of an EM algorithm. *Psychometrika*, 46(4):443–459, 1981.

Boskin, M. J. A conditional logit model of occupational choice. *Journal of Political Economy*, 82 (2, Part 1):389–398, 1974.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R. L. The mathematics of statistical machine translation: Parameter estimation. In *Association for Computational Linguistics*, 1993.

Brown, R. S., Moon, M., and Zoloth, B. S. Incorporating occupational attainment in studies of male-female earnings differentials. *Journal of Human Resources*, 15(1):3–28, 1980.

Brunner, G., Liu, Y., Pascual, D., Richter, O., Ciaramita, M., and Wattenhofer, R. On identifiability in transformers. In *International Conference on Learning Representations*, 2019.

Bureau of Labor Statistics. National Longitudinal Survey of Youth 1979 cohort (rounds 1-27). Produced and distributed by the Center for Human Resource Research (CHRR), The Ohio State University. Columbus, OH, 2019a.

Bureau of Labor Statistics. National Longitudinal Survey of Youth 1997 cohort (rounds 1-19). Produced and distributed by the Center for Human Resource Research (CHRR), The Ohio State University. Columbus, OH, 2019b.

Canny, J. GaP: A factor model for discrete data. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004.

Card, D., Boydstun, A., Gross, J. H., Resnik, P., and Smith, N. A. The media frames corpus: Annotations of frames across issues. In *Proceedings of ACL*, 2015.

Card, D., Cardoso, A. R., and Kline, P. Bargaining, sorting, and the gender wage gap: Quantifying the impact of firms on the relative pay of women. *The Quarterly journal of economics*, 131(2): 633–686, 2016.

Cemgil, A. T. Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, 2009, 2009.

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., and Federico, M. Report on the 11th IWSLT evaluation campaign. In *International Workshop on Spoken Language Translation*, 2014.

Chen, J., Song, L., Wainwright, M., and Jordan, M. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, 2018.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. Double/debiased machine learning for treatment and structural parameters, 2018.

Chernozhukov, V., Cinelli, C., Newey, W., Sharma, A., and Syrgkanis, V. Long story short: Omitted variable bias in causal machine learning. Technical report, National Bureau of Economic Research, 2022a.

Chernozhukov, V., Newey, W., Quintas-Martınez, V. M., and Syrgkanis, V. RieszNet and ForestRiesz: Automatic debiased machine learning with neural nets and random forests. In *International Conference on Machine Learning*, pp. 3901–3914. PMLR, 2022b.

Chong, D. and Druckman, J. N. Framing theory. *Annual Review of Political Science*, 10:103–126, 2007.

Chvatal, V. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

Clinton, J., Jackman, S., and Rivers, D. The statistical analysis of roll call data. *American Political Science Review*, 98(2):355–370, 2004.

Cohen, J. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

Cortes, G. M. Where have the middle-wage workers gone? A study of polarization using panel data. *Journal of Labor Economics*, 34(1):63–105, 2016.

Crump, R. K., Hotz, V. J., Imbens, G., and Mitnik, O. Moving the goalposts: Addressing limited overlap in the estimation of average treatment effects by changing the estimand, 2006.

Dave, V. S., Zhang, B., Al Hasan, M., AlJadda, K., and Korayem, M. A combined representation learning approach for better job and skill recommendation. In *ACM Conference on Information and Knowledge Management*, 2018.

Denil, M., Demiraj, A., and De Freitas, N. Extraction of salient sentences from labelled documents. In *International Conference on Learning Representations*, 2014.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

DeYoung, J., Jain, S., Rajani, N. F., Lehman, E., Xiong, C., Socher, R., and Wallace, B. C. ERASER: A benchmark to evaluate rationalized NLP models. In *Association for Computational Linguistics*, 2020.

Dietterich, T. G. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pp. 1–15. Springer, 2000.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Entman, R. M. Framing: Toward clarification of a fractured paradigm. *Journal of Communication*, 43(4):51–58, 1993.

Fairlie, R. W. and Sundstrom, W. A. The emergence, persistence, and recent widening of the racial unemployment gap. *Industrial and Labor Relations Review*, 52(2):252–270, 1999.

Feng, S., Wallace, E., Grissom II, A., Iyyer, M., Rodriguez, P., and Boyd-Graber, J. Pathologies of neural models make interpretations difficult. In *Association for Computational Linguistics*, 2018.

Fortin, N., Lemieux, T., and Firpo, S. Decomposition methods in economics. In *Handbook of Labor Economics*, volume 4, pp. 1–102. Elsevier, 2011.

Gentzkow, M., Shapiro, J. M., and Taddy, M. Congressional record for the 43rd-114th Congresses: Parsed speeches and phrase counts. *Stanford Libraries*, 2018. URL `https://data.stanford.edu/congress_text`.

Gentzkow, M., Shapiro, J. M., and Taddy, M. Measuring group differences in high-dimensional choices: Method and application to congressional speech. *Econometrica*, 87(4):1307–1340, 2019.

Gerrish, S. and Blei, D. M. Predicting legislative roll calls from text. In *Proceedings of ICML*, 2011.

Gerrish, S. and Blei, D. M. How they vote: Issue-adjusted models of legislative behavior. In *Proceedings of NeurIPS*, 2012.

Ghorbani, A., Abid, A., and Zou, J. Interpretation of neural networks is fragile. In *Association for the Advancement of Artificial Intelligences*, 2019.

Ghosh, A., Woolf, B., Zilberstein, S., and Lan, A. Skill-based career path modeling and recommendation. In *IEEE International Conference on Big Data*, 2020.

Gokaslan, A. and Cohen, V. Openwebtext corpus. `http://Skylion007.github.io/OpenWebTextCorpus`, 2019.

Gopalan, P., Hofman, J. M., and Blei, D. M. Scalable recommendation with Poisson factorization. In *Proceedings of UAI*, 2015.

Gopalan, P. K., Charlin, L., and Blei, D. M. Content-based recommendations with Poisson factorization. In *Proceedings of NeurIPS*, 2014.

Guvenen, F., Kuruscu, B., Tanaka, S., and Wiczer, D. Multidimensional skill mismatch. *American Economic Journal: Macroeconomics*, 12(1):210–44, 2020.

Guyon, I. and Elisseeff, A. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.

Hahn, P. R., Carvalho, C. M., Puelz, D., and He, J. Regularization and confounding in linear regression for treatment effect estimation. *Bayesian Analysis*, 13(1):163–182, 2018.

Hall, R. E. Turnover in the labor force. *Brookings Papers on Economic Activity*, 1972(3):709–764, 1972.

Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning: Data mining, inference, and prediction*. Springer Science & Business Media, 2009.

He, M., Zhan, X., Shen, D., Zhu, Y., Zhao, H., and He, R. What about your next job? predicting professional career trajectory using neural networks. In *Machine Learning and Machine Intelligence*, 2021.

Hendrycks, D. and Gimpel, K. Gaussian error linear units (GELUs). *arXiv:1606.08415*, 2016.

Herzog, A. and Benoit, K. The most unkindest cuts: Speaker selection and expressed government dissent during economic crisis. *The Journal of Politics*, 77(4):1157–1175, 2015.

Ho, D. E., Quinn, K. M., et al. Measuring explicit political positions of media. *Quarterly Journal of Political Science*, 3(4):353–377, 2008.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Hofman, J. M., Watts, D. J., Athey, S., Garip, F., Griffiths, T. L., Kleinberg, J., Margetts, H., Mullainathan, S., Salganik, M. J., Vazire, S., et al. Integrating explanation and prediction in computational social science. *Nature*, 595(7866):181–188, 2021.

Hooker, S., Erhan, D., Kindermans, P.-J., and Kim, B. A benchmark for interpretability methods in deep neural networks. *Neural Information Processing Systems*, 2019.

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A. M., Hoffman, M. D., Dinculescu, M., and Eck, D. Music transformer: Generating music with long-term structure. In *International Conference on Learning Representations*, 2019.

Huber, M. Causal pitfalls in the decomposition of wage gaps. *Journal of Business & Economic Statistics*, 33(2):179–191, 2015.

Imai, K., Lo, J., and Olmsted, J. Fast estimation of ideal points with massive data. *American Political Science Review*, 110(4):631–656, 2016.

Iyyer, M., Enns, P., Boyd-Graber, J., and Resnik, P. Political ideology detection using recursive neural networks. In *Proceedings of ACL*, 2014.

Jackman, S. Multidimensional analysis of roll call data via Bayesian simulation: Identification, estimation, inference, and model checking. *Political Analysis*, 9(3):227–241, 2001.

Jacovi, A. and Goldberg, Y. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Association for Computational Linguistics*, 2020.

Jain, S. and Wallace, B. C. Attention is not explanation. In *North American Chapter of the Association for Computational Linguistics*, 2019.

Jain, S., Wiegreffe, S., Pinter, Y., and Wallace, B. C. Learning to faithfully rationalize by construction. In *Association for Computational Linguistics*, 2020.

Jethani, N., Sudarshan, M., Aphinyanaphongs, Y., and Ranganath, R. Have we learned to explain?: How interpretability methods can learn to encode predictions in their interpretations. In *Artificial Intelligence and Statistics*, 2021.

Jin, X., Barbieri, F., Kennedy, B., Davani, A. M., Neves, L., and Ren, X. On transferability of bias mitigation effects in language model fine-tuning. In *North American Chapter of the Association for Computational Linguistics*, 2021.

Johansson, F., Shalit, U., and Sontag, D. Learning representations for counterfactual inference. In *International conference on machine learning*, pp. 3020–3029. PMLR, 2016.

Johnson, K., Jin, D., and Goldwasser, D. Leveraging behavioral and social information for weakly supervised collective classification of political discourse on Twitter. In *Proceedings of ACL*, 2017.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

Kádár, A., Chrupała, G., and Alishahi, A. Representation of linguistic form and function in recurrent neural networks. In *Association for Computational Linguistics*, 2017.

Kambourov, G. and Manovskii, I. Rising occupational and industry mobility in the United States: 1968–97. *International Economic Review*, 49(1):41–79, 2008.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv:2001.08361*, 2020.

Keane, M. P. and Wolpin, K. I. The career decisions of young men. *Journal of Political Economy*, 105(3):473–522, 1997.

Kim, I. S., Londregan, J., and Ratkovic, M. Estimating spatial preferences from votes and text. *Political Analysis*, 26(2):210–229, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *Proceedings of ICLR*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. In *Proceedings of ICLR*, 2014.

Kobayashi, G., Kuribayashi, T., Yokoi, S., and Inui, K. Attention is not only a weight: Analyzing transformers with vector norms. In *Association for Computational Linguistics*, 2020.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Lauderdale, B. E. and Clark, T. S. Scaling politically meaningful dimensions using texts and votes. *American Journal of Political Science*, 58(3):754–771, 2014.

Lauderdale, B. E. and Herzog, A. Measuring political positions from legislative speech. *Political Analysis*, 24(3):374–394, 2016a.

Lauderdale, B. E. and Herzog, A. Replication data for: Measuring political positions from legislative speech. *Harvard Dataverse*, 2016b. URL `https://doi.org/10.7910/DVN/RQMIV3`.

Lauerova, J. S. and Terrell, K. What drives gender differences in unemployment? *Comparative Economic Studies*, 49(1):128–155, 2007.

Laver, M., Benoit, K., and Garry, J. Extracting policy positions from political texts using words as data. *American Political Science Review*, 97(2):311–331, 2003.

Lee, D. D. and Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

Lei, T., Barzilay, R., and Jaakkola, T. Rationalizing neural predictions. In *Association for Computational Linguistics*, 2016.

Lewis, J. B., Poole, K., Rosenthal, H., Boche, A., Rudkin, A., and Sonnet, L. Voteview: Congressional roll-call votes database. 2020. URL `https://voteview.com/`.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Li, J., Chen, X., Hovy, E., and Jurafsky, D. Visualizing and understanding neural models in NLP. In *Association for Computational Linguistics*, 2016a.

Li, J., Monroe, W., and Jurafsky, D. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016b.

Li, L., Jing, H., Tong, H., Yang, J., He, Q., and Chen, B.-C. NEMO: Next career move prediction with contextual embedding. In *World Wide Web Conference*, 2017.

Lipton, Z. C. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. In *Queue*, volume 16, pp. 31–57. ACM New York, NY, USA, 2018.

Lo, J., Proksch, S.-O., and Slapin, J. B. Ideological clarity in multiparty competition: A new measure and test using election manifestos. *British Journal of Political Science*, 46(3):591–610, 2016.

Lowe, W. Understanding wordscores. *Political Analysis*, 16(4):356–371, 2008.

Martin, A. D. and Quinn, K. M. Dynamic ideal point estimation via Markov Chain Monte Carlo for the US Supreme Court, 1953–1999. *Political Analysis*, 10(2):134–153, 2002.

McCall, B. P. Occupational matching: A test of sorts. *Journal of Political Economy*, 98(1):45–69, 1990.

McCarty, N. M., Poole, K. T., and Rosenthal, H. *Income redistribution and the realignment of American politics*. American Enterprise Institute Press, 1997.

Meng, Q., Zhu, H., Xiao, K., Zhang, L., and Xiong, H. A hierarchical career-path-aware neural network for job mobility prediction. In *ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. Efficient estimation of word representations in vector space. In *Workshop Track at ICLR*, 2013a.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013b.

Mohankumar, A. K., Nema, P., Narasimhan, S., Khapra, M. M., Srinivasan, B. V., and Ravindran, B. Towards transparent and explainable attention models. In *Association for Computational Linguistics*, 2020.

Mullainathan, S. and Spiess, J. Machine learning: an applied econometric approach. *Journal of Economic Perspectives*, 31(2):87–106, 2017.

Naudé, M., Adebayo, K. J., and Nanda, R. A machine learning approach to detecting fraudulent job types. *AI & Society*, pp. 1–12, 2022.

Neal, D. The complexity of job mobility among young men. *Journal of Labor Economics*, 17(2): 237–261, 1999.

Nguyen, V.-A., Boyd-Graber, J., Resnik, P., and Miler, K. Tea Party in the house: A hierarchical ideal point topic model and its application to Republican legislators in the 112th Congress. In *Proceedings of ACL*, 2015.

