job2vec: Learning a Representation of Jobs*

Sarah Bana[1], Erik Brynjolfsson[1,2], Daniel Rock[1,3], and Sebastian Steffen[1,4]

Abstract

Job postings provide unique insights about the demand for skills, tasks, and occupations. Using the full text of data from millions of online job postings, we train and evaluate a natural language processing (NLP) model with over 100 million parameters to classify job postings' occupation labels and salaries. To derive additional insights from the model, we develop a method of injecting deliberately constructed text snippets reflecting occupational content into postings. We apply this text injection technique to understand the returns to several information technology skills including machine learning itself. We further extract measurements of the topology of the labor market, building a "jobspace" using the relationships learned in the text structure. Our measurements of the jobspace imply expansion of the types of work available in the U.S. labor market from 2010 to 2019. We also demonstrate that this technique can be used to construct indices of occupational technology exposure with an application to remote work. Moreover, our analysis shows that data-driven hierarchical taxonomies can be constructed from job postings to augment existing occupational taxonomies like the SOC (Standard Occupational Classification) system. Exploring further the model structure, we find that between 2010 and 2019, occupations have become increasingly distinct from each other in their language, suggesting a rise in specialization of tasks in the economy. This trend is strongest for managerial, computer science, and sales occupations.

*[1] Stanford Digital Economy Lab, USA, [2]National Bureau of Economic Research, USA, [3]Wharton School, University of Pennsylvania, USA [4] MIT Sloan, USA. We thank Google Cloud Platform (GCP) and Stanford's Institute for Human-Centered Artificial Intelligence for supporting the computing necessary for this project. We also thank Nabeel Gillani, Lindsey Raymond, Dokyun Lee, Yonadav Shavit, participants of the Workshop on Information Systems and Economics (WISE), and the seminar participants at CMU Tepper, the University of Nebraska–Lincoln, and the Bureau of Labor Statistics for helpful comments. We thank Bledi Taska and Burning Glass Technologies for providing the job postings data, and Cary Sparrow and Greenwich.HR for providing the salary data that made this work possible. The Stanford Digital Economy Lab provided generous financial support.

1 Introduction

IT-enabled innovations are transforming work. Traditional survey-based measurements are too static and coarse to keep up with the rapidly-changing demand for labor and skills. The complexity of these labor market dynamics has led to the rise of new jobs as well as the redefinition and mixing of the task-content of occupations. Documenting and analyzing these dynamics is an essential endeavor to understand the nature of work and to help workers navigate its rapid changes. Data from job postings contain the high granularity information required for this. The text of job vacancy postings provide a market-tested list of demanded skills, geography, timing, and other pertinent information about the work. There is a deep literature in the economics of information systems that studies the transformation of occupations, their technical content, and the strong complementarities between technology and human capital (Ang and Slaughter, 2001; Ang et al., 2002; Bresnahan et al., 2002; Bapna et al., 2013; Mehra et al., 2014; Tambe, 2014). Our work continues in this tradition by studying occupational change through the text of job postings.

Advances in natural language processing (NLP) provide a tremendous opportunity to quantify the language of jobs. Machine learning techniques distill substantial quantities of information into forms that support rigorous analysis. To give a simple example, despite the intuition that jobs posted for the same occupation differ across firms, it would be challenging to evaluate this hypothesis using traditional methods because structured data cannot cover all relevant dimensions. Coding ten thousand job postings and counting the distinct words they were using would create an extremely sparse matrix, yielding little value for analysis. Furthermore, this approach would be misleading if the word "machine" appeared in different contexts: "operate machine" versus "machine learning" or if the order of the words matters: "machine learning" versus "learning machine." Contextual word embeddings codify words into a numeric vector space, based on their context in a posting combined with previous uses of those words in sources such as Wikipedia, thereby turning words into objects that can be quantified and compared.

Job postings are inherently challenging to analyze because their format is usually unstructured. For example, simple methods such as counting words in a posting may misconstrue actual meaning: the word Python could highlight the need for a zookeeper, a computer programmer or even a a British comedian - context matters! In this paper, we leverage developments in natural language processing (NLP) on contextual embedding spaces to create a high-dimensional vector representation of the latent dimensions of jobs. We call our system *job2vec* in homage to word2vec (Mikolov et al., 2013), a pioneering word embedding model. We then use our approach to better understand the language of work. We explore the resulting vector space of jobs and seek to make our machine learning approach as interpretable as possible through thorough validation and text injection exercises.

Using the full text of data from millions of online job postings, we train and evaluate a natural language processing (NLP) model with over 100 million parameters to learn the language of jobs. First, we use our model to build an occupational classifier from job text data that matches the text from a set of postings to one or more possible occupational labels. Our classifier accurately predicts the exact occupational label for over sixty percent of previously unseen job postings - an impressive feat considering the multi-class problem involves nearly 1000 labels. We use BERT (Devlin et al., 2018), a contextual word embedding, as a pre-processing layer of a multi-class classification neural network architecture. This approach has reached superhuman performance on several language tasks and vastly outperforms non-contextual approaches, such as word2vec and other bag-of-words methods. The model architecture includes this BERT layer, followed by a convolution layer, global max pooling, and batch normalization (described in depth below). While there have been some other early efforts to apply deep learning for occupational text (Elsafty et al., 2018; Li et al., 2020; Shi et al., 2020), to our knowledge, we are the first to apply transformer-based deep learning to understand the shifting structure of the labor market.

Training two models on 2010 and 2019, and comparing the generated predictions, we identify a series of important findings about the past decade's evolution of labor demand. This approach allows us to detect the relative rates of transformation for different occupations in the economy. We operationalize these transformation rates with the aggregated entropy scores of the predicted classifications from our model for each posting.

First, while the accuracy for our models is similar across both years, the entropy is much higher overall in 2010 than it is in 2019, which suggests that occupations, on average, have become more distinct over time. This time trend effect is biggest for IT occupations, managerial occupations, and sales occupations, all of which have been affected deeply by information technology over the past decade.

Our approach provides a formal method for addressing the important question of job creation and destruction (Acemoglu and Restrepo, 2018). We find that the "jobspace", the convex hull of our spatial representation of job postings by year, is larger in 2019 relative to 2010, indicating a greater variety of demanded jobs. Because calculating a convex hull on such a large dataset (both in terms of rows and columns) is computationally demanding, we introduce a new method for calculating statistics on the higher dimensional convex hull that we call the "M-Bag". This M-Bag approach yields estimates of the change in volume and surface area of the spaces represented by postings across years. Our methods show that the volume of the jobspace has expanded considerably in the past decade, reflecting greater varieties in the extensive margin of work. At the same time, very few combinations of work are disappearing. This suggests that the last decade has primarily been a period of innovation in the types of viable work and that there have been few examples of occupational varieties that have been destroyed.

2 Data

We use data from Burning Glass Technologies (BGT), which scrapes and annotates job postings from over 40,000 distinct online job platforms in the United States. Their sample covers ten years of job postings, from January 2010 to December 2019–a total of 206,567,391 postings. These postings are annotated with several labels, including one of the 824 six-digit 2010 Standard Occupation Classification (SOC) codes.[1] To the best of our knowledge, we are the first to use the actual text of job postings from BGT, as opposed to specific word or skill counts.

[1]A small share of BGT postings, 7,986,131 (4%) are not tagged with six digit SOC codes. We exclude these postings in this part of the analysis.

While the BGT data set is by far the largest and most current data set of job postings, it is not comprehensive. Not all companies post their jobs on the Internet and some specifically block BGT from scraping their postings. Furthermore, the scraping technology is imperfect, inevitably missing some postings and collecting duplicates of others. While BGT seeks to de-duplicate the postings, postings that are similar but not identical may not be removed and therefore may get counted multiple times. Our data runs from 2010 until 2019 but is more comprehensive in later years, both because of improvements in BGT's techniques for finding postings and because more and more postings are digitized and placed on the internet. These variations are not necessarily equally likely to affect all types of occupations, firms, geographies or other dimensions of interest evenly. In ongoing work, Dalton, Kahn and Mueller (2021) match BGT with data from the Quarterly Census of Employment and Wages (QCEW) to determine the relationship between vacancies and online postings. We discuss this work and potential approaches to adjust our sample more robustly in Section 6.

Because BGT does not have salary information associated with the majority of postings, we supplement the full text of the posting data from BGT with data from another vendor: Greenwich.HR (GHR). Greenwich.HR began tracking postings in specific industry sectors, such as health, starting in 2016, expanding over time to publicly available companies and to virtually every sector in the economy. As of 2019, their postings contain the near-universe of jobs. Until recently they did not retain the text of the job postings, as their methodology involves using the metadata of the posting, along with proprietary methods to identify the salary of a posting. Salaries exist for 76.1 percent of the GHR postings, a considerably more representative sample.

We perform probabilistic record linkage (also known as fuzzy matching) to connect the full text from the BGT postings, which is crucial for our analysis, with the posting salaries from GHR. We match exactly on city and state, and use Jaro-Winkler distance to match on job title and firm name. We supplement these string matches with the date scraped by each of the data vendors to develop the sample.[2]

[2] Thus far, the date scraped determines which postings are pulled for matching, as this is a computationally intensive endeavor.

Our primary output values from our models are the salaries from the Greenwich.HR data and 2010 SOC Code classifications provided by Burning Glass. These labels are noisy in themselves; there is no specific ground truth to this taxonomy, though there is a signal insofar as these classifications might immediately make sense (or be clearly wrong) to a human observer. We therefore do not have a corpus with gold-standard labels, but rather, with silver-standard ones. In spite of the errors, the labeling makes for a well-specified machine learning problem. Since there is some signal, but also noise on the output side of the equation, this noise constitutes a form of irreducible error for our prediction task. Of course there is noise as well on the inputs, making it harder to train an effective classifier. Overall the noise arising from an imperfect "ground truth" inherently bounds the predictive capacity of our text processing approach.

3 Model Architecture and Accuracy

We develop two models to predict occupation code and salary from the data described above. These models will provide the basis for the analysis. The model architecture is described more explicitly in the Appendix.

3.1 Model Architecture

Our first step is to build a model that predicts a Standard Occupation Classification (SOC) code from a job posting. These occupation codes are the smallest unit of labor measured in the economy, in datasets provided through the American Communities Survey (Census Bureau) and the Occupational Employment Statistics (Bureau of Labor Statistics).

Our primary approach is one of transfer learning, i.e. using models previously trained for a different task on a new task (possibly with additional neural net architecture changes). We train new weights in addition to the BERT layers, following the text approach pioneered by Howard and Ruder (2018) that is becoming commonplace among NLP researchers. Building a model on labeled data represents a classic supervised Machine Learning problem. Given that we have multiple output labels, and each posting is assigned to exactly one of them, it is a multiclass classification problem. Thus, to train a neural network, we employ a softmax output layer, which returns a probability distribution over all the output labels:

$$P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^T \mathbf{w}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^T \mathbf{w}_k}}$$

To predict the final label the model then simply picks the label corresponding to the highest probability.