Noonan, M. C., Corcoran, M. E., and Courant, P. N. Pay differences among the highly trained: Cohort differences in the sex gap in lawyers' earnings. *Social forces*, 84(2):853–872, 2005.

Oaxaca, R. Male-female wage differentials in urban labor markets. *International economic review*, 14(3):693–709, 1973.

Och, F. J. and Ney, H. Improved statistical alignment models. In *Association for computational linguistics*, 2000.

Ott, M., Edunov, S., Grangier, D., and Auli, M. Scaling neural machine translation. In *Conference on Machine Translation (WMT)*, 2018.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. Fairseq: A fast, extensible toolkit for sequence modeling. In *Association for Computational Linguistics*, 2019.

Panel Study of Income Dynamics. Public use dataset, produced and distributed by the Survey Research Center, Institute for Social Research, University of Michigan, Ann Arbor, MI, 2021.

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, Q. N., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., and Fernández, R. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Association for Computational Linguistics*, 2016.

Pearl, J. Direct and indirect effects. *Probabilistic and Causal Inference: The Works of Judea Pearl*, pp. 373, 2001.

Perez, E., Kiela, D., and Cho, K. Rissanen data analysis: Examining dataset characteristics via description length. *arXiv preprint arXiv:2103.03872*, 2021.

Poerner, N., Roth, B., and Schütze, H. Evaluating neural network explanation methods using hybrid documents and morphological agreement. In *Association for Computational Linguistics*, 2018.

Poole, K. T. *Spatial models of parliamentary voting*. Cambridge University Press, 2005.

Poole, K. T. and Rosenthal, H. A spatial model for legislative roll call analysis. *American Journal of Political Science*, pp. 357–384, 1985.

Poole, K. T. and Rosenthal, H. *Congress: A political-economic history of roll call voting*. Oxford University Press on Demand, 2000.

Poterba, J. M. and Summers, L. H. Reporting errors and labor market dynamics. *Econometrica*, 54(6):1319–1338, 1986.

Proksch, S.-O. and Slapin, J. B. How to avoid pitfalls in statistical analysis of political texts: The case of Germany. *German Politics*, 18(3):323–344, 2009.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training. 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Rajkumar, K., Simon, L., and Athey, S. *A Bayesian Approach to Predicting Occupational Transitions*. PhD thesis, Stanford University, 2021.

Ranganath, R., Tang, L., Charlin, L., and Blei, D. M. Deep exponential families. In *Proceedings of AISTATS*, 2015.

Ravfogel, S., Elazar, Y., Gonen, H., Twiton, M., and Goldberg, Y. Null it out: Guarding protected attributes by iterative nullspace projection. In *Association for Computational Linguistics*, 2020.

RealClearPolitics. Bernie Sanders on USA Freedom Act: "I may well be voting for it," does not go far enough, May 2015. URL https://www.realclearpolitics.com/video/2015/05/31/bernie_sanders_on_usa_freedom_act_i_may_well_be_voting_for_it_does_not_go_far_enough.html. Online; posted 31-May-2015.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of ICML*, 2014.

Ribeiro, M. T., Singh, S., and Guestrin, C. "Why should I trust you?" Explaining the predictions of any classifier. In *Special Interest Group on Knowledge Discovery and Data*, 2016.

Rissanen, J. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

Robins, J. M. and Greenland, S. Identifiability and exchangeability for direct and indirect effects. *Epidemiology*, pp. 143–155, 1992.

Rosenbaum, P. R. and Rubin, D. B. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Ruiz, F. J. R., Athey, S., and Blei, D. M. SHOPPER: A probabilistic model of consumer choice with substitutes and complements. *Annals of Applied Statistics*, 14(1):1–27, 2020.

Rush, A. M., Chopra, S., and Weston, J. A neural attention model for abstractive sentence summarization. In *Association for Computational Linguistics*, 2015.

Schmidt, P. and Strauss, R. P. The prediction of occupation using multiple logit models. *International Economic Review*, 16(2):471–486, 1975.

Schubert, G., Stansbury, A., and Taska, B. Employer concentration and outside options. *SSRN:3599454*, 2021.

Schwaller, P., Laino, T., Gaudin, T., Bolgar, P., Hunter, C. A., Bekas, C., and Lee, A. A. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS Central Science*, 5(9):1572–1583, 2019.

Schwarz, D., Traber, D., and Benoit, K. Estimating intra-party preferences: Comparing speeches to votes. *Political Science Research and Methods*, 5(2):379–396, 2017.

Serrano, S. and Smith, N. A. Is attention interpretable? In *Association for Computational Linguistics*, 2019.

Shi, C., Blei, D., and Veitch, V. Adapting neural networks for the estimation of treatment effects. In *Neural Information Processing Systems*, 2019.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Slapin, J. B. and Proksch, S.-O. A scaling model for estimating time-series party positions from texts. *American Journal of Political Science*, 52(3):705–722, 2008.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1): 1929–1958, 2014.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, 2017.

Tsur, O., Calacci, D., and Lazer, D. A frame of mind: Using statistical models for detection of framing and agenda setting campaigns. In *Proceedings of ACL*, 2015.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Variš, D. and Bojar, O. Sequence length is a domain: Length-based overfitting in transformer models. In *Empirical Methods in Natural Language Processing*, 2021.

Vashishth, S., Upadhyay, S., Tomar, G. S., and Faruqui, M. Attention interpretability across NLP tasks. *arXiv preprint arXiv:1909.11218*, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Neural Information Processing Systems*, 2017.

Veitch, V., Sridhar, D., and Blei, D. Adapting text embeddings for causal inference. In *Conference on Uncertainty in Artificial Intelligence*, pp. 919–928. PMLR, 2020.

Voita, E., Sennrich, R., and Titov, I. Analyzing the source and target contributions to predictions in neural machine translation. In *Association for Computational Linguistics*, 2021.

VoxGovFEDERAL. U.S. senators tweets from the 114th Congress, 2020. URL `https://www.voxgov.com`.

Wainwright, M. J., Jordan, M. I., et al. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

Wang, J., Tuyls, J., Wallace, E., and Singh, S. Gradient-based analysis of NLP models is manipulable. In *Empirical Methods in Natural Language Processing*, 2020.

Webber, D. A. Firm-level monopsony and the gender pay gap. *Industrial Relations: A Journal of Economy and Society*, 55(2):323–345, 2016.

Wexler, M. N. Successful resume fraud: Conjectures on the origins of amorality in the workplace. *Journal of Human Values*, 12(2):137–152, 2006.

Wiegreffe, S. and Pinter, Y. Attention is not not explanation. In *Empirical Methods in Natural Language Processing*, 2019.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Transformers: State-of-the-art natural language processing. In *Empirical Methods in Natural Language Processing*, 2019.

Xu, H., Yu, Z., Yang, J., Xiong, H., and Zhu, H. Dynamic talent flow analysis with deep sequence prediction modeling. *IEEE Transactions on Knowledge and Data Engineering*, 31(10):1926–1939, 2018.

Yoon, J., Jordon, J., and van der Schaar, M. INVASE: Instance-wise variable selection using neural networks. In *International Conference on Learning Representations*, 2018.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, 2014.

Zhang, D., Liu, J., Zhu, H., Liu, Y., Wang, L., Wang, P., and Xiong, H. Job2Vec: Job title benchmarking with collective multi-view representation learning. *arXiv preprint arXiv:2009.07429*, 2020.

Zhang, L., Zhou, D., Zhu, H., Xu, T., Zha, R., Chen, E., and Xiong, H. Attentive heterogeneous graph embedding for job mobility prediction. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021.

Appendix A: Rationales for Sequential Predictions

A.1 Algorithm Details

We present greedy rationalization in Algorithm 1.

Algorithm 1: Greedy rationalization

Input: Sequence $y_{1:t}$ generated from p.
Output: Rationale S for y_t.
Initialize: $S = \emptyset$
while $\arg\max_{y'_t} p(y'_t|y_S) \neq y_t$ **do**
  $k^* = \arg\max_{k \in [t-1] \setminus S} p(y_t|y_{S \cup k})$
  $S = S \cup k^*$
end
return S

Mose sequence models, including transformers, use the representation of a token y_{t-1} to predict the next token, y_t. As such, a rationale S always needs to contain y_{t-1}. In practice, we initialize $S = \{y_{t-1}\}$.

This method and paradigm extend easily to conditional sequence models, such as those used in machine translation. In this setting, a model uses a source sequence $x_{1:N}$ to generate a target sequence $y_{1:T}$. Thus, a context for a prediction y_t contains both $y_{<t}$ and $x_{1:N}$. The set of all possible rationales is the cross product of power sets $\mathcal{S} = 2^{[N]} \times 2^{[t-1]}$, and the combinatorial objective is

$$S(x_{1:N}, y_{1:t}) = \underset{S_x, S_y \in \mathcal{S}}{\arg\min} |S_x| + |S_y| \quad \text{s.t.} \quad \underset{y'_t}{\arg\max}\, p(y'_t|x_{S_x}, y_{S_y}) = y_t. \tag{A.1}$$

To perform greedy rationalization in this setting, we consider adding either a source token or a target token at each step, choosing the one that results in the largest increase in the full model's prediction.

A.2 Optimality of Deterministic Rationales

Here, we show that the selection distribution $q(S|x, y)$ that maximizes the classification rationale objective in Equation 2.8 is deterministic. We re-write the objective below:

$$\mathbb{E}_{x,y\sim F}\,\mathbb{E}_{S\sim q(S|x,y)}[\log p(y|x_S) - \lambda|S|]. \tag{A.2}$$

Theorem 1. For any $p(y|x_S)$, the $q(S|x, y)$ that maximizes Equation A.2 is a point-mass.

Proof. Denote by $g(x, y, S) = \log p(y|x_S) - \lambda|S|$. Then:

$$\begin{aligned}\max_q\ \mathbb{E}_{x,y\sim F}\,\mathbb{E}_{S\sim q(S|x,y)}[g(x, y, S)] &\leq \max_q\,\mathbb{E}_{x,y\sim F}\,\max_S[g(x, y, S)] \\ &= \mathbb{E}_{x,y\sim F}\,\max_S[g(x, y, S)].\end{aligned}$$

The inequality uses the fact that the expectation of a random variable is bounded by its maximum value. When $q(S|x, y)$ is a point-mass at $\arg\max_S[g(x, y, S)]$, the inequality becomes tight. □

The fact that the optimal rationale is deterministic for each example justifies using combinatorial strategies such as our objective in Equation 2.4.

A.3 Efficiency

In Table A.1, we provide a detailed version of our complexity analysis from Section 2.3: For transformers, greedy rationalization can be performed at no extra asymptotic complexity if the rationale length is $O(t^{1/3})$ for a sequence length t.

We evaluate the computational efficiency of greedy rationalization in Table A.2. We compare greedy rationalization to an exhaustive search, which enumerates all possible context context subsets from shortest to longest to optimize Equation 2.4. To show the efficiency of evaluating transformers on arbitrarily sized inputs, we also compare to a version of greedy rationalization that evaluates a transformer on the full input. To make predictions on sparse subsets, this approach

Step	Complexity	Evaluations	Total
1	1^2	t	$1^2 t$
2	2^2	$t-1$	$2^2(t-1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$O(t^{1/3})$	$O(t^{2/3})$	$O(t)$	$O(t^{5/3})$
Total			$O(t^2)$

Table A.1: For transformers, the asymptotic complexity of greedy rationalization matches the asymptotic complexity of forming a single prediction on the full sequence, as long as the rationale size is $O(t^{1/3})$ for a sequence of length t.

Method	**Time (s)**
Exhaustive search	>60
Greedy rationalization with full inputs	1.22
Greedy rationalization with sparse inputs	0.30

Table A.2: Greedy rationalization is efficient, especially when evaluating transformers on sparse inputs. We report the average wall clock time in seconds for finding rationales on the templated analogies dataset of Mikolov et al. (2013a). We cannot complete exhaustive search for the longer examples, so in reality the average runtime is larger than the listed one.

masks tokens that aren't in a candidate rationale during each attention step. In contrast, the efficient version of greedy rationalization only takes as input the tokens in the candidate rationale, so there is no need for masking.

We perform these comparisons on the templated analogies dataset of Mikolov et al. (2013a). We use GPT-2 Large as our sequence model (Radford et al., 2019) and perform each method on a single GPU. We compare the two greedy rationalization approaches for all of the examples for which the full model predicts the templated output. Since exhaustive search is intractable, we cannot perform it on every example due to computational constraints. Thus, we only run exhaustive search on examples where the optimal rationale has 6 or less tokens. In reality, the average runtime for exhaustive search is larger than the listed one.

A.4 Training and Fine-Tuning Details

Our experiments consist of three models and datasets: a transformer decoder (Vaswani et al., 2017) trained on a majority-class language, GPT-2 (Radford et al., 2019) fine-tuned on Open WebText (Gokaslan & Cohen, 2019), and a transformer machine translation model trained and fine-tuned with word dropout on IWSLT14 De-En (Cettolo et al., 2014).

For the majority-class language, we generate the dataset as described in Section 2.4. We include 50,000 examples in the training set, 5,000 in the validation set, and 5,000 in the test set.

We use a 4-layer transformer decoder with 2 attention heads per layer. We use an embedding dimension of 64, and a hidden dimension of 256 for the feedforward layers. This corresponds to 200,000 parameters. We train with 0.1 weight dropout, and optimize using Adam (Kingma & Ba, 2015) with a learning rate of 0.005 and an inverse square root learning rate scheduler. We use a warmup period of 4000 steps and an initial warmup learning rate of 10^{-7}. We include a maximum of 64,000 tokens in each batch. We implement this model in Fairseq (Ott et al., 2019).

To approximate the compatibility objective in Equation 2.7, we train with varying amounts of word dropout. In practice, this amounts to masking out each token we drop out at each attention layer. We use two levels of word dropout in Figure 2.3; none (which corresponds to training with the standard maximum likelihood objective in Equation 2.1) and 0.5. We train each model on a single GPU. Each model takes less than 20,000 steps to converge, less than 90 minutes. Table A.3 verifies that fine-tuning with word dropout does not hurt the heldout perplexity.

To fine-tune GPT-2, we use the pretrained GPT-2 Large model available on Hugging Face (Wolf et al., 2019). This model has 774M parameters. We don't change any of the model settings when we fine-tune. Sampling context subsets uniformly at random as stated in the objective in Equation 2.7 results in a distribution of subsets heavily skewed towards those containing half the words in the sequence. This is fine for the majority-class language, since each sequence contains less than 20 tokens and thus all possible context sizes will be seen during training. However, GPT-2's sequence length is 1,024. 99% of the time, sampling from the objective as stated would result

in contexts with size 464-560. Notably, the probability of a context with less than 10 tokens is less than 10^{-284}.

We make two adjustments to make sure the model is trained on both small and large subsets. With probability 0.5, we condition on the full context. With the remaining 0.5, we first randomly sample context sizes uniformly at random from 1 to the sequence length. We then sample a random context subset of this size. This guarantees that all possible sequence lengths will be seen during training.

Since the WebText dataset used to train GPT-2 is not publicly available, we use Open WebText (Gokaslan & Cohen, 2019), an open source reproduction effort. The corpus is in English. Rather than using the entire dataset, we take "Subset 9" and use the first 163M words. Our validation set is also from this subset and contains 160,000 words. We use a test set of 300,000 words from a different subset.

We fine-tune GPT-2 Large using Adam. We use a constant learning rate of 0.0001, using a single batch per training step. We stop training after 62,500 steps. This takes 15 hours on a single GPU. Table A.3 shows that fine-tuning with word dropout actually improves the heldout perplexity, although we believe that the improvement is due to our test set bearing more resemblance to the fine-tuning set than to the pretraining set.

We use a standard transformer encoder/decoder to train a machine translation model on IWSLT14 De-En (Cettolo et al., 2014). We follow the preprocessing and model architecture recommended by Fairseq.[1] The training set has 160,239 translation pairs, the validation set has 7,283, and the test set has 6,750.

As for the model, both the encoder and decoder are transformers with 6 layers, 4 attention heads per layer, 512 embedding dimensions, and 1024 feedforward dimensions. This corresponds to 40M parameters. We train with 0.3 weight dropout and 0.1 label smoothing, using 4,096 tokens for each train step. We train with Adam with a learning rate of 5×10^{-4} and use an inverse square root learning rate scheduler with 4,000 warmup steps.

[1] `https://github.com/pytorch/fairseq/tree/master/examples/translation`

Model	Dataset	Eval Metric	Standard Training	Compatible Training
Transformer decoder	Majority-Class	Perplexity	1.8	1.8
GPT-2	Open WebText	Perplexity	18.3	17.1
Transformer encoder/decoder	IWSLT14 EnDe	BLEU	34.8	34.8

Table A.3: Fine-tuning for compatibility does not hurt heldout performance. The first two rows are language models and the evaluation metric is heldout perplexity; the last row is machine translation, for which the evaluation metric is BLEU.

When we fine-tune for compatibility, we again condition on the full context with probability 0.5. With the remaining probability, we drop out each source and target token independently at each attention head with probability $1 - 1/T$, where T is the sequence length (so the dropout probability varies for the source and target sequence). Although we drop out different tokens at each attention head of a layer, we make sure that the same tokens are dropped out at each layer. Our word dropout procedure ensures that our objective will be trained on small contexts since rationales for machine translation are typically very sparse. We fine-tune using Adam with a constant learning rate of 10^{-5} for 410,000 steps. The heldout BLEU scores of both models are equal; see Table A.3.

A.5 Experimental Details

A.5.1 Long-Range Agreement

Table A.4 contains the template we used for the set of experiments containing the analogies from Mikolov et al. (2013a). To avoid rationales containing partial antecedents, we only include examples where both words in the analogy correspond to single word-pieces using GPT-2's tokenizer. Since it only makes sense to rationalize correct predictions, we also only include the examples where GPT-2 correctly completes the analogy. In total, this results in 175 examples. Of these, we randomly sample 50 to perform exhaustive search, which we use to compute the approximation ratio of each method. Since we cannot run exhaustive search when the minimal sufficient rationale is too large, we use the 40 that converge with rationales of length 6 or less. We use 100 steps to approximate the path integral for the integrated gradients baseline (Sundararajan et al., 2017).

To confirm that the baseline performances are not being hindered by fine-tuning for compatibility, we re-run the experiment for each rationalization method on the pretrained GPT-2 Large, without any fine-tuning. The results are depicted in Table A.5. As expected, the baselines perform even worse when GPT-2 is not fine-tuned to form compatible distributions. We do not include comparisons to exhaustive rationales because it is computationally infeasible to run exhaustive search on incompatible models, since optimization takes much longer to converge.

A.5.2 Machine Translation

For the distractor experiment, we randomly concatenate 500 pairs of source and target sequences generated by our fine-tuned model on the test set. We evaluate rationales by counting how many times they "cross over" and contain words from the distractor sequence. We do not penalize rationales that include special tokens like the beginning of sentence or end of sentence tokens.

For the alignment experiment, we use a public corpus of annotated rationales.[2] Not every word in the dataset has an alignment, and some words have multiple alignments. Although the human annotations are on word-level alignments, our machine translation models are trained on subwords, so the rationales contain subwords in addition to full words. To make these comparable to the human annotations, we define the rationale of a full target word to contain the union of the subword rationales. Since each source word may also be a subword, we also take the union of source words in a rationale. To calculate top-1 accuracy, we define the rationale for a full word to be accurate if the rationales for any of the subwords in the rationale contain any source subwords that are in the annotated alignment.

The alignment dataset contains both "sure" and "possible" alignments. These are used to differentiate between different errors when calculating the alignment error rate (Och & Ney, 2000). For the other metrics, we include both kinds of alignments as part of the annotated alignments.

For both machine translation experiments, we use 50 steps to approximate the path integral for the integrated gradients baseline (Sundararajan et al., 2017).

[2] https://www-i6.informatik.rwth-aachen.de/goldAlignment/

Relationship	Example
Capital countries	When my flight landed in **Greece**, I converted my currency and slowly fell asleep. (I had a terrifying dream about my grandmother, but that's a story for another time). I was staying in the capital, **Athens**
Currency	As soon as I arrived in **Japan**, I checked into my hotel and took a long nap. (I had finally finished the book I was reading and it was amazing). I had to figure out the exchange rate to the local currency, which is apparently called the **yen**
City in state	As soon as I arrived in **Florida**, I checked into my hotel and watched a movie before falling asleep. (I had a great call with my husband, although I wish it were longer). I was staying in my favorite city, **Miami**
Family	I initially invited my **uncles**, who gladly accepted my invitation. (My favorite song just came on, so I was able to relax). When I learned that women were allowed, I went ahead and also invited my **aunts**
Opposite	I thought it was **pleasant**. (Just then an ad came on the TV, but that's irrelevant). It was the opposite of that: it was **unpleasant**
Comparative	I knew it was **tall**, but that's before I saw it in person. (Just then I thought about my ex-wife, but I had to stop thinking about her). When I did end up seeing it in person, it was even **taller**
Superlative	I thought it would be the **smallest** thing I'd ever encounter. (I tried to ignore my phone vibrating in my pocket). But when I did end up encountering it, it turned out it wasn't so **small**
Present participle	Every other day, it started **working** in the morning. (I tried to remember the name of the woman at the bar). But today, it did not **work**
Nationality adjective	I had never been friends with any **French** people before. (The funniest thing happened to me the other day, but that's a story for another time). In fact, I had never even been to **France**
Past tense	Although I **listened** yesterday, I had a million things to do today. (I suddenly felt a pinched nerve, so I made a mental note to get that checked out). So today I wouldn't have time to do any more **listen**
Plural	I really wanted to buy the **computer**, more than I ever wanted to buy anything before. (I was also behind on my homework, but that's another story). So I went to the store and asked if they had any **computers**
Plural verbs	I can usually **sing** by myself. (I was so behind on work but I tried to distract myself). Although it's so much better when someone else also **sings**

Table A.4: Template using analogies from Mikolov et al. (2013a).

	Length	Ante	No D
Gradient norms	24.8	1.0	0.08
Gradient x embedding	41.1	0.99	0.00
Integrated gradients	34.9	1.0	0.00
Attention rollout	38.4	1.0	0.05
Last attention layer	20.1	0.99	0.03
All attention layers	19.5	1.0	0.02
Greedy	13.1	1.0	0.30

Table A.5: The performance of each rationalization method on the templated version of the analogies dataset (Mikolov et al., 2013a) when we don't fine-tune for compatibility. As expected, fine-tuning for compatibility (Table 2.1) improves performance across the board.

A.5.3 Annotated Lambada

We work with volunteers to annotate Lambada (Paperno et al., 2016). Each example requires two annotators: a selector, and a predictor. A selector's goal is to choose the most important words for predicting the final word of a passage, known as the target word. Predictors will only be seeing the words chosen by a selector, and their goal is to predict the final word of the passage.

The selector first takes a passage and ranks 15 words. The top-ranked word always needs to be the word before the final word of the passage. They cannot select the final word of the passage. Each of their selections needs to be a complete word. They cannot select the same word twice, and they need to use all 15 spots. They know that a predictor will be predicting words, one-at-a-time, using the order they create.

When a selector is finished ranking the top 15 words, a predictor begins by seeing the top ranked word. They use this to predict the last word. Words are revealed one-at-a-time in the order chosen by the selector. Selectors can see how much space is between the words that have been revealed. Selectors are not told if they predicted a word correctly; the goal of the exercise is to capture the predictor's true predictions, so if they knew that previous guesses were incorrect, they may use this information to guess a new word at every step. If a predictor is not able to guess the target word at the end of the exercise, we re-assign the example to another predictor.

In total, we annotate 107 examples, and use all of them for the rationalization experiment. For each example, we define a human's rationale to be all the words that were revealed by the selector

Target word: again

I wanted to make sure you were still comfortable with the arrangements. I can always do something different."

You're too good to me, Max. But I'm fine. I promise. I'm going to be okay this time. I've **learned from** my past **mistakes**. I **don't want** to **make them** _____

Target word: ring

I joined Mark, Tony, and his son back in the crowd as the event started. I wanted to catch Yegor's match before I got ready for my debut. He was **wrestling** first; there were only four matches on the card.
Minutes after I got seated the lights dimmed. That cold music filled with horns played. It was Yegor's time to come to **the** _____

Target word: fire

"We aren't out of danger yet," Horatius said. He headed northwest without thinking about it. It just seemed the right way to go.
Chloe squirmed as she became more alert. "We have to go back. Now. We can't leave my house **burning** with my family in there."
Horatius didn't know what to do about **the** _____

Target word: contract

'Your **services** will be **required** for the period of **three** months.'

I press my lips together. I was very drunk last night, but I am **sure** he **said** one month. 'Can I speak to him?'

'Of course.' He picks up the phone and speed dials his client's number. 'Mr. Barrington, Miss Bloom **would like** to have a **word** about **the length of the** _____

Figure A.1: Sample rationales from our annotated Lambada dataset. Highlighted text corresponds to greedy rationales, and **bolded text** corresponds to human annotated rationales.

before the predictor first predicted the true target word or a synonym of it. The average rationale length is 6.0.

To compare human rationales to those found by various methods, we first tokenize the text with GPT-2's tokenizer, and convert an annotated rationale to its set of corresponding subwords. Each method's rationale is also a set of subwords. We use set-comparison metrics like intersection over union (IOU) and F1 to compare the similarity of rationales. We use 100 steps to approximate the path integral for the integrated gradients baseline (Sundararajan et al., 2017).

A.6 Qualitative Examples

Figure A.1 contains examples of rationales on our annotated Lambada dataset.

Appendix B: Text-Based Ideal Points

B.1 Algorithm

We present the full procedure for training the text-based ideal point model (TBIP) in Algorithm 2. We make a final modification to the model in Equation 3.3. If some political authors are more verbose than others (i.e. use more words per document), the learned ideal points may reflect verbosity rather than a political preference. Thus, we multiply the expected word count by a term that captures the author's verbosity compared to all authors. Specifically, if n_s is the average word count over documents for author s, we set a weight:

$$w_s = \frac{n_s}{\frac{1}{S}\sum_{s'} n_{s'}}, \tag{B.1}$$

for S the number of authors. We then multiply the rate in Equation 3.3 by w_{a_d}. Empirically, we find this modification does not make much of a difference for the correlation results, but it helps us interpret the ideal points for the qualitative analysis.

B.2 Data and inference settings

Senator speeches We remove senators who made less than 24 speeches. To lessen non-ideological correlations in the speaking patterns of senators from the same state, we remove cities and states in addition to stopwords and procedural terms. We include all unigrams, bigrams, and trigrams that appear in at least 0.1% of documents and at most 30%. To ensure that the inferences are not influenced by procedural terms used by a small number of senators with special appointments, we only include phrases that are spoken by 10 or more senators. This preprocessing leaves us with 19,009 documents from 99 senators, along with 14,503 terms in the vocabulary.