To calculate the loss function, we employ (Categorical) Cross Entropy and Kullback-Leibler (KL) Divergence as is typical for these types of problems:

$$D_{KL}(\mathbf{p} \mid\mid \mathbf{q}) = \sum_{x \in X} \mathbf{p}(x) \log \frac{\mathbf{p}(x)}{\mathbf{q}(x)}$$

$$H(\mathbf{p}, \mathbf{q}) = -\sum_{x \in X} \mathbf{p}(x) \log(\mathbf{q}(x))$$

The model architecture is laid out visually in Figure 1. To encode the text of the job postings into numeric vectors, the model architecture includes a BERT embedding as its first layer. As BERT is trained on the entirety of the English language Wikipedia and the Brown Corpus, this layer performs a significant role in the model. Each posting is represented as a series of 512 tokens which are inputs to the BERT model, resulting in output tensors of N postings by 512 tokens by 768 output values. Following the BERT layer are 64 one-dimensional convolution layers that have a window of 16 output values each, with stride length 4. These convolution layers are followed by max pooling layers, which take the maximum of the inputs from the convolutional layers over eight convolved output values. These max pooling layers reduce the number of parameters within the model (downsampling), while keeping the valuable information. The max pooled output is flattened into a one-dimensional vector, then followed by a batch normalization layer. This layer normalizes the output of the previous layer, subtracts the batch mean, and divides by the batch standard deviation. The final step is the predictor layer, which outputs a vector of weights via a softmax activation function. There are a total of nearly 110 million parameters in the model, including 259,655 trainable parameters and 109,482,368 non-trainable parameters from the BERT architecture layer (frozen before fine tuning). We estimate two occupation label models with the same structure on postings from 2010 and 2019. Each model is trained on over a million postings.

The salary model has the same layers as the occupation labeling model, except it concludes with a dense layer instead of a softmax layer. Our final dense layer in the salary model then feeds to a mean squared error loss function on the difference between the (logged) actual and predicted salaries.

3.2 Model Accuracy

The performance of our models on previously unseen job postings is unprecedented. We correctly classify 60 percent of unseen postings. The model's validation accuracy is displayed in Figure 1. Atalay et al. (2020) is the only other estimate in the literature, at 36 percent, though there are significant differences in the time frame (1950-2000) and the approach.

Our salary model is trained on far fewer observations (only 150,000 at the time of this writing). The R^2 is 0.74, which is below Marinescu and Wolthoff (2020) but for a more representative, albeit smaller sample.

Since the integer encoding is based on the ordered six-digit SOC codes, errors in offsetting directions in "nearby" integer buckets can reflect near misses for the model. This is best demonstrated visually by Figure 2. This figure compares Actual Class versus Predicted Class for 300,000 postings previously unseen by the model. The diagonal represents the correct label, and close-misses are near the diagonal. For example, Occupational Therapists (SOC 29-1122) and Physical Therapists (SOC 29-1123) correspond to label integers 306 and 307 respectively. Since the ground truth classification is only partially represented by our target labels, this noise technically constitutes a form of irreducible error, but reflects that our model is learning the underlying occupational structure in the postings data. These occupations are similar and therefore likely misclassifications for our model.

We can also look more closely into the model's predictions and misses to understand the extent to which the pre-existing occupational labeling system truly reflects the structure of the economy. The model struggles with the difference between 25-4031 (Library Technicians) and 43-4121 (Library Assistants, Clerical). For 19-4031 (Chemical Technician), the model often predicts 17-3029 (Engineering Technicians, Except Drafters, All Other). Finally, for 31-9011 (Massage Therapists), the second best false label is 39-5094 (Skincare Specialists). These examples are notable because upon further inspection, they are situations where the major group (first two digits) is not the same across pairs. This suggests the opportunity to improve the Standard Occupation Classification system to provide an alternative that takes into account posting similarity.

3.3 Additional Model Validation via the Automated Generation of Labor Market Hierarchies

Another way to validate that the text classification model is generating meaningful predictions is to check the implied occupational hierarchies it generates against the SOC system. Prediction accuracy is limited to discerning if the same label is predicted as is tagged by Burning Glass. Generating a hierarchy allows us to see if the model classifications are generating near misses when incorrect or if the errors are more severe. For example, a classifier that misclassifies an administrative manager as a farmer would perform worse on this dimension than one that categorized administrative managers as belonging to the "managers, all other" category. We build a 23 cluster system to mimic the scale of the SOC 2-digit taxonomy and recover many of the SOC relationships, in addition to some new ones. The prototype 23 cluster taxonomy is included in the appendix. A dendrogram of the generated hierarchy is depicted in Figure 4. We use 180 thousand postings randomly selected from the 2019 test set (but following the training sampling procedure) to generate these results.

We construct the hierarchical clustering as follows: First, each posting's label is predicted by the model and returned as a vector of probabilities over all possible labels. Labels are then grouped by Burning Glass tag and averaged over the probability vectors within occupational tag groupings.

Subsequently the pairwise cosine distance between each occupation tag grouping with the each other is calculated and used to generate an agglomerative clustering. We use Ward linkage to target variance explained by the clustering across probability vectors.

Some of the candidate clusterings we generate include: Management-intensive roles (includes Chief Executives, General and Operations Managers, Administrative Service Managers), Equipment Specialists (includes Wind Turbine Service Technicians, Aircraft Mechanics, and Mining Machine Operators), Computational and Web Roles (includes Software Developers, Computer Network Architects, Mathematicians, and Information Security Analysts), and Production roles (includes Fabric Menders, Pourers and Casters, and Printing Press Operators). More granular clusterings allow this unsupervised technique even more flexibility; these clusterings are fairly coarse and 23 clusters is not quite enough to capture the true data structure. Nevertheless, the results are encouraging. Other techniques, including Poincar/'e embeddings, might improve the automated generation accuracy even further.

4 Model Driven Implications for the Labor Market

The model described above provides a structure to derive insights into the labor market. To measure changes over time, we use the entropy of the predictions. Intuitively, entropy is a measure of disorder or uncertainty. It ranges from 0 to 1, where a low value represents little disorder, whereas a high value represents substantial disorder. The entropy of a fair coin flip is 1: it cannot be distinguished ex ante whether it will land heads or tails.

In our context, if occupations are quite distinct, and thus easy to classify, the entropy will be low. Alternatively, if occupations are very similar and therefore, hard to classify, the entropy will be high.

More formally, entropy is defined as follows:

$$E_j = \frac{-\sum_i^I p * \ln(p)}{I}$$

for all postings i within occupation label j.

The probability measure p is the predicted probability of a posting belonging to an occupation j given the text. So in our context, entropy E corresponds both to an uncertainty level for a model with respect to a posting as well as, compared over time, a measure of how much content a posting's text has for discerning an accurate label.

Table **??** displays the overall economy results by model year and prediction year.

We train five models, one on every other year (2011, 2013, 2015, 2017, 2019). Each model is trained with the same number of postings. Each model is then used to predict the occupation label of 50,000 postings from each of these five years.

Unsurprisingly, the model trained on a given year perform better at predicting postings for the same year in terms of producing lower entropy. The entropy of the 2010 model predicting on 2010 postings is quite similar to the entropy of the 2019 model predicting on the 2019 postings. However, the off diagonal elements are quite different. The 2019 model finds that the 2010 postings are considerably more difficult to classify, entropy over time remains similar in the 2010 model. Further work will estimate year-specific models to truly disentangle the time trend from the model.

Interpreting the results from the 2019 model, as a whole, over time, occupations have become more distinct in their language, making it easier for the model to predict different occupations.

Figure 5 shows changes in entropy using the 2019 model by Standard Occupation Classification (SOC) major group. We find that entropy is decreasing the most for managers, computer science related occupations, and sales occupations. In contrast, we find that entropy is increasing for teaching occupations, science occupations, security occupations and production occupations. This implies a shift in the locus of information processing capabilities of the firm (Galbraith, 1973; Mani et al., 2010). Over the last decade, software engineering and technical roles have become more specialized, while scientist roles require more diffuse expertise.

We then conduct analyses of this time trend by occupational employment and median wage within the occupation. We find that entropy is decreasing more for bigger occupations, and decreasing more for better paid occupations. This finding highlights that at least amongst traditionally posted occupations, firms are requesting more specialists as opposed to generalists.

This complements recent work by Fuller, Hansen, Ramdas and Sadun (2020), which finds that there has been an increased demand for executive information processing skills and social and interactive skills amongst CEOs.

5 Text Injection Experiments

We also develop a new method of using cross-sectional postings data to estimate changes in worker and technology outcomes. Our model supports deeper analysis into specific tasks and skills embodied in each occupation. We conduct a series of text "injection experiments" wherein we randomly supplement a job posting with different text, including tasks, skills or credentials, that is representative of a different set of occupations. Each of these injection experiments is predicted using 50,000 randomly drawn postings from 2019. This allows us to observe the effect of changing the job posting language on the predicted occupation or predicted salary.[3]

5.1 Mathematical Underpinnings of Text Injection for Classification

We develop an approach by which we can measure how a single phrase affects the salary associated with a posting, as well as probability of classification towards (and away from) certain occupations.

This section will formalize the approach, discuss the assumptions necessary for added text to be a causal estimate, and discuss applications.

Thus far, the models described in Section X and Y (occupation labels and salary) can be summarized as

$$Y = f(X|\beta)$$

[3] This is similar in spirit to the causal approaches described by Veitch et al. (2019) and Eckles and Bakshy (2020). In particular, Veitch et al. (2019) propose that with rich enough textual structure, endogeneity concerns due to omitted variable bias are minimized.

where X is the posting text, Y is the outcome, either occupation label or salary. β is the learned parameter vector of weights derived from the BERT layer and training.

Therefore, for a given posting i,

$$y_i = f(x_i|\beta)$$

Adding text to a posting, in this case, denoted as t_i, provides an additional input to the model. Therefore, the posting without added text can be described as

$$y_{i,0} = f(x_i, t_i = 0|\beta)$$

while the posting with added text is

$$y_{i,t} = f(x_i, t_i = t|\beta)$$

We are interested in understanding the average value of t on the probability distribution of occupation labels and on salary. This amounts to an expectation:

$$\mathbb{E}[f(x_i, t_i = t|\beta) - f(x_i, t_i = 0|\beta)]$$

By sampling from all postings a large number of times, these can be treated as independent and identically distributed (i.i.d) random variables. This allows us to draw on the Central Limit Theorem (CLT) for consistency and inference. Simulation based inference methods are an objective of a later iteration of this draft.

However, postings may not be considered i.i.d. In the case of the occupation labels, with the last layer being a softmax function, because the labels are mutually exclusive, the assumption can be weakened. There is also a concern of noisy labels. In the case of a classification problem, noisy labels will attenuate predictions toward common occupations. The functional form of the softmax is useful because the same model with slight perturbations for the text injections will lead predictions to be very similar, differing only by the change in the signal represented by the text injection and the text it replaces (and a constant common to all postings). Differences-in-differences between posting changes before and after text injections can be used to remove the influence of the constant in the case of the softmax. We discuss this procedure below.

For the last layer of $f(x)$, we use the softmax function, which is a generalization of the logistic function to multiple dimensions. The softmax function is frequently used as the final layer of a neural network because it converts values to a probability distribution. It also leads to a set of (log) linear assumptions that help us solve for the effects of changing text on predicted values. Therefore we have a vector output for probabilities of all classes $\mathbf{y}$ we have $f(x) = Pr(\mathbf{Y} = \mathbf{y}|x)$ as the output of the softmax given some input x.

Without a text injection, the prediction problem for a given posting in class k, $x_{i,k}$, will be:

$$\mathbb{E}[Pr(Y = y)|X = x_{ik}] = \frac{e^{X'_{ik}\beta}}{\sum_{k=1}^{K} e^{X'_{ik}\beta}} \forall k \in K$$

The text injection will be defined as Z. As BERT limits its inputs to 512 tokens, in order to add text to a random subsample of postings of varying length, some pre-existing text from the posting will be omitted in order to include additional text. We define α as the masked set of tokens in X that is replaced by Z. In some cases α_i would represent a unit-specific mask on the posting features, but we assume identical masks across postings for now. That is, $\alpha_i = \alpha$.

Let $u_i = (1-\alpha)X_i + X_i'Z$ represent the new posting with the injected and replaced text. Then the function is as follows:

$$\mathbb{E}[Pr(Y = y)|X, Z, \alpha; \beta] = \frac{e^{((1-\alpha)X_i + Z'X_i)'\beta}}{\sum_{j=1}^{J} e^{((1-\alpha)X_j + Z'X_j)'\beta}}$$

Define $X'_m\beta = w_m \forall m$,

$$\ln\Big(\frac{\mathbb{E}[Pr(Y=f(u_i))|X,Z,\alpha;\beta]}{\mathbb{E}[Pr(Y=f(x_i))|X;\beta]}\Big) = \ln\left(\frac{\frac{e^{((1-\alpha)X_i+Z'X_i)'\beta}}{\sum_{j=1}^{J} e^{((1-\alpha)X_j'+Z'X_j)'\beta}}}{\frac{e^{X_i'\beta}}{\sum_{j=1}^{J} e^{X_j'\beta}}}\right)$$

$$\ln\Big(\frac{\mathbb{E}[Pr(Y=f(u_i))|X,Z,\alpha;\beta]}{\mathbb{E}[Pr(Y=f(x_i))|X;\beta]}\Big) = \ln\Big(\frac{e^{w_i'(Z-\alpha)}}{d}\Big)$$

where d is the ratio of the denominators above.

That leaves the final log expected difference between the injected and standard prediction vectors as

$$\delta_i = w_i'Z - w_i'\alpha - \ln(d) \tag{1}$$

The assumption we need to ensure that the changes in predictions are informative is that $w_i'\alpha$ is in expectation zero. That is, the text that we are omitting from the posting does not substantively change the prediction in a certain direction. Instead, the correlation between the text we are adding and the features in the posting, $w_i'Z = \beta X_i Z$ plus a constant $\ln(d)$ drive the change in predicted value. Additionally if the ratio of denominators is 1, the assumption of orthogonality of the mask vector α and weight signal w_m guarantees that the only change in prediction comes from the covariance of the injected signal with the weight signal. This unity ratio of denominators is unlikely, but can be removed as an intercept.

Since this expression above includes a constant term d for any given posting that gets the same text injection, we can describe the difference between *the differences* of two postings i and j in expectation as:

$$\delta_i - \delta_j = w_i'Z - w_i'\alpha - ln(d) - (w_j'Z - w_j'\alpha - ln(d)) = (w_i - w_j)'Z - (w_i - w_j)'\alpha \tag{2}$$

This difference-in-difference procedure removes the influence of d and introduces a possibly weaker condition that helps recover the effect of the injection. Empirically, differencing to remove

d might not be necessary; both denominators for softmax estimation (with and without the text injection) are calculated in the process of generating output probabilities for the function. But with d removed, if the expected $w_i'\alpha$ is zero for all i (a stronger condition), the change in the predicted posting is purely due to the injected signal Z. But if the expected $w_i'\alpha$ is instead the same across all i, but not necessarily zero, then differencing changes between starting and injection-modified postings allows for recovery of the signal effect. Alternatively, if the weight differences between the postings is orthogonal to the difference between the signal and masked token vectors, the residual delta will be zero. This assumption can be explicitly written as:

$$\mathbb{E}\left[(w_i - w_j)'\alpha\right] = 0 \tag{3}$$

In our sample case we are looking at *ranked* changes by occupation, so the common components from d are not relevant but we must maintain the assumption that $w_i'\alpha$ is the same in expectation across postings. For two postings that have both received the same text injection signal Z and the same mask α, the common denominator factor from d can be differenced away as the difference between changes in the postings with and without the text injection. That procedure explicitly describes the change in probabilities, since d is common to all changes across postings.

The above equation yields the difference in predicted probability for each occupation and each posting as a result of the text injection. We aggregate predicted probabilities over a random sample of postings and report these aggregates in the empirical application. This approach can also be used to identify characteristics of occupations which are not well-enumerated or measured in other datasets, but nevertheless describable using natural language.

5.2 Application to IT Skills

We highlight skills associated with information security, machine learning tools, education credentials, and a combination of characteristics. The text injection approach makes it possible to characterize the IT labor market and labor demand using this new source of data.

We begin by focusing on information security. With the rise of big data in general, and the increased adoption of work-from-home practices during, and likely after, COVID-19, firms have become more vulnerable to cybercrime. This makes cybersecurity ever more important for researchers and policymakers (Manyika et al., 2011).[4]

Figure 6(a) displays the top five occupations most likely to be predicted when a cybersecurity-related term is inserted into the posting (see the footnote for the exact text). The occupation with the biggest increase in predicted probabilities is 13-1199: Business Operations Specialists, All Others. This occupation comprises Security Management Specialists. The next occupation predicted is Information Security Analysts. These occupations have substantially larger changes in prediction probability than the next three occupations, which are less consistent with the injected text. Moreover, the text meaningfully changes the predicted probability distribution over the occupational labels. The delta suggests that if that description were added onto 50,000 postings, approximately 800 more of them would be labeled as these two occupations. It is also interesting to see which occupations were less likely: Teachers and Instructors (All Others), Family and General Practitioners, Home Health Aides, Elementary School Teachers, Except Special Education, and Technical Writers. This suggests that adding these skills to workers in these occupations would not be helpful, implying a clear heterogeneity in the returns to skills.

The next experiment involves adding machine learning/deep learning tools: "TensorFlow, Pytorch, Keras, scikit-learn, NLP." This injection experiment is not as powerful as the information security one. The top five occupations predicted, Web Developers, Computer Systems Analysts, Computer Occupations (All Other), Computer Programmers, and Software Developers, Applications, all seem like reasonable fits for these skill sets, although the model misses Computer and Information Research Scientists (colloquially, data scientists). These tools are, for the most part, open source software. The strong predictive shift in the model implies that open source tools continue to be important signals of human capital quality in the market for data science labor (Mehra and Mookerjee, 2012; Nagle, 2019; Rock, 2019).

[4]Also see https://www.ibm.com/security/data-breach and https://newsroom.ibm.com/2020-06-22-IBM-Security-Study-Finds-Employees-New-to-Working-from-Home-Pose-Security-Risk

Thus far, the experiments have only used skills. However, education credentials are another major component of job postings - results and details can be seen in Figure 6(c). Adding the requirement for a Masters or PhD degree in Computer Science moves the postings, as expected, towards Computer and Information Research Scientists and Financial Analysts (a big employer of statistics doctorates), among others. On the other hand, sales workers and medical secretaries seem to infrequently require these credentials.

Our final text injection, displayed in Figure 6(d), combines skills and their application in a realistic way (description in the footnote of Figure 6). In this case, the model readily accepts Computer and Information Research Scientists, but also considers sales workers and sales engineers amongst other occupations. This suggests that sales engineers are similar to these other IT related occupations, despite being considered sales occupations.

We can conduct the same experiment using the salary model, and predict the return to particular skills by calculating the difference in expected wages (i.e. the area under the wage distributions). These salary text injection experiments can be viewed through the lens of an audit study: what salary would the firm offer if it had added that additional skill or education requirement?

In predicting the occupation associated with a given skill, we added the discussed text to a random sample of postings. However, to achieve more reasonable estimates, especially when considering wage premiums, a key decision is the choice of control group. The concept of common support may be necessary: this may not be achieved if the thought experiment is to add the skill of Tensorflow to a truck driver's job posting. This would limit the usefulness of the corresponding estimate. We are in the process of developing estimates with three different relevant control groups:

- All postings: hypothetically, every worker could benefit from "critical thinking," or "communication" skills. For a certain type of skill, the relevant control group could be all workers
- Using the occupation classification model developed independently from the salary model, we can use the outputs of the occupation classification model to develop another control group: a sample of postings which fall within a certain radius of the injected text

- A sample that includes any posting that is considered in the same occupation as a posting that truly contains the injected text

The results described below use the control group of all postings; however, we are in the process of creating the other groups.

These results are displayed in Figure 7. Adding IT skills and education requirements leads to significant increases in salary predictions. For example, in Figure 7 Panel (b), we can see that adding "TensorFlow, Pytorch, Keras, scikit-learn, NLP" lead to 2.5 percent increase in salary. The possibilities for this tool are endless: here, the examples include adding educational requirements, or longer descriptions pertaining to the nature of the work.

We also experimented with injections of skills which are traditionally considered to be 'low skills'. Indeed, we find that manual and physical skill requirements, such as "Move, lift, carry, push, pull, and place objects weighing less than or equal to 50 pounds without assistance", can be associated with decreases in predicted salary.

An alternative approach is to compare the differences in salaries across geographies. Given that a geographic location is generally embedded in the posting, this takes a bit more nuance than merely adding additional text to a posting. Instead, we propose to use Named Entity Recognition to identify geographic entities (such as cities or states) and then replace them with a reference city, thereby transplanting job postings to different locations.

Currently we include phrases and sentences instead of singular words or skills. This is because the NLP model is built using only 512 tokens, and therefore, longer statements provide more accurate signal. This is important to note because it suggests that it is challenging to compare across text injection experiments for the salary model. For even shorter injections, we plan to use integrated gradient methods or avoid the token limitation by summarizing sentence-level embeddings (instead of relying on embeddings that are effectively based on truncated documents). These methods compute an attribution or importance value for individual features (i.e. each word in a posting). The attribution value then explains how much that word, when occurring alongside other words, would have changed the predicted value. This approach is in the spirit of Gillani et al. (2021).

5.3 Application to Remote Work Potential

Finally, the text injection approach can provide insight into characteristics of jobs that may not be well-measured. One example is remote work – there has been substantial emphasis on understanding remotability and occupations throughout the COVID-19 pandemic Dingel and Neiman (2020); **?**. To answer this question, we add the following text to a posting to identify which occupations are least and most remoteable:

> This is a full time remote position, and employees can be based anywhere in the United States.

Occupations that have been typical candidates for remote work in the past are likely to be more easily converted to remote work now. We therefore use the change in predicted probability of a post to fall within each occupational class following the inclusion of the remote work text injection above as a proxy for an increase in the probability of being able to work from home. Aggregating over all occupations for common shifts toward particular classes (more remotable) and common shifts away from others (less remotable), we build an index of remote work suitability. The following occupations, characterized as "Most remotable," experience the greatest shift in predicted probability towards their occupations. The occupations characterized as "Least remotable" experience the greatest shift in predicted probability away from their occupations. Similar to the word2vec vector math converting "King" to "Queen" the vector of (prediction with remote text injection) - (prediction without text injection) yields the (prediction from remote work text alone).

The rankings in Figure 8 reflect this calculation. We find that managerial roles are particularly likely to show up as more exposed to remote work, similar to the findings from correlating large scale survey data with government occupational composition data in Brynjolfsson et al. (2020). Many manufacturing, repair, and service technician roles tend to be less likely to work from home, reflecting the capital intensity of their respective roles. We represent the differences between the baseline prediction without our remote text and the new predictions in Figure 9. The changes for this text are considerably more diffuse.

Our scores compare favorably to Dingel and Neiman (2020). Table 2 shows the results of logistic, probit, and linear probability model regressions of the Dingel-Neiman scores on our model-derived indices of remote work. We have qualitatively similar results across all models, with positive and statistically significant correlations between our scores and the Dingel-Neiman benchmark. Part of this similarity comes from common classification of managerial roles as being possibly remotable.

6 Further quantifying the Jobspace

One of the principal advantages of representing text from job postings in the form of a numerical lower-dimensional space is the ability to use the spatial arrangement of postings in this space to make inferences about how the labor market is changing. The arrangement of text in our models' internal embeddings suggests an empirical manifold that can shrink, expand, become more densely populated, or more porous as the structure of work changes (or, more specifically, the language used to describe work). We additionally have an interpretable simplex from the final softmax layer. Recall that the support of the predicted values for each occupational label is between 0 and 1, which means that the softmax layer's output is interpretable as a probability. The simplex of all 824 occupational label prediction values therefore serves as a "jobspace" of sorts, with each posting represented by a 824-dimensional vector of probabilities. Also, since the occupational labels are constant over the sample period from 2010 to 2019, the location of each posting in the occupational simplex is comparable across time when the same model is used to map text into prediction probability vectors. While the models we train as classifiers may be practically useful for prediction, it is this spatial representation of the structure of job postings that is the core goal of our analysis.