Algorithm 2: The text-based ideal point model (TBIP)

Input: Word counts $\boldsymbol{y}$, authors $\boldsymbol{a}$, and number of topics K (D documents and V words)

Output: Document intensities $\hat{\boldsymbol{\theta}}$, neutral topics $\hat{\boldsymbol{\beta}}$, ideological topic offsets $\hat{\boldsymbol{\eta}}$, ideal points $\hat{\boldsymbol{x}}$

Pretrain: Hierarchical Poisson factorization (Gopalan et al., 2015) to obtain initial estimates $\hat{\boldsymbol{\theta}}$ and $\hat{\boldsymbol{\beta}}$

Initialize: Variational parameters $\boldsymbol{\sigma}_\theta^2, \boldsymbol{\sigma}_\beta^2, \boldsymbol{\mu}_\eta, \boldsymbol{\sigma}_\eta^2, \boldsymbol{\mu}_x, \boldsymbol{\sigma}_x^2$ randomly, $\boldsymbol{\mu}_\theta = \log(\hat{\boldsymbol{\theta}})$, $\boldsymbol{\mu}_\beta = \log(\hat{\boldsymbol{\beta}})$

Compute weights $\boldsymbol{w}$ as in Equation B.1

while the *evidence lower bound* (ELBO) has not converged **do**

- **sample** a document index $d \in \{1, 2, \ldots, D\}$
- **sample** $\boldsymbol{z}_\theta, \boldsymbol{z}_\beta, \boldsymbol{z}_\eta, \boldsymbol{z}_x \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ *Sample noise distribution
- Set $\tilde{\boldsymbol{\theta}} = \exp(\boldsymbol{z}_\theta \odot \boldsymbol{\sigma}_\theta + \boldsymbol{\mu}_\theta)$ and $\tilde{\boldsymbol{\beta}} = \exp(\boldsymbol{z}_\beta \odot \boldsymbol{\sigma}_\beta + \boldsymbol{\mu}_\beta)$ *Reparameterize
- Set $\tilde{\boldsymbol{\eta}} = \boldsymbol{z}_\eta \odot \boldsymbol{\sigma}_\eta + \boldsymbol{\mu}_\eta$ and $\tilde{\boldsymbol{x}} = \boldsymbol{z}_x \odot \boldsymbol{\sigma}_x + \boldsymbol{\mu}_x$ *Reparameterize
- **for** $v \in \{1, \ldots, V\}$ **do**
  - Set $\lambda_{dv} = \left(\sum_k \tilde{\theta}_{dk} \tilde{\beta}_{kv} \exp(\tilde{\eta}_{kv} \tilde{x}_{a_d})\right) * w_{a_d}$
  - Compute $\log p(y_{dv} | \tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{x}}) = \log \text{Pois}(y_{dv} | \lambda_{dv})$ *Log-likelihood term
- **end**
- Set $\log p(\boldsymbol{y}_d | \tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{x}}) = \sum_v \log p(y_{dv} | \tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{x}})$ *Sum over words
- Compute $\log p(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{x}})$ and $\log q(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{x}})$ *Prior and entropy terms
- Set $\text{ELBO} = \log p(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{x}}) + N * \log p(\boldsymbol{y}_d | \tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{x}}) - \log q(\tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\beta}}, \tilde{\boldsymbol{\eta}}, \tilde{\boldsymbol{x}})$
- Compute gradients $\nabla_\phi \text{ELBO}$ using automatic differentiation
- Update parameters ϕ

end

return approximate posterior means $\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\eta}}, \hat{\boldsymbol{x}}$

	Speeches 111		Speeches 112		Speeches 113		Tweets 114	
	Corr.	SRC	Corr.	SRC	Corr.	SRC	Corr.	SRC
WORDFISH	0.52	0.49	0.51	0.51	0.71	0.65	0.79	0.74
WORDSHOAL	0.62	0.66	0.58	0.51	0.46	0.46	—	—
TBIP	**0.82**	**0.77**	**0.85**	**0.85**	**0.89**	**0.86**	**0.94**	**0.88**

Table B.1: The TBIP learns ideal points most similar to DW-NOMINATE vote ideal points for U.S. senator speeches and tweets. It learns closer ideal points than WORDFISH and WORDSHOAL in terms of both correlation (Corr.) and Spearman's rank correlation (SRC). The numbers in the column titles refer to the Senate session of the corpus. WORDSHOAL cannot be applied to tweets because there are no debate labels.

To train the TBIP, we perform stochastic gradient ascent using Adam (Kingma & Ba, 2015), with a mini-batch size of 512. To curtail extreme word count values from long speeches, we take the natural logarithm of the counts matrix before performing inference (appropriately adding 1 and rounding so that a word count of 1 is transformed to still be 1). We use a single Monte Carlo sample to approximate the gradient of each batch. We assume 50 latent topics and posit the following prior distributions: $\theta_{dk}, \beta_{kv} \sim \text{Gamma}(0.3, 0.3)$, $\eta_{kv}, x_s \sim \mathcal{N}(0, 1)$.

We train the vote ideal point model by removing all votes that are not cast as "yea" or "nay" and performing mean-field variational inference with Gaussian variational distributions. Since each variational family is Gaussian, we approximate gradients using the reparameterization trick (Rezende et al., 2014; Kingma & Ba, 2015).

For the comparisons against WORDFISH and WORDSHOAL, we preprocess speeches in the same way as Lauderdale & Herzog (2016a). We train each Senate session separately, thereby only including one timestep for WORDFISH. For this reason, our results on the U.S. Senate differ from those reported by Lauderdale & Herzog (2016a), who train a model jointly over all time periods. Additionally, we use variational inference with reparameterization gradients to train all methods. Specifically, we perform mean-field variational inference, positing Gaussian variational families on all real variables and lognormal variational families on all positive variables.

Senator tweets Our Senate tweet preprocessing is similar to the Senate speech preprocessing, although we now include all terms that appear in at least 0.05% of documents rather than 0.01%

to account for the shorter tweet lengths. We remove cities and states in addition to stopwords and the names of politicians. This preprocessing leaves us with 209,779 tweets. We use the same model and hyperparameters as for speeches, although we no longer take the natural logarithm of the counts matrix since individual tweets cannot have extreme word counts due to the character limit. We use a batch size of 1,024.

2020 Democratic candidates We scrape the Twitter feeds of 19 candidates, including all tweets between January 1, 2019 and February 27, 2020. We do not include Andrew Yang, Jay Inslee, and Marianne Williamson since it is difficult to define the political preferences of non-traditional or single-issue candidates. We follow the same preprocessing we used for the 114th Senate, except we include tokens that are used in more than 0.05% of documents rather than 0.1%. We remove phrases used by only one candidate, along with stopwords and candidate names. This preprocessing leaves us with 45,927 tweets for the 19 candidates. We use the same model and hyperparameters as for senator tweets.

B.3 Comparison to DW-Nominate

DW-NOMINATE (Poole, 2005) is a dynamic method for learning ideal points from votes. As opposed to the vote ideal point model in Equation 3.1, it analyzes votes across multiple Senate sessions. It also learns two latent dimensions per legislator. We also compare text ideal points to the first dimension of DW-Nominate, which corresponds to economic/redistributive preferences (Lewis et al., 2020). We use the fitted DW-NOMINATE ideal points available on Voteview (Lewis et al., 2020). The TBIP learns ideal points closer to DW-NOMINATE than WORDFISH and WORDSHOAL; see Table B.1.

In Section 5.4, we observed that Bernie Sanders' vote ideal point is somewhat moderate under the scalar ideal point model from Equation 3.1. It is worth noting that Sanders' vote ideal point is more extreme under DW-NOMINATE than under the scalar model: his DW-NOMINATE ideal point is the third-most extreme among Democrats. Since DW-NOMINATE uses two dimensions to

model each legislator's latent preferences, it can more flexibly model Sanders' voting deviations. Additionally, the dynamic nature of DW-NOMINATE may capture salient information from other Senate sessions. However, restricting the vote ideal point to be static and a scalar, like it is for the TBIP, results in the more moderate vote ideal point in Section 5.4.

Appendix C: CAREER: Transfer Learning for Labor Sequence Data

C.1 Econometric Baselines

In this section, we describe baseline occupation models that economists have used to model jobs and other discrete sequences.

Markov models and regression. A first-order Markov model assumes the job at each timestep depends on only the previous job (Hall, 1972; Poterba & Summers, 1986). Without covariates, a Markov model takes the form $p(y_t = j \,|\, \boldsymbol{y}_{t-1}) = p(y_t = j \,|\, y_{t-1})$. The optimal transition probabilities reflect the overall frequencies of individuals transitioning from occupation y_{t-1} to occupation j. In a second-order Markov model, the next job depends on the previous two.

A multinomial logistic regression can be used to incorporate covariates:

$$p(y_t = j \,|\, \boldsymbol{y}_{t-1}, \boldsymbol{x}_t) \propto \exp\left\{\beta_j^{(0)} + \beta_j^{(1)} \cdot y_{t-1} + \textstyle\sum_c \beta_j^{(c)} \cdot x_{tc}\right\}, \tag{C.1}$$

where $\beta_j^{(0)}$ is an occupation-specific intercept and y_{t-1} and x_{tc} denote J- and N_c-dimensional indicator vectors, respectively. Equation C.1 depends on history only through the most recent job, although the covariates can also include hand-crafted summary statistics about the past, such as the duration of the most recent job (McCall, 1990). This model is fit by maximizing the likelihood with gradient-based methods.

Bag-of-jobs. A weakness of the first-order Markov model is that it only uses the most recent job to make predictions. However, one's working history beyond the last job may inform future transitions (Blau & Riphahn, 1999; Neal, 1999).

Another baseline we consider is a *bag-of-jobs* model, inspired by SHOPPER, a probabilistic

	Model	Overall	Consecutive Repeat	Non-Consecutive Repeat	New Job
No covariates	First-order Markov	5.73	1.25	258.75	479.65
	Second-order Markov	5.66	1.25	154.35	471.10
	Bag-of-Jobs	5.53	1.25	59.07	458.53
	Transformer	5.38	1.23	36.01	429.32
Covariates	Regression	5.34	1.22	130.25	414.76
	Bag-of-Jobs	5.26	1.23	56.58	410.62
	CAREER	**5.05**	**1.21**	**33.72**	**358.80**

Table C.1: Held-out perplexity on the large resumes dataset (lower is better).

model of consumer choice (Ruiz et al., 2020). Unlike the Markov and regression models, the bag-of-jobs model conditions on every job in an individual's history. It does so by learning a low-dimensional representation of an individual's history. This model learns a unique embedding for each occupation, similar to a word embedding (Bengio et al., 2003; Mikolov et al., 2013b); unlike CAREER, which learns complicated nonlinear interactions between jobs in a history, the bag-of-jobs model combines jobs into a single representation by averaging their embeddings.

The bag-of-jobs model assumes that job transitions depend on two terms: a term that captures the effect of the most recent job, and a term that captures the effect of all prior jobs. Accordingly, the model learns two types of representations: an embedding $\alpha_j \in \mathbb{R}^D$ of the most recent job j, and an embedding $\rho_{j'} \in \mathbb{R}^D$ for prior jobs j'. To combine the representations for all prior jobs into a single term, the model averages embeddings:

$$p(y_t = j \,|\, \boldsymbol{y}_{t-1}) \propto \exp\left\{\beta_j^{(1)} \cdot \alpha_{y_{t-1}} + \beta_j^{(2)} \cdot \left(\tfrac{1}{t-2} \textstyle\sum_{t'=1}^{t-2} \rho_{y_{t'}}\right)\right\}. \tag{C.2}$$

Covariates can be added to the model analogously; for a single covariate, its most recent value is embedded and summed with the average embeddings for its prior values. All parameters are estimated by maximizing the likelihood in Equation C.2 with SGD.

C.2 Resume Predictions

Although our focus is on modeling survey datasets, we also compare CAREER to several econometric baselines for predicting job sequences in resumes. We consider a series of models without covariates: a first- and second-order Markov model, a bag-of-jobs model (Equation C.2), and a transformer with the same architecture as CAREER except without covariates. We also compare to econometric models that use covariates: a second-order linear regression with covariates and hand-constructed features (such as how long an individual has worked in their current job), and a bag-of-jobs model with covariates (Section C.9 has more details).

We randomly divide the resumes dataset into a training set of 23.6 million sequences, and a validation and test set of 23 thousand sequences each. Table C.1 compares the test-set predictive performance of all models. CAREER is the best at predicting held-out sequences. To understand the types of transitions contributing to CAREER's predictive advantage, we decompose predictions into three categories: consecutive repeats (when the next job is the same as the previous year's), non-consecutive repeats (when the next job is different from the previous year's, but is the same as one of the prior jobs in the career), and new jobs. CAREER has a clear advantage over the baselines in all three categories, but the biggest improvement comes when predicting jobs that have been repeated non-consecutively. The transformer model is at an advantage over the Markov models for these kinds of predictions because it is able to condition on an individual's entire working history, while a Markov model is constrained to use only the most recent job (or two). The bag-of-jobs model, which can condition on all jobs in a worker's history but cannot learn complex interactions between them, outperforms the Markov models but still falls short of CAREER, which can recognize and represent complex career trajectories. In Section C.3, we demonstrate that CAREER is well-equipped at forecasting future trajectories as well.

	Overall	2015	2016	2017
Regression	20.71	7.78	27.97	40.85
Bag-of-Jobs	19.45	7.57	25.63	37.93
CAREER	**17.37**	**7.07**	**23.06**	**32.11**

Table C.2: Forecasting perplexity (lower is better) for unseen years in the large resumes dataset. Each model is trained on sequences before 2015 and makes forecasts three years into the future. The "overall" column averages perplexities across all three forecasted years.

C.3 Forecasting Resumes

We also perform the forecasting experiment on the large dataset of resumes. Each model is trained on resumes before 2015. To predict occupations for individuals after 2015, a model samples 1,000 trajectories for each individual, and averages probabilities to form a single prediction for each year. For more experimental details, see Section C.9.

Table C.2 depicts the forecasting results for the resumes dataset. Each fitted model is used to forecast occupation probabilities for three years into the future. CAREER makes the best forecasts, both overall and for each individual year.

C.4 Qualitative Analysis

Representation similarity. To demonstrate the quality of the learned representations, we use CAREER's fine-tuned representations on NLSY97 to find pairs of individuals with the most similar career trajectories. Specifically, we compute CAREEER's representation $h_t(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t)$ for each individual in NLSY97 who has worked for four years. We then measure the similarity between all pairs by computing the cosine similarity between representations. In order to depict meaningful matches, we only consider pairs of individuals with no overlapping jobs in their histories (otherwise the model would find individuals with the exact same career trajectories). Figure C.1 depicts the career histories with the most similar CAREER representations. Although none of these pairs have overlapping jobs, the model learns representations that can identify similar careers.