There is some parallel work that is similar in spirit but operates on patent text data instead. We follow (Boden et al., 2004), (Cheng et al., 2020), who characterize patent innovation into three distinct categories, by identifying three distinct occupational innovation categories.

First, there is *combinatorial* innovation. This happens when units within an existing spatial boundary combine in a way that populates an interior point in the space, i.e. if S is the full space represented by the simplex, for some points $X_1, X_2 \in S$ with some $\alpha \in [0,1]$ there is a new point $X_3 = \alpha X_1 + (1-\alpha)X_2$. The unit vectors representing perfect certainty for each occupation form a basis and bound for our jobspace, and interior points are described by the set of points spanned by this basis. The convex hull of points in a given dataset defines the jobspace for those postings. The 2019 jobspace is therefore defined by the convex hull of the points from 2019 postings within the 824-dimensional occupational simplex. A new combination of two jobs, for example, might be difficult to classify, but still exists within the same jobspace.

Second is an *exploratory* innovation in a job. This happens when the boundary of the jobspace is expanded in some way toward the boundaries of the simplex. It's possible that α in the example above is no longer within the unit interval. Exploratory innovation in occupations means a change in what occupations are doing that was not present in an earlier time period. We study these two types of occupational innovation with summary statistics on projections of the convex hull into a lower dimensional space.

The third type of innovation, *transformative* innovation, involves expanding the simplex dimensionality. In our context, that entails the creation of an entirely new occupation. Since we preserve the same occupational classification system across years for tractability purposes, this type of innovation will be partially allocated, where and when it occurs in the previous decade, to the other two varieties of change.

Importantly, combinatorial innovation possibilities are a major component of the return to exploratory or transformative changes. When employers invent new types of work, they expand the frontier. But they also expand the possible combinations on the interior. At the margin, the benefits for creating new types of potential work should equal the costs of creating those new types of work. That suggests that in equilibrium, the marginal incentives to create transformative, exploratory, or combinatorial improvements in job design are equivalent. This is, in a sense, a multidimensional view of (Acemoglu and Restrepo, 2018).

We describe how the convex hull of the jobspace has changed from 2010 to 2019 using the predicted occupational classification vectors for 100,000 postings from each year. First, we train a model that predicts occupational classifications using 2019 job text data as described earlier. We then apply that model to generate predictions for 2010 and 2019 test data, i.e. another randomly selected 100,000 postings from 2010 and 2019. The rest of the process using those prediction vectors is described below.

6.1 Results on Jobspace Changes and the M-Bag algorithm

We develop and implement a new set of techniques for spatial measurement of the volume of the jobspace. In particular, we would like to know the volume and surface area of the convex hull in some spatial mapping of all postings. This convex hull defines the span of demanded jobs, as opposed to the space of all theoretically possible jobs represented by the simplex. A higher volume indicates that there are more viable combinations of tasks and roles in the economy, whereas regions where the hull might have collapsed or shrunk in recent years indicates that labor demand for those jobs with those "coordinates" is drying up. The surface area of the convex hull, meanwhile, is representative of the extent to which jobs can be on the frontier of the job space. A higher surface area of the convex hull indicates the presence of more frontier jobs. The expansion of the frontier in a given direction suggests that the volume of possible combinations can increase exponentially, while the frontier expansion itself leads to only linear growth in the direction of the expansion. This is because the new types of work enabled by a frontier expansion can be combined with all of the other existing work varieties. On the other hand, a new linear combination of jobs within the space does not create new possibilities outside of the existing jobspace. Comparing the surface area to the volume of the hull therefore proxies for the extent to which a job uniformly selected over the entire job space might itself be a frontier job, or how spherical the space is.

The challenge in calculating this convex hull is primarily computational. Even for small quantities of postings, calculating a 824 dimensional hull (one dimension for each of the 824 possible output labels) is prohibitively costly to compute. To address this issue, we apply a series of machine learning techniques to simultaneously 1) bootstrap 1000 samples of postings from 2010 and 2019, 2) create a lower dimensional embedding space of the probability outputs (25 dimensions to be precise) by stacking these 1000 postings from each year on top of each other and applying a non-negative matrix factorization transformation, and 3) resample the embedding space features such that a convex hull is trivial to compute without sacrificing too much in terms of information.

Our approach allows us to create thousands of draws of convex hull calculations for each year embedded in the lower dimensional space. We refer to this algorithm as the "M-Bag", as we are using lower dimensional embeddings to bootstrap aggregate (bag) measurements of slices of the convex hull object we wish to study. For this study, we project the stacked 2019 and 2010 postings predicted 824-dimensional vectors (1000 postings from each year sampled with replacement) into a 25-dimensional space, then sample 5 dimensions at a time to calculate convex hulls and convex hull statistics. The non-negative matrix factorization solves:

$$\hat{Z} = \min_{W,H} \left\| Z - W'H \right\|_F subject\ to W,H \geq 0 \tag{4}$$

$\hat{Z}$ is the predicted matrix of occupational posting predictions using weight matrix W and factor matrix H. F denotes the Frobenius norm. W' is n postings by 25 factors and H is 25 factors by 824 occupations. Resampling five dimensions at random from W yields $\hat{W}$ as an approximation, and then we proceed by calculating the convex hull $T(\hat{W})$. Importantly the matrix approximation is constructed using both 2010 and 2019 data, and therefore observations in the lower dimensional space $\hat{W}$ are comparable across years. We calculate distributions of the volume and surface area of $T(\hat{W})$ for different draws of the columns of W. These give us insight as to the geometric changes in the estimated jobspace. The results are shown in Figures 10 and 11.

The first of these two figures shows the distribution of convex hull volumes retrieved by following this procedure for 2019, while the latter shows the cumulative density functions for the volumes of both 2010 and 2019. The 2010 distribution nearly first order stochastically dominates the 2019 draws. This suggests that there are very few areas where the convex hull is smaller in 2019 than in 2010 by volume. In other words, there is very little destruction of the total variety of roles over the sample period. If the entire space has shifted somewhat, however, some regions of the 2010 jobspace might no longer apply in 2019.

The relative volume expansion is remarkable. The online job postings space has expanded by approximately 37.5 percent. Of course, the types of roles that have been posted online have grown more diverse and numerous during the sample period. Some of the jobspace expansion represents changes in which types and how many of each occupational category shifted to online markets from 2010 to 2019. Additionally, as shown in 12, the area of the 2019 hull distribution is substantially larger (about 26.8 percent larger). These distributional differences in means are significant with t-stats well over 6. Meanwhile, the surface area to volume (SAV) ratio in Figure 13 shows that the differences between 2010 and 2019 are relatively smaller. With a p-value of .054 on the difference, the 2010 SAV ratio for the distribution of convex hulls is larger than in 2019. The SAV ratio represents how close to spherical the space generated by the data is. More surface area per unit of volume means more frontier opportunities per unit of internal space. Though we fail to reject the null that the distributions are the same, the point estimate evidence that 2010's SAV ratio is larger suggests that there are fewer occupational opportunities on the frontier in 2019 as a proportion of the total possible combinations.

The foregoing discussion assumes that the centroid of the job surface is fixed. We can still make relative claims about the convex hull features without pinning their specific locations to a known center point, but absolute comparisons of possible work types necessitate a common location over years in jobspace. It would be possible, for example, for all jobs to drift in a common direction such that area and volume changes reflected differences in different locations. Fortunately we can calculate the centroid of the space as the average job vector within each year. In simpler terms, a possible challenge to our analysis is that all jobs in our data are compositionally different in a meaningful way. Describing and differencing out the average drift over time of the center helps us quantify the extent to which our sample is moving. The average between year cosine distance over all labels from 2010 to 2019 is relatively small, at only 0.027 (representing a difference of approximately 13 degrees). Our distribution of occupations within sample is changing, but the full change in the jobspace is likely not attributable to the centroid of the jobspace moving.

In essence, the job space is expanding in relative size. As it grows in volume, the myriad possibilities for recombinations of occupations are increasing (at least the ones available online). On the margin, we might expect that the returns to creating a new type of work or recombining existing types of work should be the same. At this point, with so much recent frontier expansion, it stands to reason that there are opportunities to mix-and-match existing types of roles.

7 Conclusion

Job postings provide credible and detailed data on the IT-related skills and activities sought by employers and are available in staggering quantities. By applying and extending recent developments in machine learning, we can understand the key dimensions that differentiation one job from another and understand how individual occupations, and even the entire space of occupations in the US economy are changing. We can create a prediction machine that maps skills and tasks to occupations based on the underlying text of the job postings.

The classification and insights created from these tools allows us to make numerous comparisons and explore hypothetical scenarios. While the predictions are not necessarily causal, they provide a valuable tool for exploring potential effects of training programs or job redesign. We can also identify other nearby jobs, examine the nature of changes in occupations over time, and explore the geography of tasks, skills, and other job characteristics. By drawing on a large body of posted employer requests for labor, our approach provides a path toward better characterizing an occupational taxonomy based on tasks and skills, thereby understanding the changing nature of work.

References

ACEMOGLU, D. AND P. RESTREPO (2018): "The race between man and machine: Implications of technology for growth, factor shares, and employment," *American Economic Review*, 108, 1488–1542.

ANG, S., S. SLAUGHTER, AND K. YEE NG (2002): "Human capital and institutional determinants of information technology compensation: Modeling multilevel and cross-level interactions," *Management Science*, 48, 1427–1445.

ANG, S. AND S. A. SLAUGHTER (2001): "Work outcomes and job design for contract versus permanent information systems professionals on software development teams," *Mis Quarterly*, 321–350.

Atalay, E., P. Phongthiengtham, S. Sotelo, and D. Tannenbaum (2020): “The Evolution of Work in the United States,” *American Economic Journal: Applied Economics*, 12, 1–34.

Bapna, R., N. Langer, A. Mehra, R. Gopal, and A. Gupta (2013): “Human capital investments and employee performance: an analysis of IT services industry,” *Management Science*, 59, 641–658.

Boden, M. A. et al. (2004): *The creative mind: Myths and mechanisms*, Psychology Press.

Bresnahan, T. F., E. Brynjolfsson, and L. M. Hitt (2002): “Information technology, workplace organization, and the demand for skilled labor: Firm-level evidence,” *The quarterly journal of economics*, 117, 339–376.

Brynjolfsson, E., J. Horton, A. Ozimek, D. Rock, G. Sharma, and H. Y. T. Ye (2020): “Covid-19 and remote work: An early look at us data,” *NBER Working Paper*.

Cheng, Z., D. Lee, and P. Tambe (2020): “Robot Inventors? Patents, Generative Algorithms, and Innovation Frontiers,” *Unpublished Working Paper*.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018): “{BERT:} Pre-training of Deep Bidirectional Transformers for Language Understanding,” *CoRR*, abs/1810.0.

Dingel, J. I. and B. Neiman (2020): “How many jobs can be done at home?” Tech. rep., National Bureau of Economic Research.

Eckles, D. and E. Bakshy (2020): “Bias and high-dimensional adjustment in observational studies of peer effects,” *Journal of the American Statistical Association*, 1–30.

Elsafty, A., M. Riedl, and C. Biemann (2018): “Document-based recommender system for job postings using dense representations,” in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, 216–224.

Galbraith, J. R. (1973): *Designing complex organizations*, Addison-Wesley Longman Publishing Co., Inc.

Gillani, N., E. Chu, D. Beeferman, R. Eynon, and D. Roy (2021): “Parents’ Online School Reviews Reflect Several Racial and Socioeconomic Disparities in K–12 Education,” *AERA Open*, 7, 2332858421992344.