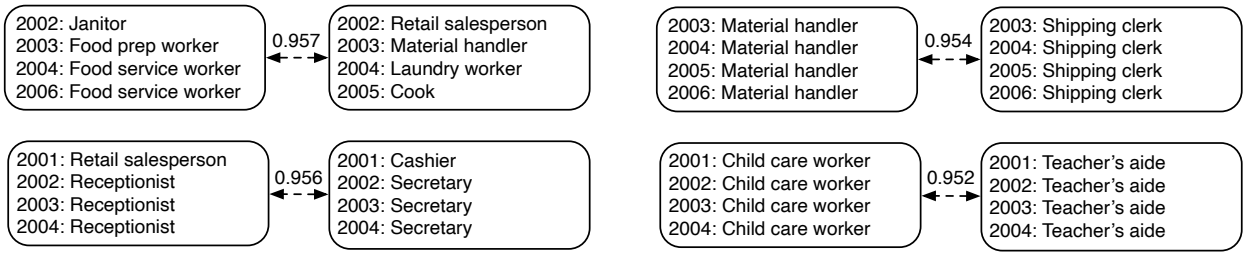


Figure C.1: The work experiences with the most similar CAREER representations (measured with cosine similarity) for individuals with no overlapping jobs in NLSY97.

C.5 Transformer Details

In this section, we expand on the simplified description of transformers in Section 4.2.3 and describe CAREER in full detail. Recall that the model estimates representations in L layers, $h_t^{(1)}(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t), \ldots, h_t^{(L)}(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t)$, with each representation $h_t^{(\ell)} \in \mathbb{R}^D$. The final representation $h_t^{(L)}(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t)$ is used to represent careers. We drop the explicit dependence on $\boldsymbol{y}_{t-1}$ and $\boldsymbol{x}_t$, and instead denote each representation as $h_t^{(\ell)}$.

The first transformer layer combines the previous occupation, the most recent covariates, and the position of the occupation in the career. It first embeds each of these variables in D-dimensional space. Define an embedding function for occupations, $e_y : [J] \to \mathbb{R}^D$. Additionally, define a separate embedding function for each covariate, $\{e_c\}_{c=1}^C$, with each $e_c : [N_c] \to \mathbb{R}^D$. Finally, define $e_t : [T] \to \mathbb{R}^D$ to embed the position of the sequence, where T denotes the number of possible sequence lengths. The first-layer representation $h_t^{(1)}$ sums these embeddings:

$$h_t^{(1)} = e_y(y_{t-1}) + \textstyle\sum_c e_c(x_{tc}) + e_t(t). \tag{C.3}$$

The occupation- and covariate-specific embeddings, e_y and $\{e_c\}$, are model parameters; the positional embeddings, e_t, are set in advance to follow a sinusoidal pattern (Vaswani et al., 2017). While these embeddings could also be parameterized, in practice the performance is similar, and using sinusoidal embeddings allows the model to generalize to career sequence lengths unseen in the training data.

At each subsequent layer, the transformer combines the representations of all occupations in a history. It combines representations by performing *multi-headed attention*, which is similar to the process described in Section 4.2.3 albeit with multiple attention weights per layer.

Specifically, it uses A specific attention weights, or *heads*, per layer. The number of heads A should be less than the representation dimension D. (Using $A = 1$ attention head reduces to the process described in Equations 4.5 and 4.6.) The representation dimension D should be divisible by A; denote $K = D/A$. First, A different sets of attention weights are computed:

$$\begin{aligned} z_{a,t,t'}^{(\ell)} &= \left(h_t^{(\ell)}\right)^\top W_a^{(\ell)} h_{t'}^{(\ell)} \quad \text{for } t' \leq t \\ \pi_{a,t,t'} &= \frac{\exp\{z_{a,t,t'}\}}{\sum_k \exp\{z_{a,t,k}\}}, \end{aligned} \tag{C.4}$$

where $W_a^{(\ell)} \in \mathbb{R}^{D \times D}$ is a model parameter, specific to attention head a and layer l.[1] Each attention head forms a convex combination with all previous representations; to differentiate between attention heads, each representation is transformed by a linear transformation $V_a^{(\ell)} \in \mathbb{R}^{K \times D}$ unique to an attention head, forming $b_{a,t}^{(\ell)} \in \mathbb{R}^K$:

$$b_{a,t}^{(\ell)} = \textstyle\sum_{t'=1}^{t} \pi_{a,t,t'}^{(\ell)} \left(V_a^{(\ell)} h_{t'}^{(\ell)}\right). \tag{C.5}$$

All attention heads are combined into a single representation by concatenating them into a single vector $g_t^{(\ell)} \in \mathbb{R}^D$:

$$g_t^{(\ell)} = \left(b_{1,t}^{(\ell)}, b_{2,t}^{(\ell)}, \ldots, b_{A,t}^{(\ell)}\right). \tag{C.6}$$

To complete the multi-head attention step and form the intermediate representation $\tilde{h}_t^{(\ell)}$, the concatenated representations $g_t^{(\ell)}$ undergo a linear transformation and are summed with the pre-

[1] For computational reasons, $W_a^{(\ell)}$ is decomposed into two matrices and scaled by a constant, $W_a^{(\ell)} = \frac{Q_a^{(\ell)}\left(K_a^{(\ell)}\right)^\top}{\sqrt{K}}$, with $Q_a^{(\ell)}, K_a^{(\ell)} \in \mathbb{R}^{D \times K}$.

attention representation $h_t^{(\ell)}$:

$$\tilde{h}_t^{(\ell)} = h_t^{(\ell)} + M^{(\ell)} g_t^{(\ell)}, \tag{C.7}$$

with $M^{(\ell)} \in \mathbb{R}^{D \times D}$.

The intermediate representations $\tilde{h}_t^{(\ell)} \in \mathbb{R}^D$ combine the representation at timestep t with those preceding timestep t. Each layer of the transformer concludes by taking a non-linear transformation of the intermediate representations. This non-linear transformation does not depend on any previous representation; it only transforms $\tilde{h}_t^{(\ell)}$. Specifically, $\tilde{h}_t^{(\ell)}$ is passed through a neural network:

$$h_t^{(\ell+1)} = \tilde{h}_t^{(\ell)} + \text{FFN}^{(\ell)}\left(\tilde{h}_t^{(\ell)}\right), \tag{C.8}$$

where $\text{FFN}^{(\ell)}$ denotes a two-layer feedforward neural network with N hidden units, with $\text{FFN}^{(\ell)} : \mathbb{R}^D \to \mathbb{R}^D$.

We repeat the multi-head attention and feedforward neural network updates above for L layers, using parameters unique to each layer. We represent careers with the last-layer representation, $h_t(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t) = h_t^{(L)}(\boldsymbol{y}_{t-1}, \boldsymbol{x}_t)$.

For our experiments, we use model specifications similar to the generative pretrained transformer (GPT) architecture (Radford et al., 2018). In particular, we use $L = 12$ layers, a representation dimension of $D = 192$, $A = 3$ attention heads, and $N = 768$ hidden units and the GELU nonlinearity (Hendrycks & Gimpel, 2016) for all feedforward neural networks. In total, this results in 5.6 million parameters. This model includes a few extra modifications to improve training: we use 0.1 dropout (Srivastava et al., 2014) for the feedforward neural network weights, and 0.1 dropout for the attention weights. Finally, we use layer normalization (Ba et al., 2016) before the updates in Equation C.4, after the update in Equation C.7, and after the final layer's neural network update in Equation C.8.

General	Number of individuals	23,731,674
	Number of tokens	245,439,865
	Median year	2007
Geography	Percent Northeast	17.6
	Percent Northcentral	20.7
	Percent South	39.9
	Percent West	19.4
	Percent without location	2.4
Education	Percent high school diploma	7.2
	Percent associate degree	8.6
	Percent bachelor degree	23.1
	Percent graduate degree	4.5
	Percent empty	52.8
Broad Occupation Groups	Percent managerial/professional specialty	38.4
	Percent technical/sales/administrative support	34.2
	Percent service	12.0
	Percent precision production/craft/repair	7.9
	Percent operator/fabricator/laborer	7.2

Table C.3: Exploratory data analysis of the resume dataset used for pretraining CAREER.

C.6 Exploratory Data Analysis

Table C.3 depicts summary statistics of the resume dataset provided by Zippia that is used for pretraining CAREER. Table C.4 compares this resume dataset with the longitudinal survey datasets of interest.

	Resumes	NLSY79	NLSY97	PSID
Number of individuals	24 million	12 thousand	9 thousand	12 thousand
Unemployed/out-of-labor-force/student available	No	Yes	Yes	Yes
Median year	2007	1991	2007	2011
Percent manual laborers	7%	17%	13%	12%
Percent college graduates	56%	23%	29%	28%
Demographic covariates available	No	Yes	Yes	Yes

Table C.4: Comparing the resume dataset used for pretraining with the three longitudinal survey datasets of interest.

	PSID	NLSY79	NLSY97
Markov regression (two-stage)	15.60 ± 0.03	13.30 ± 0.02	15.47 ± 0.00
NEMO (two-stage)	15.23 ± 0.08	12.37 ± 0.04	15.13 ± 0.03
Job rep. learning (two-stage)	15.98 ± 0.06	13.97 ± 0.03	15.43 ± 0.01
Job2Vec (two-stage)	15.80 ± 0.04	13.91 ± 0.01	15.31 ± 0.02
Bag-of-jobs (two-stage)	15.40 ± 0.05	12.68 ± 0.01	15.11 ± 0.02
CAREER	**13.88 ± 0.01**	**11.32 ± 0.00**	**14.15 ± 0.03**

Table C.5: Perplexity of economic baselines when they are modified to make predictions in two stages.

C.7 One-Stage vs Two-Stage Prediction

Table C.5 compares the predictive performance of occupation models when they are modified to make predictions in two stages, following Equations 4.1 to 4.3. Incorporating two-stage prediction improves the performance of these models compared to Figure 4.2(a); however, CAREER still makes the best predictions on all survey datasets.

C.8 Data Preprocessing

In this section, we go over the data preprocessing steps we took for each dataset.

Resumes. We were given access to a large dataset of resumes of American workers by Zippia, a career planning company. This dataset coded each occupation into one of 1,073 O*NET 2010 Standard Occupational Classification (SOC) categories based on the provided job titles and descriptions in resumes. We dropped all examples with missing SOC codes.

Each resume in the dataset we were given contained covariates that had been imputed based off other data in the resume. We considered three covariates: year, most recent educational degree, and location. Education degrees had been encoded into one of eight categories: high school diploma, associate, bachelors, masters, doctorate, certificate, license, and diploma. Location had been encoded into one of 50 states plus Puerto Rico, Washington D.C., and unknown, for when location could not be imputed. Some covariates also had missing entries. When an occupation's

year was missing, we had to drop it from the dataset, because we could not position it in an individual's career. Whenever another covariate was missing, we replaced it with a special "missing" token. All personally identifiable information had been removed from the dataset.

We transformed each resume in the dataset into a sequence of occupations. We included an entry for each year starting from the first year an individual worked to their last year. We included a special "beginning of sequence" token to indicate when each individual's sequence started. For each year between an individual's first and last year, we added the occupation they worked in during that year. If an individual worked in multiple occupations in a year, we took the one where the individual spent more time in that year; if they were both the same amount of time in the particular year, we broke ties by adding the occupation that had started earlier in the career. For the experiments predicting future jobs directly on resumes, we added a "no-observed-occupation" token for years where the resume did not list any occupations (we dropped this token when pretraining). Each occupation was associated with the individual's most recent educational degree, which we treated as a dynamic covariate. The year an occupation took place was also considered a dynamic categorical covariate. We treated location as static. In total, this preprocessing left us with a dataset of 23.7 million resumes, and 245 million individual occupations.

In order to transfer representations, we had to slightly modify the resumes dataset for pretraining to encode occupations and covariates into a format compatible with the survey datasets. The survey datasets we used were encoded with the "occ1990dd" occupation code (Autor & Dorn, 2013) rather than with O*NET's SOC codes, so we converted the SOC codes to occ1990dd codes using a crosswalk posted online by Destin Royer. Even after we manually added a few missing entries to the crosswalks, there were some SOC codes that did not have corresponding occ1990dd's. We gave these tokens special codes that were not used when fine-tuning on the survey datasets (because they did not correspond to occ1990dd occupations). When an individual did not work for a given year, the survey datasets differentiated between three possible states: unemployed, out-of-labor-force, and in-school. The resumes dataset did not have these categories. Thus, we initialized parameters for these three new occupational states randomly. Additionally, we did not include the

"no-observed-occupation" token when pretraining, and instead dropped missing years from the sequence. Since we did not use gender and race/ethnicity covariates when pretraining, we also initialized these covariate-specific parameters randomly for fine-tuning. Because we used a version of the survey datasets that encoded each individual's location as a geographic region rather than as a state, we converted each state in the resumes data to be in one of four regions for pretraining: northeast, northcentral, south, or west. We also added a fifth "other" region for Puerto Rico and for when a state was missing in the original dataset. We also converted educational degrees to levels of experience: we converted associate's degree to represent some college experience and bachelor's degree to represent four-year college experience; we combined masters and doctorate to represent a single "graduate degree" category; and we left the other categories as they were.

NLSY79. The National Longitudinal Survey of Youth 1979 (NLSY79) is a survey following individuals born in the United States between 1957-1964. The survey included individuals who were between 14 and 22 years old when they began collecting data in 1979; they interviewed individuals annually until 1994, and biennially thereafter.

Each individual in the survey is associated with an ID, allowing us to track their careers over time. We converted occupations, which were initially encoded as OCC codes, into "occ1990dd" codes using a crosswalk (Autor & Dorn, 2013). We use a version of the survey that has entries up to 2014. Unlike the resumes dataset, NLSY79 includes three states corresponding to individuals who are not currently employed: unemployed, out-of-labor-force, and in-school. We include special tokens for these states in our sequences. We drop examples with missing occupation states. We also drop sequences for which the individual is out of the labor force for their whole careers.

We use the following covariates: years, educational experience, location, race/ethnicity, and gender. We drop individuals with less than 9 years of education experience. We convert years of educational experience into discrete categories: no high school degree, high school degree, some college, college, and graduate degree. We convert geographic location to one of four regions: northeast, northcentral, south, and west. We treat location as a static variable, using each individ-

ual's first location. We use the following race/ethnicities: white, African American, Asian, Latino, Native American, and other. We treat year and education as dynamic covariates whose values can change over time, and we consider the other covariates as static. This preprocessing leaves us with a dataset consisting of 12,270 individuals and 239,545 total observations.