Howard, J. and S. Ruder (2018): “Universal language model fine-tuning for text classification,” *arXiv preprint arXiv:1801.06146*.

Li, S., B. Shi, J. Yang, J. Yan, S. Wang, F. Chen, and Q. He (2020): “Deep Job Understanding at LinkedIn,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2145–2148.

Mani, D., A. Barua, and A. Whinston (2010): “An empirical analysis of the impact of information capabilities design on business process outsourcing performance,” *Mis Quarterly*, 39–62.

Manyika, J., M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers (2011): “Big data: The next frontier for innovation, competition, and productivity,” Tech. rep.

Marinescu, I. and R. Wolthoff (2020): “Opening the Black Box of the Matching Function: The Power of Words,” *Journal of Labor Economics*, 38, 535–568.

Mehra, A., N. Langer, R. Bapna, and R. Gopal (2014): “Estimating Returns to Training in the Knowledge Economy,” *MIS Quarterly*, 38, 757–772.

Mehra, A. and V. Mookerjee (2012): “Human capital development for programmers using open source software,” *MIS Quarterly*, 107–122.

MIKOLOV, T., K. CHEN, G. CORRADO, AND J. DEAN (2013): "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*.

NAGLE, F. (2019): "Open source software and firm productivity," *Management Science*, 65, 1191–1215.

ROCK, D. (2019): "Engineering Value: The Returns to Technological Talent and Investments in Artificial Intelligence," *Available at SSRN 3427412*.

SHI, B., S. LI, J. YANG, M. E. KAZDAGLI, AND Q. HE (2020): "Learning to Ask Screening Questions for Job Postings," *arXiv preprint arXiv:2004.14969*.

TAMBE, P. (2014): "Big data investment, skills, and firm value," *Management Science*, 60, 1452–1469.

VEITCH, V., D. SRIDHAR, AND D. M. BLEI (2019): "Using text embeddings for causal inference," *arXiv preprint arXiv:1905.12741*.

	2011	2013	2015	2017	2019
Top 1 Validation Accuracy	0.5906	0.6017	0.5909	0.6042	0.5788
Top 2 Validation Accuracy	0.7007	0.7035	0.6867	0.7064	0.6871
Top 5 Validation Accuracy	0.7979	0.7932	0.7858	0.8025	0.7865
Top 10 Validation Accuracy	0.8504	0.8470	0.8450	0.8573	0.8444
Top 15 Validation Accuracy	0.8737	0.8736	0.8709	0.8838	0.8712
KL Divergence	11253.24	11384.76	11175.69	11289.20	11297.42
Validation Loss	1.9458	1.9570	2.0320	1.8838	2.0371

Table 1: Model Accuracy

Dingel-Neiman (2020) Score Regressions	Logistic	Probit	LPM
J2V Remote Index	0.0428**	0.0259**	0.00899*
	-2.11E-2	-1.29E-2	-4.66E-3
Constant	-0.524***	-0.326***	0.351***
	-7.48E-2	-4.61E-2	-1.68E-2
Observations	773	773	773

Table 2: Comparison with Dingel-Neiman (2020)

Notes: Regression results for three specifications of regressions of Dingel and Neiman (2020) scores on the job2vec text injection experiment results for remote work. *** $p<0.01$, ** $p<0.05$, * $p<0.1$

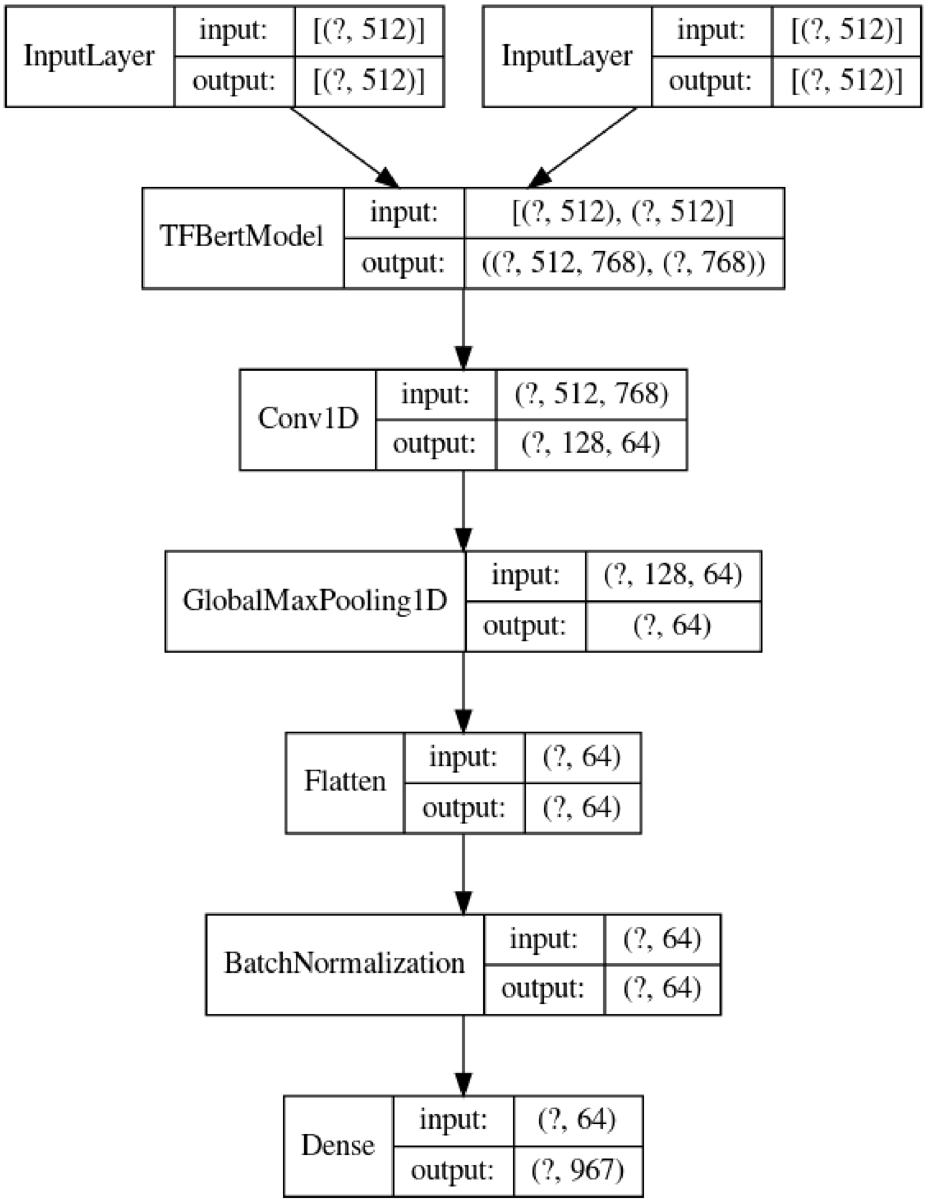


Figure 1: Model Architecture

Notes: Model architecture of the models estimated. The input layers are the tokenized postings (limited to 512 tokens) and corresponding attention masks. They are fed into a BERT layer, which assigns 768 dimensions to each token in the posting. The next layer is a convolutional neural network, which reduces the dimensionality. Then there is a global max pooling layer, which continues to reduce dimensionality while selecting the important characteristics (thus the max). Finally, batch normalization and then a dense layer to output the predictions for each occupation.

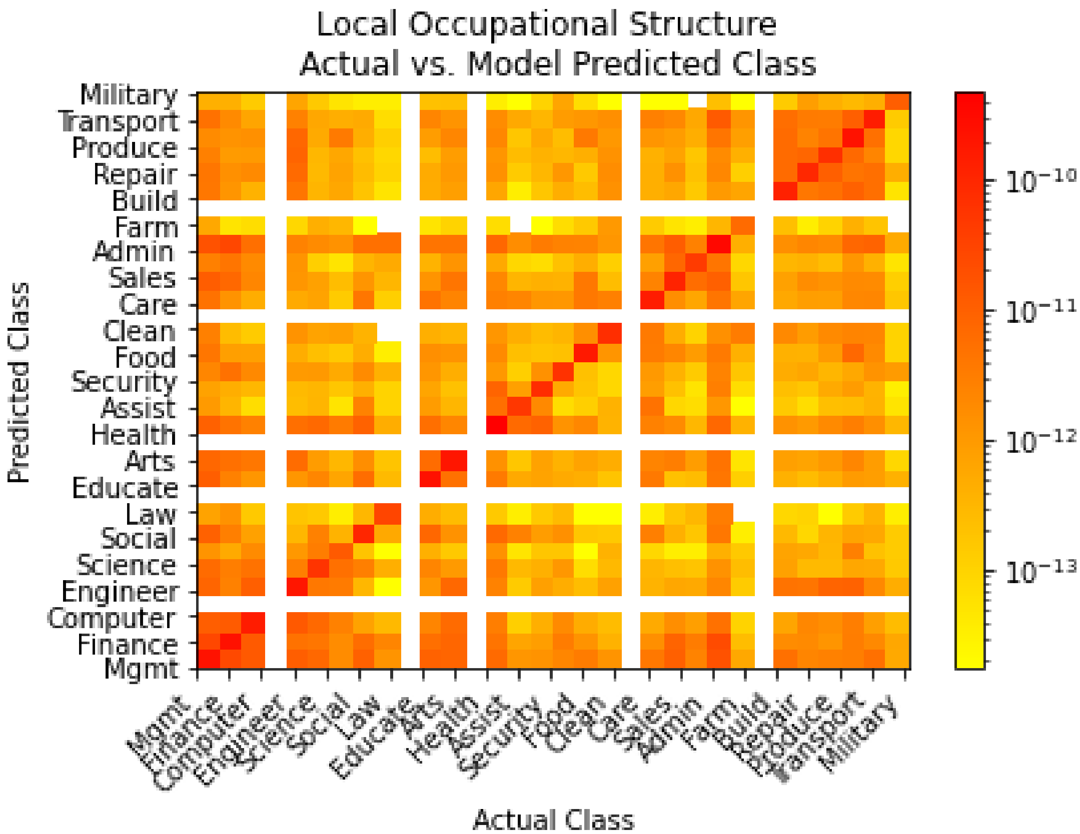


Figure 2: Actual vs Model Predicted Class (Confusion Matrix) for the main 2019 model

Notes: This figure demonstrates the actual versus predicted classes for the model for 300,000 postings. There are 824 classes (occupations). For ease of interpretation, the axis ticks represent only the 23 major occupation groups, based on the Standard Occupational Classification (SOC) system.

Figure 3: Overall Entropy by Model Year and Prediction Year

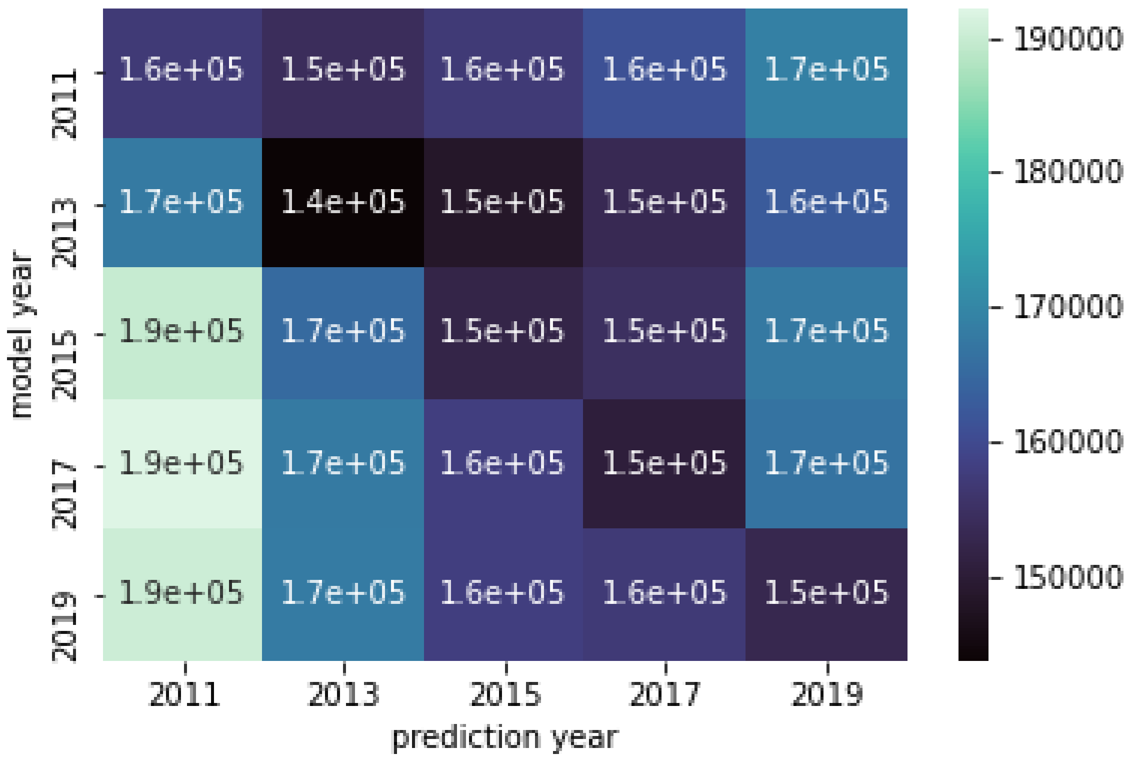
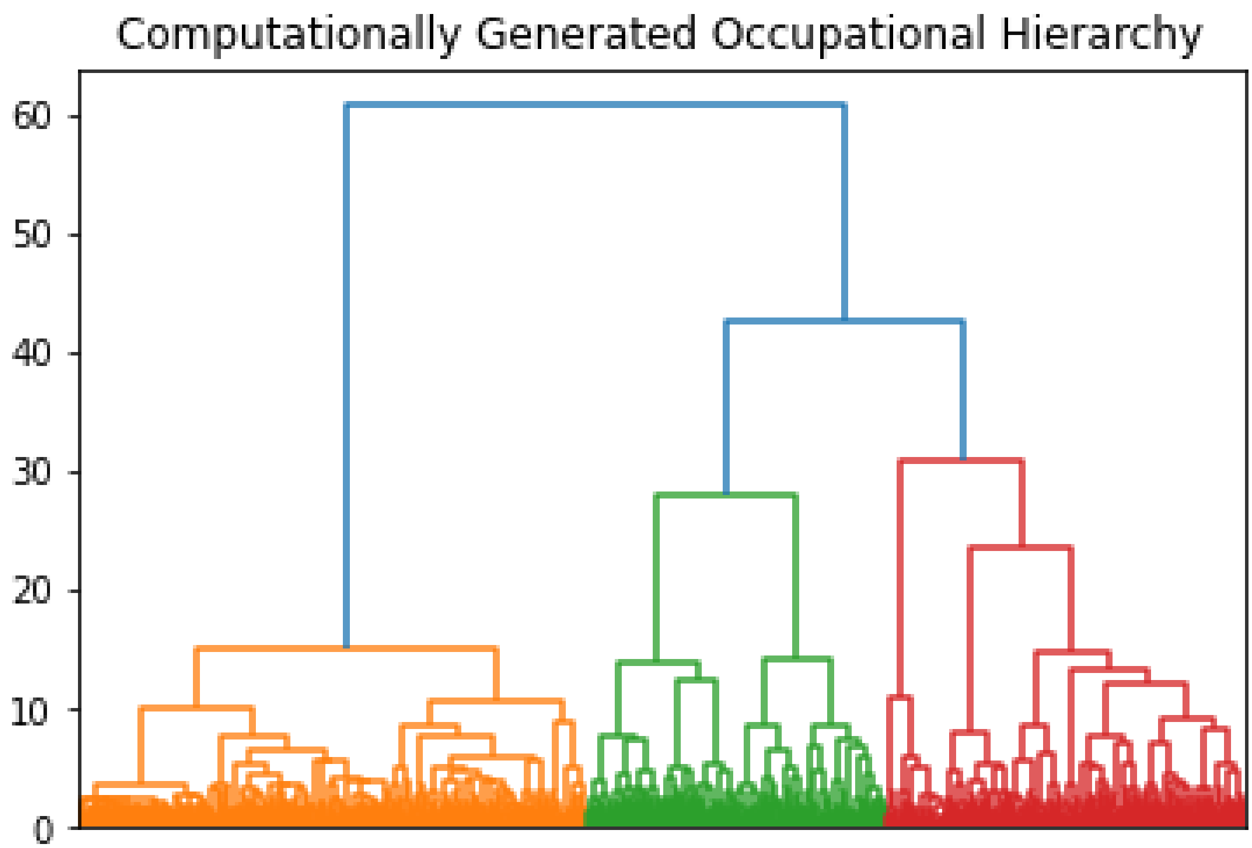
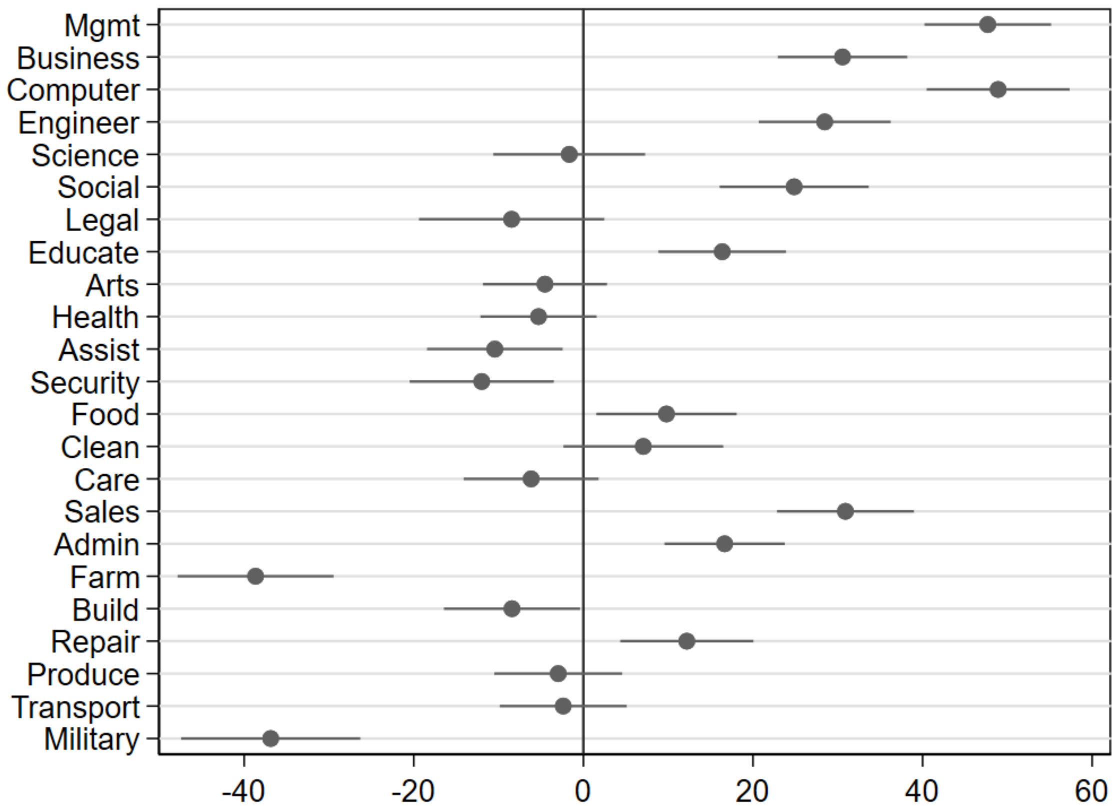


Figure 4: Dendrogram of Agglomerative Clustering of Prediction Similarities, 2019 Data



Notes: Dendrogram of SOC 2-digit parallel system learned from prediction data on 180 thousand 2019 postings. Each posting's label is predicted by the model and returned as a vector of probabilities over all possible labels. Labels are then grouped by Burning Glass tag and averaged over the probability vectors within occupational tag groupings. Subsequently the pairwise cosine distance between each occupation tag grouping with the each other is calculated and used to generate an agglomerative clustering. We use Ward linkage to target variance explained by the clusterings across probability vectors.

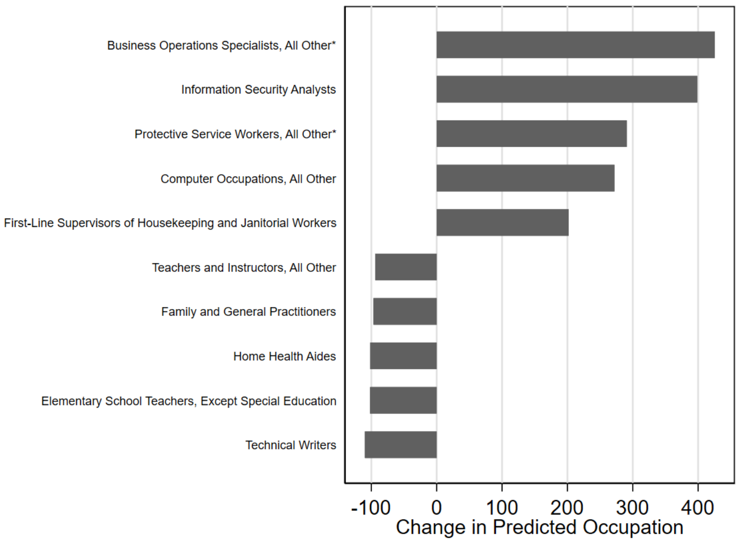
Figure 5: Change in Entropy Over Time



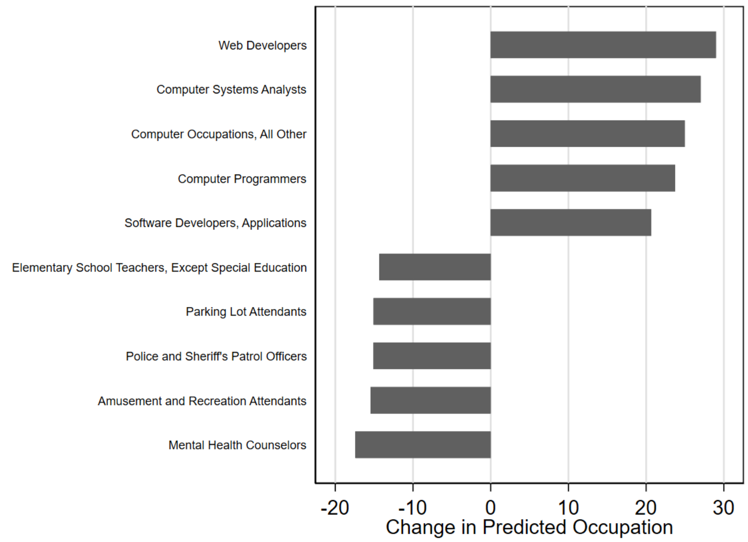
Notes: Entropy is defined as in Section 4. Each category represents one of 23 two digit Standard Occupation Classification major group. Lines denote 95 percent confidence intervals.