NLSY97. The National Longitudinal Survey of Youth 1997 (NLSY97) is a survey following individuals who were between 12 and 17 when the survey began in 1997. Individuals were interviewed annually until 2011, and biennially thereafter.

Our preprocessing of this dataset is similar to that of NLSY79. We convert occupations from OCC codes into "occ1990dd" codes. We use a version of the survey that follows individuals up to 2019. We include tokens for unemployed, out-of-labor-force, and in-school occupational states. We only consider individuals who are over 18 and drop military-related occupations. We use the same covariates as NLSY79. We use the following race/ethnicities: white, African-aAmerican, Latino, and other/unknown. We convert years of educational experience into discrete categories: no high school degree, high school degree, some college degree, college degree, graduate degree, and a special token when the education status isn't known. We use the same regions as NLSY79. We drop sequences for which the individual is out of the labor force for their whole careers. This preprocessing leaves us with 8,770 individuals and 114,141 total observations.

PSID. The Panel Study of Income Dynamics (PSID) is a longitudinal panel survey following a sample of families in the United States. It was collected annually between 1968 and 1997, and biennially afterwards.

The dataset tracks families over time, but it only includes occupation information for the household head and their spouse, so we only include these observations. Occupations are encoded with OCC codes, which we convert to "occ1990dd" using a crosswalk (Autor & Dorn, 2013). Like the NLSY surveys, PSID also includes three states corresponding to individuals who are not currently employed: unemployed, out-of-labor-force, and in-school. We include special tokens for these states in our sequences. We drop other examples with missing or invalid occupation codes. We

also drop sequences for which the individual is out of the labor force for their whole careers.

We consider five covariates: year, education, location, gender, and race. We include observations for individuals who were added to the dataset after 1995 and include observations up to 2019. We exclude observations for individuals with less than 9 years of education experience. We convert years of education to discrete states: no high school, high school diploma, some college, college, and graduate degree. We convert geographic location to one of four regions: northeast, northcentral, south, and west. We treat location as a static variable, using each individual's first location. We use the following races: white, Black, and other. We treat year and education as dynamic covariates whose values can change over time, and we consider the other covariates as static. This preprocessing leaves us with 12,338 individuals and 62,665 total observations.

C.9 Experimental Details

Baselines. We consider a first-order Markov model and a second-order Markov model (both without covariates) as baselines. These models are estimated by averaging observed transition counts. We smooth the first-order Markov model by taking a weighted average between the empirical transitions in the training set and the empirical distribution of individual jobs. We perform this smoothing to account for the fact that some feasible transitions may never occur in the training set due to the high-dimensionality of feasible transitions. We assign 0.99 weight to the empirical distributions of transitions and 0.01 to the empirical distribution of individual jobs. We smooth the second-order model by assigning 0.5 weight to the empirical second-order transitions and 0.5 weight to the smoothed first-order Markov model.

When we add covariates to the Markov linear baseline, we also include manually constructed features about history to improve its performance. In total, we include the following categorical variables: the most recent job, the prior job, the year, a dummy indicating whether there has been more than one year since the most recent observed job, the education status, a dummy indicating whether the education status has changed, and state (for the experiments on NLSY79 and PSID, we also include an individual's gender and race/ethnicity). We also add additive effects for the

following continuous variables: the number of years an individual has been in the current job and the total number of years for which an individual has been in the dataset. In addition, we include an intercept term.

For the bag-of-jobs model, we vary the representation dimension D between 256-2048, and find that the predictive performance is not sensitive to the representation dimension, so we use $D = 1024$ for all experiments. For the LSTM model, we use 3 layers with 436 embedding dimensions so that the model size is comparable to the transformer baseline: the LSTM has 5.8 million parameters, the same number as the transformer.

We also compare to NEMO (Li et al., 2017), an LSTM-based method developed for modeling job sequences in resumes. We adapted NEMO to model survey data. In its original setting, NEMO took as input static covariates (such as individual skill) and used these to predict both an individual's next job title and their company. Survey datasets differ from this original setting in a few ways: covariates are time-varying, important covariates for predicting jobs on resumes (like skill) are missing, and an individual's company name is unavailable. Therefore, we made several modifications to NEMO. We incorporated the available covariates from survey datasets by embedding them and adding them to the job embeddings passed into the LSTM, similar to the method CAREER uses to incorporate covariates. We removed the company-prediction objective, and instead only used the model to predict an individual's job in the next timestep. We considered two sizes of NEMO: an architecture using the same number of parameters as CAREER, and the smaller architecture proposed in the original paper. We found the smaller architecture performed better on the survey datasets, so we used this for the experiments. This model contains 2 decoder layers and a hidden dimension of 200.

We compare to two additional baselines developed in the data mining literature: job representation learning (Dave et al., 2018) and Job2Vec (Zhang et al., 2020). These methods require resume-specific features such as skills and textual descriptions of jobs and employers, which are not available for the economic longitudinal survey datasets we model. Thus, we adapt these baselines to be suitable for modeling economic survey data. Job representation learning (Dave et al.,

2018) is based on developing two graphs, one for job transitions and one for skill transitions. Since worker skills are not available for longitudinal survey data, we adapt the model to only use job transitions by only including the terms in the objective that depend on job transitions. We make a few additional modifications, which we found to improve the performance of this model on our data. Rather than sampling 3-tuples from the directed graph of job transitions, we include all 2-tuple job transitions present in the data, identical to the other models we consider. Additionally, rather than using the contrastive objective in Equation 4 of Dave et al. (2018), we optimize the log-likelihood directly — this is more computationally intensive but leads to better results. Finally, we include survey-specific covariates (e.g. education, demographics, etc.) by adding them to $\mathbf{w}_x$, embedding the covariate of each most recent job to the same space as $\mathbf{w}_x$. We make similar modifications to Job2Vec (Zhang et al., 2020). Job2Vec requires job titles and descriptions of job keywords, which are unavailable for economic longitudinal survey datasets. Instead, we modify Equation 1 in Zhang et al. (2020) to model occupation codes rather than titles or keywords and optimize this log-likelihood as our objective. We also incorporate survey-specific covariates by embedding each covariate to the same space as $\mathbf{e}_i$ and adding it $\mathbf{e}_i$ before computing Equation 2 from Zhang et al. (2020), which we also found to improve performance. We follow Dave et al. (2018) and use 50 embedding dimensions for each model, and optimize with Adam using a maximum learning rate of 0.005, following the minibatch and warmup strategy described below.

When we compared the transferred version of CAREER to a version of CAREER without pretrained representations, we tried various architectures for the non-pretrained version of CAREER. We found that, without pretraining, the large architecture we used for CAREER was prone to overfitting on the smaller survey datasets. So we performed an ablation of the non-pretrained CAREER with various architectures: we considered 4 and 12 layers, 64 and 192 embedding dimensions, 256 and 768 hidden units for the feedforward neural networks, and 2 or 3 attention heads (using 2 heads for $D = 64$ and 3 heads for $D = 192$ so that D was divisible by the number of heads). We tried all 8 combinations of these parameters on NLSY79, and found that the model with the best validation performance had 4 layers, $D = 64$ embedding dimensions, 256 hidden units, and 2 attention

heads. We used this architecture for the non-pretrained version of CAREER on all survey datasets.

Training. We randomly divide the resumes dataset into a training set of 23.6 million sequences, and a validation and test set of 23 thousand sequences each. We randomly divide the survey datasets into 70/10/20 train/test/validation splits.

The first- and second-order Markov models without covariates are estimated from empirical transitions counts. We optimize all other models with stochastic gradient descent with minibatches. In total, we use 16,000 total tokens per minibatch, varying the batch size depending on the largest sequence length in the batch. We use the Adam learning rate scheduler (Kingma & Ba, 2015). All experiments on the resumes data warm up the learning rate from 10^{-7} to 0.0005 over 4,000 steps, after which the inverse square root schedule is used (Vaswani et al., 2017). For the survey datasets, we also used the inverse square root scheduler, but experimented with various learning rates and warmup updates, using the one we found to work best for each model. For CAREER with pretrained representations, we used a learning rate of 0.0001 and 500 warmup updates; for CAREER without pretraining, we used a learning rate of 0.0005 and 500 warmup updates; for the bag of jobs model, we used a learning rate of 0.0005 and 5,000 warmup updates; for the regression model, we used a learning rate of 0.0005 and 4,000 warmup updates. We use a learning rate of 0.005 for job representation learning and Job2Vec, with 5,000 warmup updates. All models besides were also trained with 0.01 weight decay. All models were trained using Fairseq (Ott et al., 2019).

When training on resumes, we trained for 85,000 steps, using the checkpoint with the best validation performance. When fine-tuning on the survey datasets, we trained all models until they overfit to the validation set, again using the checkpoint with the best validation performance. We used half precision for training all models, with the exception of the following models (which were only stable with full precision): the bag of jobs model with covariates on the resumes data, and the regression models for all survey dataset experiments.

The tables in Section 4.4 report results averaged over multiple random seeds. For the results in Figure 4.2(a), the randomness includes parameter initialization and minibatch ordering. For CA-

REER, we use the same pretrained model for all settings. For the forecasting results in Table 4.1, the randomness is with respect to the Monte-Carlo sampling used to sample multi-year trajectories for individuals.

Forecasting. For the forecasting experiments, occupations that took place after a certain year are dropped from the train and validation sets. When we forecast on the resumes dataset, we use the same train/test/validation split but drop examples that took place after 2014. When we pretrain CAREER on the resumes dataset to make forecasts for PSID and NLSY97, we use a cutoff year of 2014 as well. We incorporate two-stage prediction into the baseline models because we find that this improves their predictions.

Although we do not include any examples after the cutoff during training, all models require estimating year-specific terms. We use the fitted values from the last observed year to estimate these terms. For example, CAREER requires embedding each year. When the cutoff year is 2014, there do not exist embeddings for years after 2014, so we substitute the 2014 embedding.

We report forecasting results on a split of the dataset containing examples before and after the cutoff year. To make predictions for an individual, we condition on all observations before the cutoff year, and sample 1,000 trajectories through the last forecasting year. We never condition on any occupations after the cutoff year, although we include updated values of dynamic covariates like education. For forecasting on the resumes dataset, we set the cutoff for 2014 and forecast occupations for 2015, 2016, and 2017. We restrict our test set to individuals in the original test set whose first observed occupation was before 2015 and who were observed to have worked until 2017. PSID and NLSY97 are biennial, so we forecast for 2015, 2017, and 2019. We only make forecasts for individuals who have observations before the cutoff year and through the last year of forecasting, resulting in a total of 16,430 observations for PSID and 18,743 for NLSY97.

Rationalization. The example in Figure 4.3 shows an example of CAREER's rationale on PSID. To simplify the example, this is the rationale for a model trained on no covariates except year. In order to conceal individual behavior patterns, the example in Figure 4.3 is a slightly altered version

of a real sequence. For this example, the transformer used for CAREER follows the architecture described in Radford et al. (2018). We find the rationale using the greedy rationalization method described in Chapter 2. Greedy rationalization requires fine-tuning the model for compatibility; we do this by fine-tuning with "job dropout", where with 50% probability, we drop out a uniformly random amount of observations in the history. When making predictions, the model has to implicitly marginalize over the missing observations. (We pretrain on the resumes dataset without any word dropout). We find that training converges quickly when fine-tuning with word dropout, and the model's performance when conditioning on the full history is similar.

Greedy rationalization typically adds observations to a history one at a time in the order that will maximize the model's likelihood of its top prediction. For occupations, the model's top prediction is almost always identical to the previous year's occupation, so we modify greedy rationalization to add the occupation that will maximize the likelihood of its *second-largest* prediction. This can be interpreted as equivalent to greedy rationalization, albeit conditioning on switching occupations. Thus, the greedy rationalization procedure stops when the model's second-largest prediction from the target rationale is equivalent to the model's second-largest prediction when conditioning on the full history.

Appendix D: Adjusting the Gender Wage Gap for Full Job History

D.1 Estimation details

Here, we include extra details underlying our estimation procedure. The projection algorithm described in Chapter 5 requires minimizing the predictive error of wage before projecting to the space of sufficient representations. The minimization objective is the mean-squared error between true and predicted wages:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_i (Y_i - \hat{E}[Y | G_i, X_i, \lambda_\theta(H_i)])^2. \tag{D.1}$$

This objective is minimized with respect to the parameters that comprise the representation, θ, along with the conditional wage function parameters $g_F, g_M, \gamma_F, \gamma_M, \beta_F, \beta_M$. We use an alternating update scheme to optimize parameters: the representation parameters θ and γ_M, γ_F, g_M, and g_F are optimized with gradient descent; after each gradient step, β_M and β_F are set to their closed-form OLS optima, keeping the other parameters fixed. We find that this coordinate ascent update scheme finds better optima than updating all parameters simultaneously.

We also define a gender propensity model for projection:

$$P(G_i = F | X, \lambda_\theta(H)) = \sigma\left(\psi^\top X + \rho^\top \lambda_\theta(H)\right), \tag{D.2}$$

where $\sigma(\cdot)$ denotes the inverse-logit function and $\psi \in \mathbb{R}^P$ and $\rho \in \mathbb{R}^D$ are regression coefficients.

The full projection procedure is depicted in Algorithm 3.