Figure 6: Most Frequently Predicted and Least Frequently Predicted Occupations: Results from Text Injection Experiments

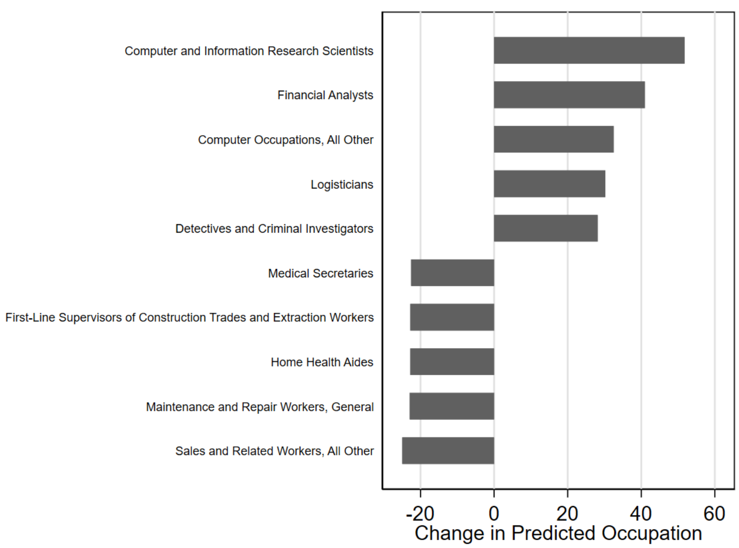
(a) Security Vulnerability Management and Cyber Threat Intelligence



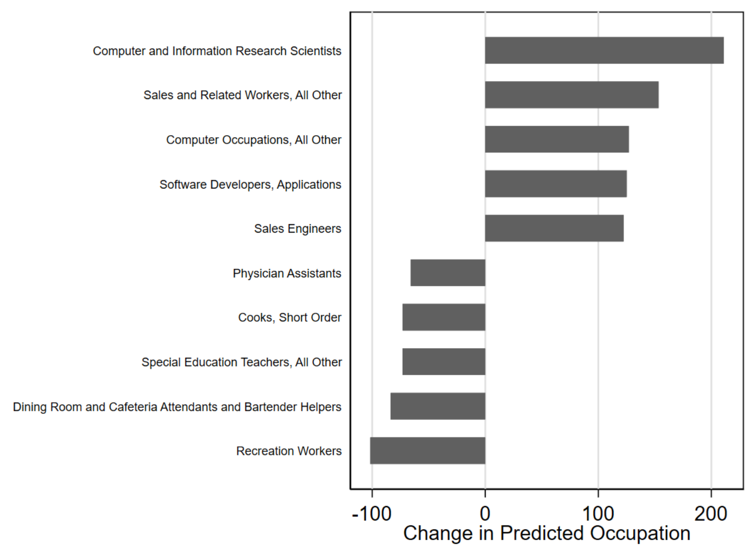
(b) TensorFlow, Pytorch, Keras, scikit-learn, NLP



(c) Required: Masters degree Preferred: Ph.D. in Computer Science or Statistics



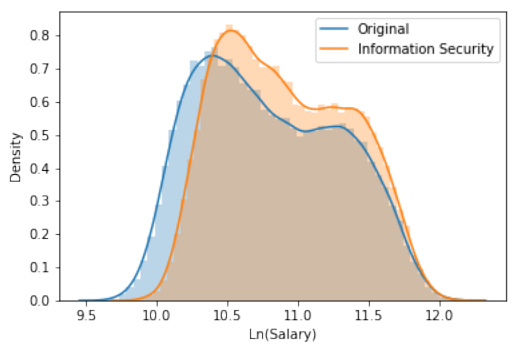
(d) Design SQL, R Python scripts for sales



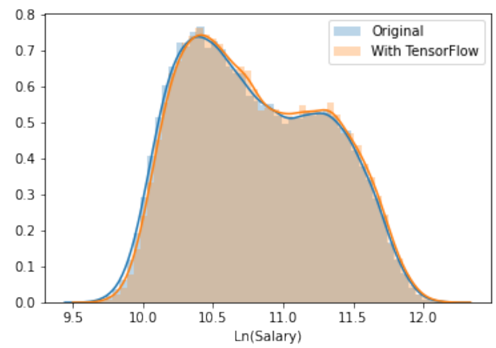
Notes: Sum of the change in predicted probabilities across 50,000 job postings when the following text is added. Only the five occupations that have the greatest change towards and greatest change against are highlighted.
(a) Text injected: "responsible for shaping company policy, advising IT Executive Staff on current and potential threats, monitoring and maintaining a proactive security posture, and rigorous testing and risk management. This includes but is not limited to Network Monitoring, Preventative and Detection Controls, Forensics and Investigations, Security Awareness, Security Vulnerability Management, and Cyber Threat Intelligence activities." (b) Text injected: "TensorFlow, Pytorch, Keras, scikit-learn, NLP" (c) Text injected: "Required: Masters degree Preferred: Ph.D. in Computer Science or Statistics" (d) Text injected: "Design SQL, R Python scripts to directly query and analyze internal and external data sources to package together clear, actionable findings. Aid in the development of predictive modeling and machine learning solutions that continually improve with the collection of real-time dealer and salesperson results from product portfolio".

Figure 7: Returns to IT Skills: Results from Text Injection Experiments

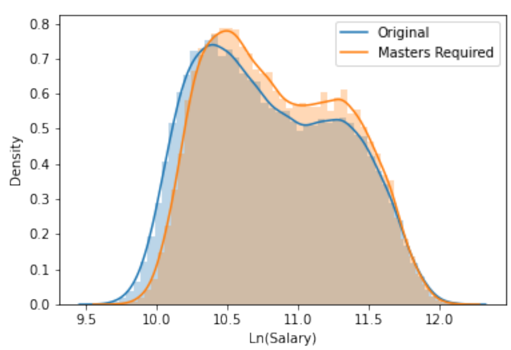
(a) Security Vulnerability Management and Cyber Threat Intelligence



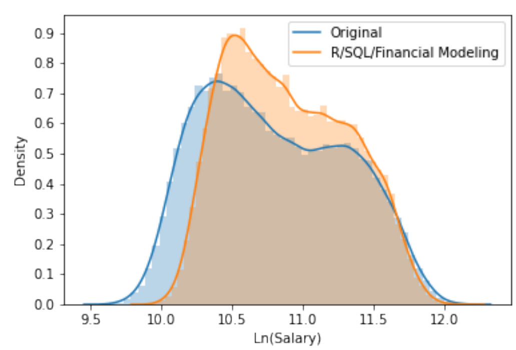
(b) TensorFlow, Pytorch, Keras, scikit-learn, NLP



(c) Required: Masters degree Preferred: Ph.D. in Computer Science or Statistics



(d) Design SQL, R & Python scripts for sales



Notes: Histogram plots of the predicted salary distribution across 40,631 job postings when the following text is added.
(a) Text injected: "responsible for shaping company policy, advising IT Executive Staff on current and potential threats, monitoring and maintaining a proactive security posture, and rigorous testing and risk management. This includes but is not limited to Network Monitoring, Preventative and Detection Controls, Forensics and Investigations, Security Awareness, Security Vulnerability Management, and Cyber Threat Intelligence activities." (b) Text injected: "TensorFlow, Pytorch, Keras, scikit-learn, NLP" (c) Text injected: "Required: Masters degree Preferred: Ph.D. in Computer Science or Statistics" (d) Text injected: "Design SQL, R & Python scripts to directly query and analyze internal and external data sources to package together clear, actionable findings. Aid in the development of predictive modeling and machine learning solutions that continually improve with the collection of real-time dealer and salesperson results from product portfolio".
Statistics on the difference: (a) Mean: 0.115 (s.e. 0.00052) (b) Mean: 0.025 (s.e. 0.00021) (c) Mean: 0.063 (s.e. 0.00035) (d) Mean: 0.096 (s.e. 0.00063)

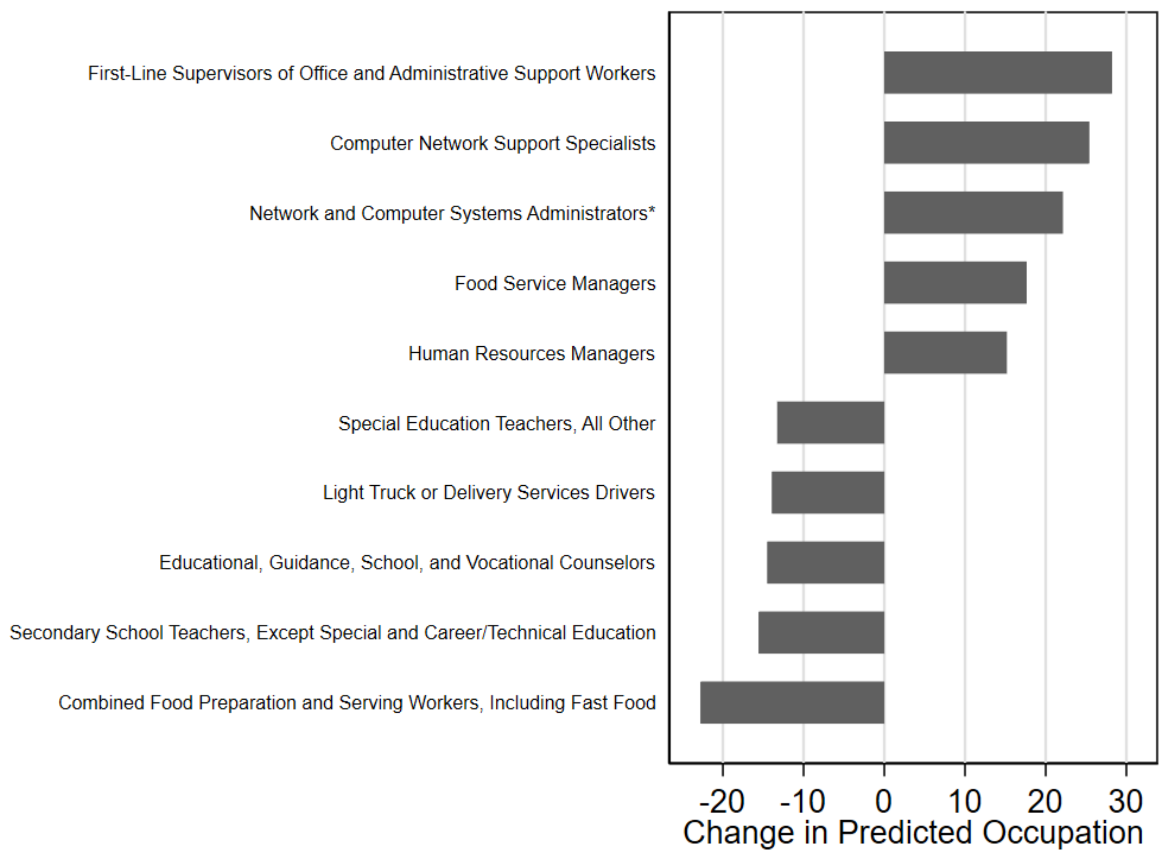


Figure 8: Sum of the change in predicted probabilities across 50,000 job postings when the following text is added: “This is a full time remote position, and employees can be based anywhere in the United States.” Only the five occupations that have the greatest change towards and greatest change against are highlighted.

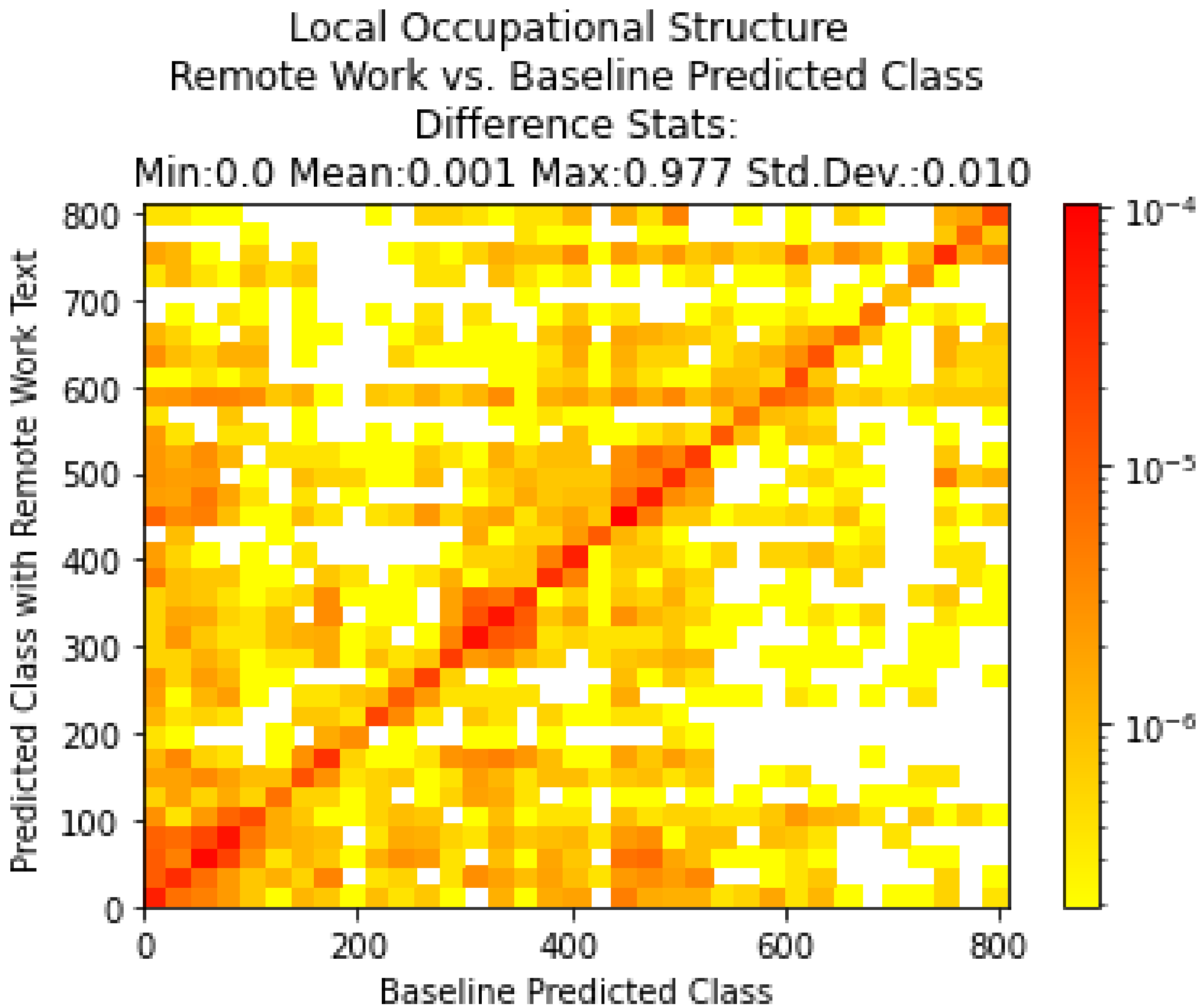


Figure 9: Two-way histogram of baseline vs text-injected posting probabilities. The injected text is: “This is a full time remote position, and employees can be based anywhere in the United States.’

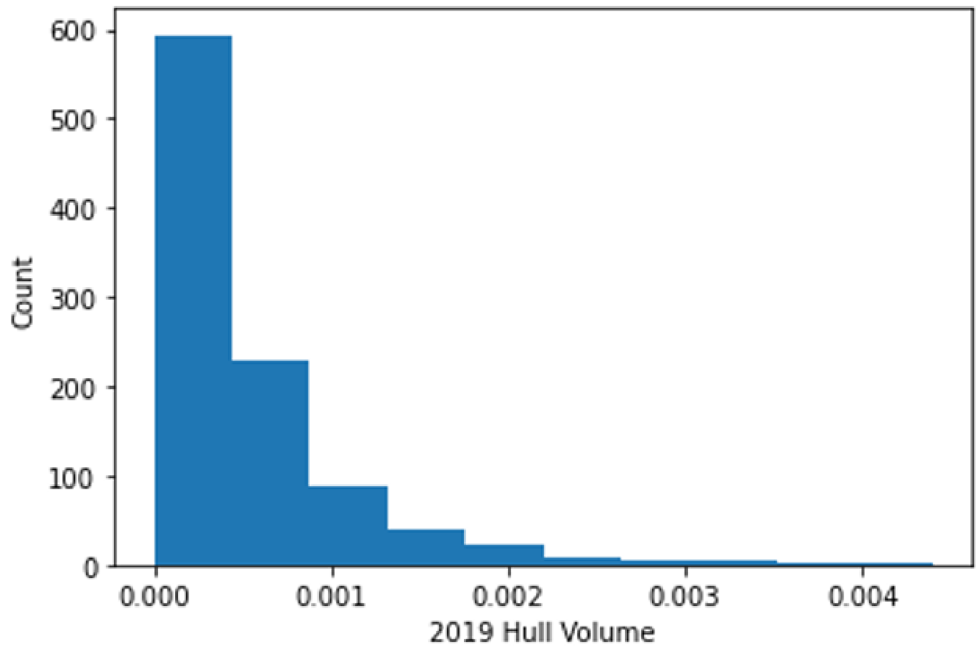


Figure 10: Histogram of the volumes of the convex hull draws for 2019 calculations.

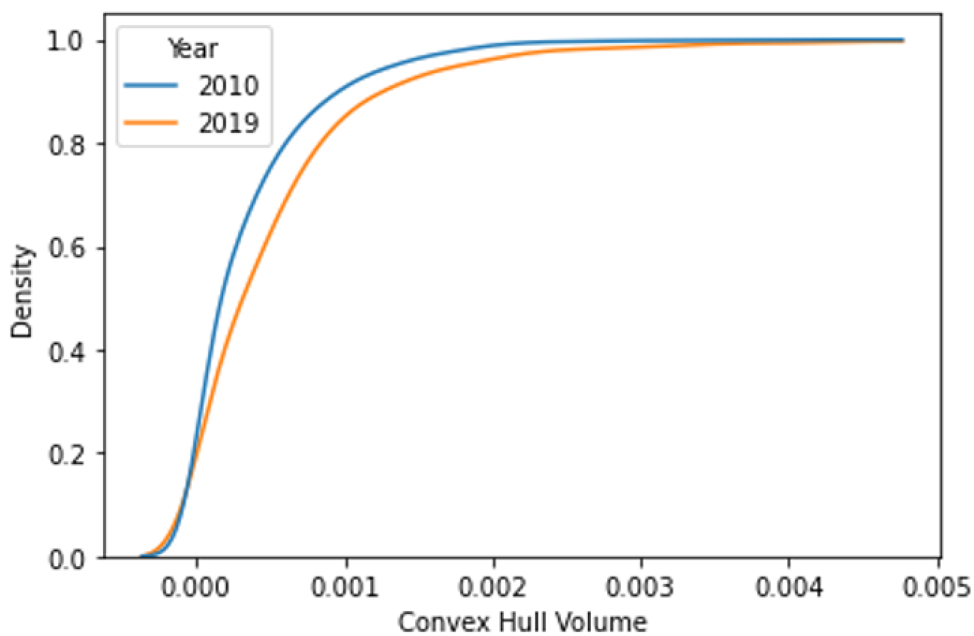


Figure 11: Cumulative Density function for bootstrapped draws of the volumes of the convex hull of the job space.

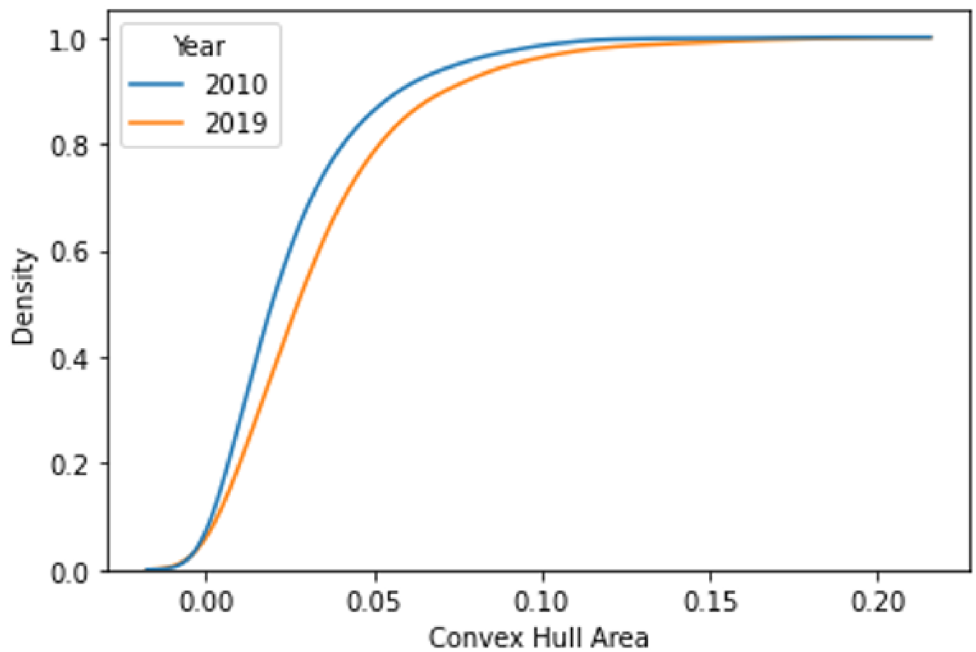


Figure 12: Cumulative Density function for bootstrapped draws of the areas of the convex hull of the job space.

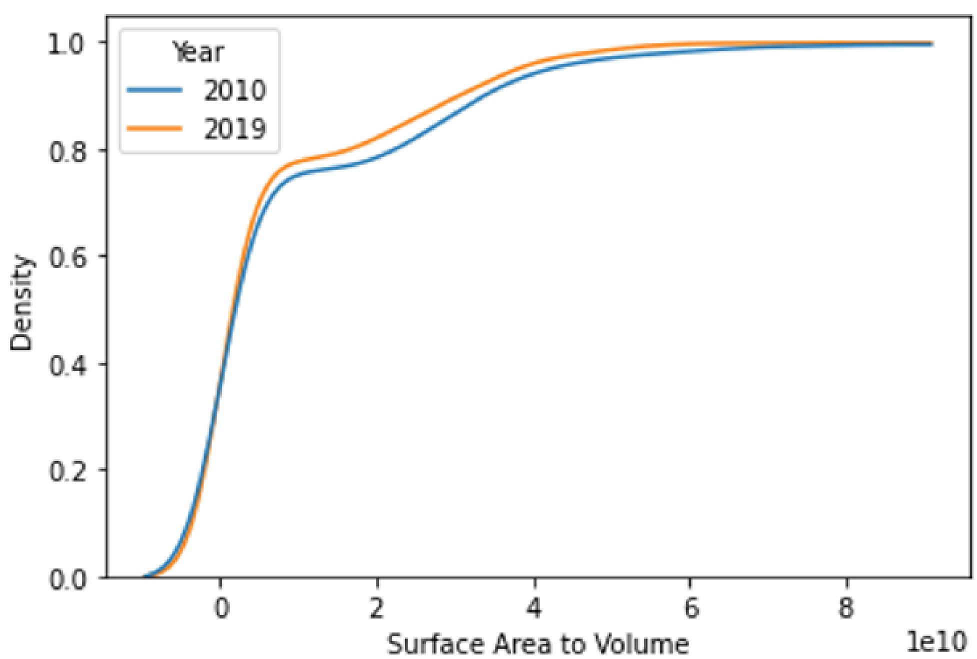


Figure 13: Cumulative Density function for bootstrapped draws of the surface area to volume ratio of the convex hull of the job space.