In practice, the adjusted wage gap estimate can be sensitive to the initialization of the model parameters on resume data. Ideally, augmenting the estimation procedure with resume data shouldn't result in additional variance. Thus, we use multiple models, or *ensembles*, to learn initial rep-

Algorithm 3: Sufficiency-constrained optimization

Input: N samples $X, Y, G, H \sim P$, initial representation $\hat{\theta}$

Output: Parameters $\hat{\theta}$ that induce approximate solution to Equations 5.7 and 5.8, estimated functions $\hat{E}[Y|G, X, \lambda_\theta(H)]$ and $\hat{P}(G|X, \lambda_\theta(H))$.

while gender and wage validation errors are improving **do**

1. **Minimization Step:**
   - Initialize $\theta = \hat{\theta}$.
   - Perform gradient descent to minimize $\frac{1}{N}\sum_i (Y_i - \hat{E}[Y|G_i, X_i, \lambda_\theta(H_i)])^2$ with respect to θ and other parameters.
   - Define $\hat{\theta}$ to be the parameters that induce the best validation loss.
2. **Projection Step:**
   - Initialize $\theta = \hat{\theta}$
   - Perform gradient descent to maximize $\frac{1}{N}\sum_i \log \hat{P}(G_i|X_i, \lambda_\theta(H_i))$ with respect to θ and other parameters.
   - Define $\hat{\theta}$ to be the parameters that induce the best validation score.

end

Perform final minimization step until validation loss convergence.

return $\hat{\theta}$, $\hat{E}[Y|G, X, \lambda_\theta(H)]$, $\hat{P}(G|X, \lambda_\theta(H))$.

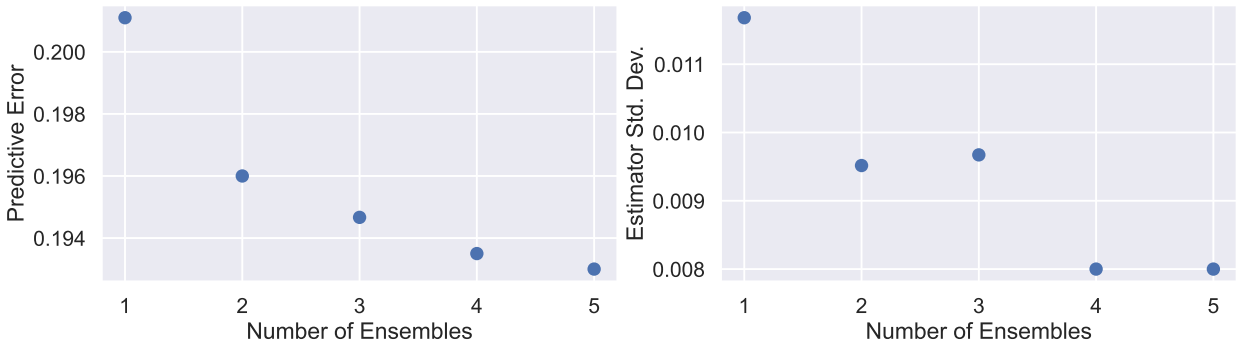


Figure D.1: The predictive error of wage and variance of the history-adjusted wage gap estimator as a function of the number of ensembles used. Increasing the number of ensembles improves the predictive accuracy of wage and lowers the variance of the estimator.

resentations on the resume data, each model identical except initialized with a different random seed. We then perform the projection process for each model, and estimate wage gaps by averaging predictions for all models. This process is inspired by ensemble methods in machine learning (Dietterich, 2000; Lakshminarayanan et al., 2017).

This procedure reduces the variance of the adjusted wage gap estimate; by the law of large numbers, as the number of ensembles goes to infinity, the variance induced by pretraining random seed goes to zero. Additionally, we find that averaging the predictions of ensembles improves wage predictions compared to models that do not ensemble. See Figure D.1 for empirical evidence.

Our end-to-end estimation procedure is given in Algorithm 4. This procedure includes resume pretraining with ensembles, projecting each representation with Algorithm 3 via cross-fitting (Chernozhukov et al., 2018), and clipping to ensure overlap. The results here and in Chapter 5 use $S = 5$ splits for cross-fitting and $\tau = 0.05$ as the clipping threshold (except where otherwise stated). We also use $K = 20$ ensembles for the 2010 and 2018 results, and $K = 10$ ensembles for all other years.

D.2 Predictive performance

Here, we provide more detailed for the predictive performance of models. Table D.1 depicts the mean-square error of predicted wage for various models. (These results are also depicted in

Algorithm 4: Estimating the history-adjusted gender wage gap

Input: Corpus of resumes $\mathcal{D}_R$ containing job sequences H; survey dataset $\mathcal{D}_S$ containing wage Y, gender G, covariates X, and histories H; number of ensembles K; number of splits S for cross-fitting; clipping threshold τ.

Output: Estimate of the history-adjusted gender wage gap (Equation 5.3) on $\mathcal{D}_S$.

Divide survey dataset $\mathcal{D}_S$ into S splits randomly.

for split s in $1 \dots S$ **do**

  Define $\mathcal{D}^{(s)}_{S,\text{test}} = \{(Y_i, G_i, X_i, H_i) \in \mathcal{D}_S \text{ s.t. } i = s\}$.

  Randomly split remainder of $\mathcal{D}^{(s)}_S$ into train set $\mathcal{D}^{(s)}_{S,\text{train}}$ and validation set $\mathcal{D}^{(s)}_{S,\text{valid}}$.

end

for ensemble k in $1 \dots K$ **do**

  Initialize parameters θ_k randomly.

  Pretrain on resume corpus $\mathcal{D}_R$ to optimize Equation 5.10 with respect to λ_{θ_k}

  for split s in $1 \dots S$ **do**

    Set $\theta_{k,s}$, $\hat{E}[Y|G, X, \lambda_{\theta_{k,s}}(H)]$, $\hat{P}(G|X, \lambda_{\theta_{k,s}}(H))$ by performing sufficiency-constrained optimization (Algorithm 3) with $\mathcal{D}^{(s)}_{S,\text{train}}$, $\mathcal{D}^{(s)}_{S,\text{valid}}$, $\theta_{k,s}$.

    for index i in $\mathcal{D}^{(s)}_{S,\text{test}}$ **do**

      Set $\hat{Y}^{(M)}_{i,k} = \hat{E}[Y|G = M, X_i, \lambda_{\theta_{k,s}}(H_i)]$.

      Set $\hat{Y}^{(F)}_{i,k} = \hat{E}[Y|G = F, X_i, \lambda_{\theta_{k,s}}(H_i)]$.

      Set $\hat{p}_{i,k} = \hat{P}(G = F|X_i, \lambda_{\theta_{k,s}}(H_i))$.

    end

  end

end

Set $\hat{Y}^{(M)}_i = \frac{1}{K}\sum_k \hat{Y}^{(M)}_{i,k}$.

Set $\hat{Y}^{(F)}_i = \frac{1}{K}\sum_k \hat{Y}^{(F)}_{i,k}$.

Set $\hat{p}_i = \frac{1}{K}\sum_k \hat{p}_{i,k}$.

Define $\mathcal{D}_{F,\text{clipped}} = \{(\hat{Y}^{(M)}_i, \hat{Y}^{(F)}_i) \in \mathcal{D}_S \text{ s.t. } \hat{p}_i > \tau \text{ and } \hat{p}_i < (1-\tau) \text{ and } G_i = F\}$.

Calculate history-adjusted wage gap: $\frac{1}{|\mathcal{D}_{F,\text{clipped}}|}\sum_{i \in \mathcal{D}_{F,\text{clipped}}}(\hat{Y}^{(F)}_i - \hat{Y}^{(M)}_i)$.

return history-adjusted wage gap

	2006	2008	2010	2012	2014	2016	2018
Summary statistics + covariates	0.235	0.220	0.217	0.221	0.230	0.229	0.241
Full history + covariates (without resumes)	0.234	0.221	0.214	0.223	0.227	0.226	0.242
Full history + covariates (with resumes)	**0.197**	**0.194**	**0.191**	**0.194**	**0.203**	**0.205**	**0.214**

Table D.1: Mean-square error of wage prediction on held-out data for different years of the PSID survey. The first row ("Summary statistics + covariates") uses the model in Blau & Kahn (2017) to predict wage, including covariates and hand-constructed summary statistics about past employment (years of full- and part-time work and their squares). The second and third rows use the methodology proposed in Chapter 5; they differ only in whether a large, passively-collected corpus of resumes is leveraged to improve the learned representations. Figure 5.2 in Chapter 5 plots these results.

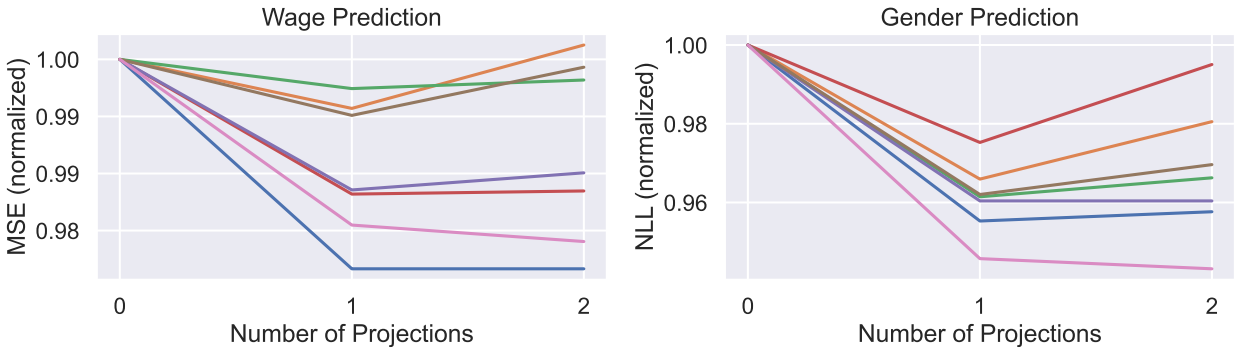


Figure D.2: Wage mean-square error and gender negative-log likelihood as a function of projection round. Each line is a different year of PSID. Prediction errors are normalized with respect to the non-projected prediction error. Projection improves the predictive performance of both wage and gender models.

in Figure 5.2 in Chapter 5.) Meanwhile, Figure D.2 shows how the predictive performance varies as a function of projection round. Projecting improves the performance of both wage and gender predictive models.

D.3 Clipping

Figure D.3 depicts how adjusted wage ratios vary as a function of the clipping threshold. While the absolute estimates are sensitive to clipping, the magnitude of the history-adjusted gender wage gap relative to the non-history adjusted gender wage gap is consistent across clipping thresholds.

Table D.2 shows how covariates and wages are affected by clipping for different clipping thresholds. Most variables have similar values across the depicted clipping thresholds. One ex-

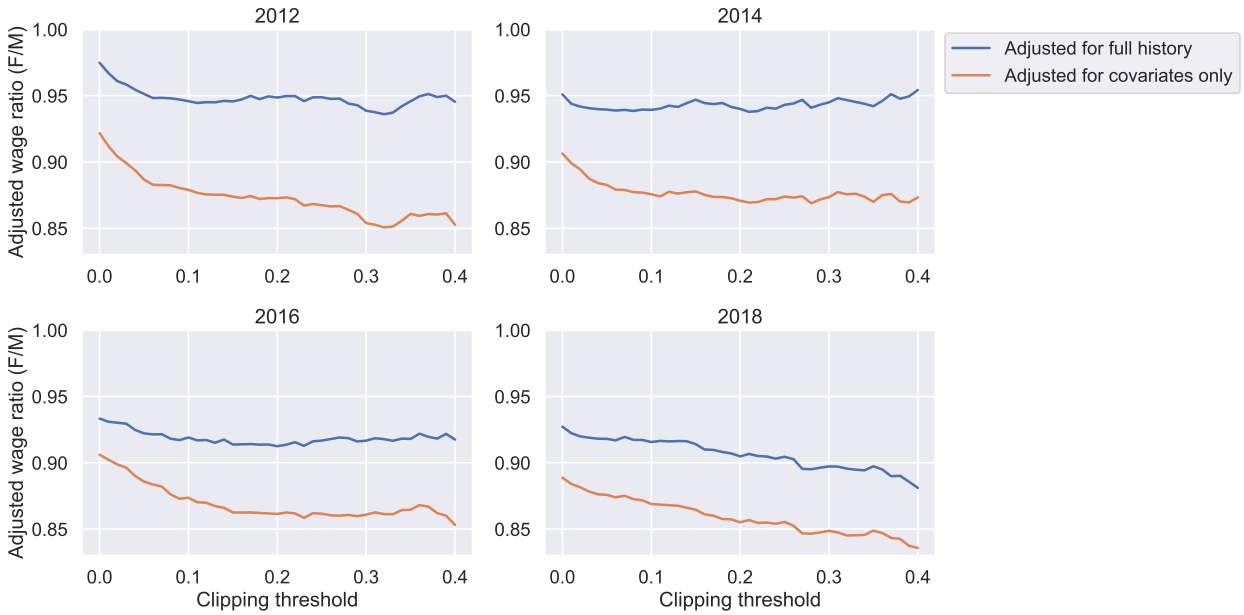


Figure D.3: The non-history adjusted wage ratio and full-history adjusted wage ratio for different clipping thresholds on four different years of PSID. Each wage function is estimated using the full data, then data is thresholded according to CAREER's propensity score estimate, which uses both history trajectories and covariates. The wage ratios are estimated on this thresholded data.

ception is part-time experience: clipping reduces the mean years of part-time experience for individuals in the dataset. Since females are likelier to have part-time experience than males, clipping reduces the extreme values for part-time experience since they are very unlikely to be males. The other noticeable change occurs for office and administrative support; the proportions of individuals in these jobs decreases as the clipping threshold increases. Females are much likelier than males to have some of these jobs, such as secretarial roles, and they are thus clipped.

D.4 Adjusted Wage Ratios

Table D.3 depicts the adjusted gender wage ratios in tabular form. These results are depicted in graphical form in the main text in Figure 5.3.

	Clipping threshold:	0.00	0.05	0.10
	Number of observations	2632	1903	1359
Wage variables	Log wage	3.10	3.16	3.22
	Unadjusted wage ratio	0.80	0.79	0.82
Experience variables	Full-time experience (years)	19.72	19.41	18.90
	Part-time experience (years)	4.35	3.92	3.88
Education variables	Education (years)	14.87	15.12	15.37
	No degree	0.56	0.50	0.44
	College degree	0.26	0.29	0.32
	Advanced degree	0.18	0.21	0.24
Region variables	Northeast	0.21	0.20	0.20
	North Central	0.28	0.28	0.27
	South	0.34	0.35	0.35
	West	0.17	0.17	0.18
Demographic variables	White	0.75	0.76	0.77
	Black	0.18	0.16	0.16
	Hispanic	0.03	0.03	0.04
	Other race	0.05	0.04	0.03
Occupation variables	Managers	0.12	0.13	0.15
	Business Operations Specialists	0.04	0.05	0.06
	Financial Operations Specialists	0.03	0.03	0.04
	Computer and Math Technicians	0.02	0.02	0.02
	Architects and Engineers	0.01	0.01	0.01
	Life, Physical, and Social Science Technicians	0.02	0.02	0.03
	Social Workers	0.03	0.04	0.04
	Postsecondary Educators	0.02	0.02	0.02
	Other Education, Legal, and Library Workers	0.11	0.11	0.11
	Art Design, Entertainment, Sports, and Media	0.01	0.02	0.02
	Lawyers and Physicians	0.02	0.02	0.03
	Nurse, Healthcare Practitioners, and Technicians	0.09	0.09	0.08
	Healthcare Support	0.05	0.02	0.02
	Protective Service	0.01	0.02	0.02
	Food Prep, Serving, and Personal Care	0.06	0.06	0.05
	Building, Grounds Cleaning, and Maintenance	0.01	0.02	0.01
	Sales Occupations	0.06	0.07	0.08
	Office and Administrative Support	0.22	0.17	0.13
	Construction, Extraction, and Installation Occupations	0.01	0.01	0.01
	Production Occupations	0.03	0.04	0.05
	Transportation and Materials Moving	0.02	0.02	0.02
Industry variables	Mining and Construction	0.01	0.01	0.01
	Durable Manufacturing	0.06	0.06	0.07
	Non-Durable Manufacturing	0.03	0.04	0.04
	Transport	0.03	0.03	0.03
	Utilities	0.01	0.01	0.01
	Communications	0.01	0.02	0.02
	Retail Trade	0.06	0.07	0.07
	Wholesale Trade	0.02	0.02	0.03
	Finance	0.10	0.11	0.11
	Social Work, Arts, Recreation, and Other	0.07	0.07	0.06
	Hotels and Restaurants	0.05	0.06	0.05
	Professional and Related Services	0.08	0.09	0.11
	Medical	0.23	0.18	0.15
	Education	0.16	0.17	0.17
	Public Administration	0.08	0.07	0.07

Table D.2: Mean summary statistics for different clipping thresholds for the 2018 PSID sample. The mean of each variable is computed on the subset of female observations whose work histories are in the slice of the gender distribution indicated by the clipping threshold.

	2006	2008	2010	2012	2014	2016	2018
Unadjusted	0.769	0.763	0.772	0.788	0.793	0.790	0.792
Adjusted for covariates only	0.828	0.854	0.888	0.887	0.883	0.886	0.876
Adjusted for full history + covariates	0.891	0.908	0.937	0.951	0.940	0.921	0.918

Table D.3: Estimates of the adjusted gender wage ratio for different years of the PSID survey. The sample includes wage and salary workers between 25 and 64 years old who worked for at least 26 weeks in non-farm jobs, and whose work histories are in the middle 90% of the gender distribution (to assure overlap). For both adjusted models, we adjust for: years of of full-time and part-time employment and their squares; years of schooling; indicators for bachelors and advanced degrees; race and ethnicity indicators; census region indicators; collective bargaining status; current occupation and current industry. The model with history also adjusts for a learned low-dimensional representation of history. Figure 5.3 depicts these results and standard error estimates.

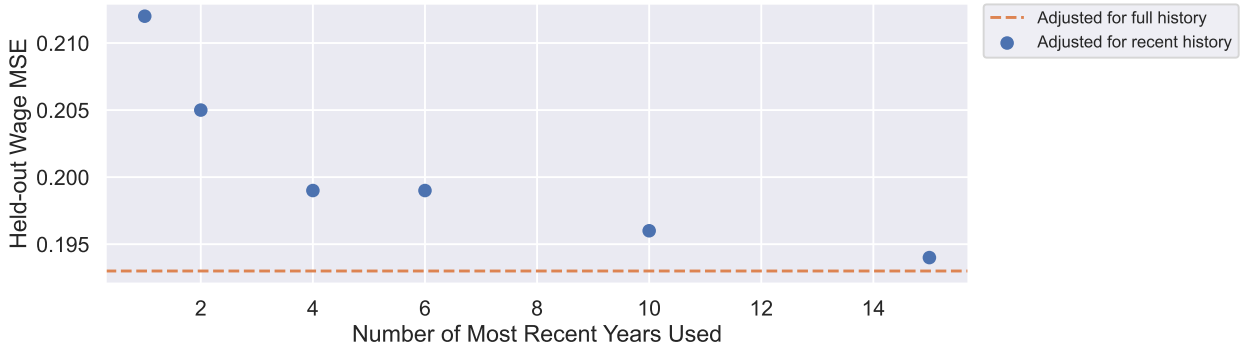


Figure D.4: Mean-square error of wage prediction on held-out data as a function of the number of most recent years used in CAREER's representation of job history for 2010.

D.5 Qualitative Analysis

We conclude by including qualitative analysis that is referenced in Chapter 5. Figure D.4 shows the mean-square error of wage prediction as a function of the number of most recent years used in CAREER's representation of job history. Early on, adding a single additional year significantly improves wage prediction accuracy. As the number of years increases, the benefit plateaus.

Figure D.5 depicts the history-adjusted gender wage ratios for specific occupations. Occupations where history explains the most of the wage gap include: production occupations, transportation and materials moving occupations, office and administrative support occupations, and managers. The least of the wage gap is explained for lawyers and physicians; Chapter 5 contains further analysis for why history doesn't explain the wage gap for lawyers and physicians and why

Occupation	Wage tier	Cluster center	Non-history ratio	Full-history ratio	Model Δ explained	Male Δ	Female Δ
Manager	High	student, student, employed, employed, employed, employed, employed, employed, employed, financial manager, financial manager, manager	0.744	0.846	4.85%	+0.120	+0.249
Office admin	High	student, student, student, student, homemaker, student, homemaker, unemployed, retired, unemployed, data entry keyer, data entry keyer, manager, retail salesperson, stock clerk, marketing manager, marketing manager, marketing manager, marketing manager, laborer, sales supervisor, sales supervisor, office supervisor, stock clerk, stock clerk, stock clerk, stock clerk, stock clerk	0.781	0.859	4.57%	-0.056	+0.039
Production	High	assembler of electrical equipment, assembler of electrical equipment, assembler of electrical equipment, assembler of electrical equipment, assembler of electrical equipment	0.737	0.821	3.94%	-0.01	+0.095
Manager	High	student, employed, student, employed, marketing manager, marketing manager, marketing manager	0.785	0.875	3.91%	-0.002	+0.108
Manager	Low	employed, student, slicing machine operator, printing machine operator, secretary, housekeeper, production supervisor, equipment cleaner, homemaker, nursing aide, food preparation worker, nursing aide, food preparation worker, janitor, janitor, janitor, janitor, cleaning service supervisor, cleaning service supervisor, education manager, cleaning service supervisor, manager, manager, manager, manager, manager	0.891	0.964	3.79%	-0.167	-0.089

Table D.4: History clusters that are most responsible for explaining the difference between the non-history adjusted wage gap and the full-history adjusted wage gap. Clusters are formed by K-Means clustering of current occupation x wage tier categories, denoted by the "Occupation" and "Wage tier" columns. The cluster center in each group is depicted in the "Cluster center" column. The cluster-specific non-history adjusted wage ratio is given in the "Non-history ratio" column, with the full-history adjusted ratio in the "Full-history ratio" column. The "Model Δ explained" column is the percent of the difference between non-history and full-history adjusted wage gaps explained by the current cluster; for example, the difference in adjusted wage gaps between the non-history and full-history adjusted wage gaps for the first cluster is 4.85% of the total difference in model gaps. The column "Male Δ" is the average predicted increase in male wage when adjusting for history, with the analogous result for females in the "Female Δ" column. For example, the average predicted male wage for the first group increases by 0.120 log dollars when conditioning on history, while the average predicted female wage increases by 0.249 log dollars.

it does for managers.

Table D.4 provides clusters of history that explain the largest portion of the difference between the wage gap estimates that do and do not adjust for full history. The cluster where incorporating history explains the most of the wage gap consists of managers with previous jobs that require specialized technical skills; this cluster is analyzed in more detail in Chapter 5.

Figure D.6 compares the representations of history between a model that uses projection to learn sufficient representations and one that does not. When projecting, the learned representations capture gender differences in work history; this is evidenced by the separate clustering of high-

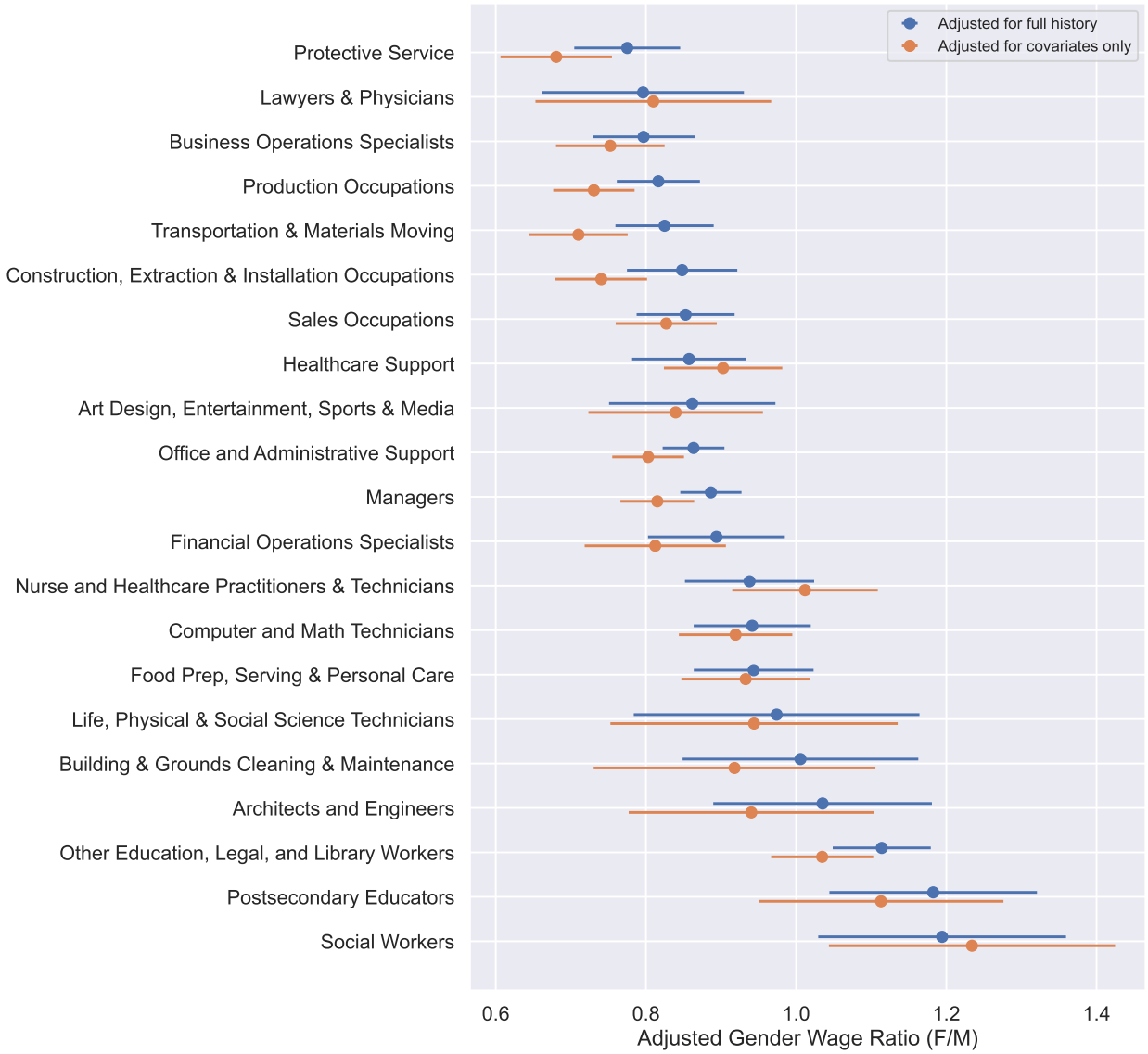


Figure D.5: The history-adjusted gender wage ratio in 2018 for each of the 21 occupational categories defined by (Blau & Kahn, 2017). These are plotted against the non-history adjusted gender wage ratios. First, a sufficient representation is learned for the full dataset by following the procedure from Algorithm 3. The data is then subsetted by an individual's current occupation, and the adjusted wage gap is estimated for each subset.

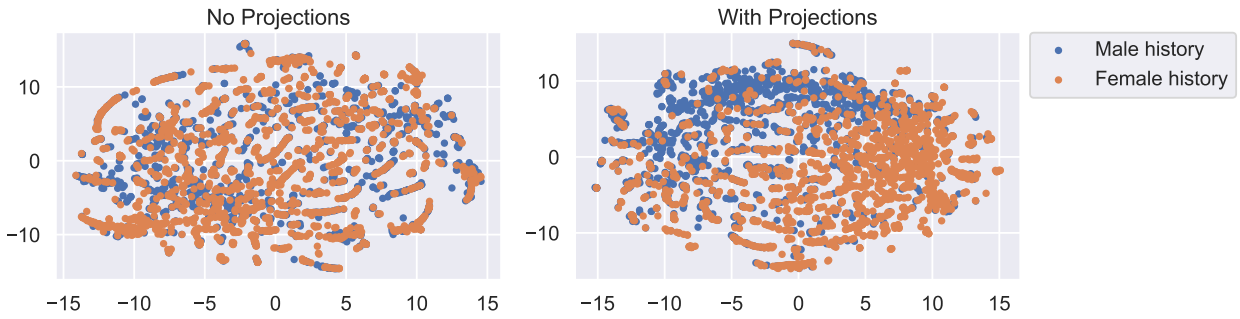


Figure D.6: The representations learned by CAREER on real data from PSID (2018), with and without projecting. The representations are depicted in 2D using t-SNE (Van der Maaten & Hinton, 2008). When projecting (right), the representations capture gender differences in work history.

propensity female histories and high-propensity male histories.