Vector Representations of Legal Belief

Elliott Ash and Daniel L. Chen*

February 12, 2018

Preliminary version: Comments welcome

Abstract

Recent work in natural language processing represents language objects (words and documents) as dense vectors that encode the relations between those objects (Blei, 2012; Mikolov et al., 2013). These methods have recently been adapted to the analysis of human social behavior (e.g. Caliskan et al., 2017). This paper explores the vectorization of legal beliefs, with the goal of understanding judicial reasoning and the causal impacts of law. We illustrate the usefulness of these vectors in three ways. First, we show that they recover intuitive institutional connections between judges. Second, we show the vectors can be used as features in a decision prediction task. Third, we show that they can be used to measure implicit bias by judges toward women and racial minorities.

1 Introduction

Recent work in computational linguistics has made breakthroughts in the representation of language as vectors (Blei, 2012; Mikolov et al., 2013). The success of Google's popular Word2Vec algorithm is that it "learns" the conceptual relations between words; a trained model can produce synonyms,

*Elliott Ash, Assistant Professor of Economics, University of Warwick, e.ash@warwick.ac.uk; Daniel L. Chen, Professor of Economics, University of Toulouse, daniel.chen@iast.fr. We thank Brenton Arnaboldi, Matthew Willian, and Lihan Yao for research assistance.

antonyms, and analogies for any given word (Mikolov et al., 2013). These "word embeddings," as the word vectors have come to be called, serve well as features in down-stream prediction tasks.

This paper brings the idea of word vectors into the law. We discuss recent work seeking to build "belief vectors" for judges based on the opinions they have written. In the same way that word embeddings encode relations between words, judge embeddings are designed to encode relations between judges. This is a somewhat different task from word or document vectors, because it can use richer representations of judge characteristics besides their language, including the directions of their decisions and their citations to previous opinions.

To illustrate, a word embedding can identify similar words in the vocabulary. For example, the closest word to "judge" might be "jury." A judge embedding can identify similar judges in the legal system. For example, the closest judge to Antonin Scalia might be Clarence Thomas.

A more intriguing exercise is to think about analogies. A functional word embedding would be able to say that "governor" is to "state" as "mayor" is to "city," through the vector algebra governor - state + city = mayor. Similarly, a judge embedding could say something like Scalia is to Thomas as Ginsburg is to Breyer, in the sense that Scalia - Thomas + Breyer = Ginsburg.

In the analyis below, we use our belief vectors as inputs for a set of supervised and unsupervised machine learning tasks. We argue that the information recovered by our model provides a meaningful signal about a judge's legal beliefs. Rather than seeking to achieve optimal performance on a particular machine learning task, we try to demonstrate versatility across a wide variety of learning tasks. Our hope is that the techniques we have developed can be used as one of many inputs for building end-to-end machine learning models.

2 Vectorization of Legal Objects

This section describes an approach for representing legal objects as vectors. Consider the following model of judical opinion generation. The unit of observation is an opinion i, written by judge j at time t in court/jurisdiction c. The opinion is a matrix of features Y_i, including the ruling (affirm/reverse), the text features of the opinion, and the set of citations to previous opinions. The case is a review of a district court opinion, represented by a vector of

features D_i, including the text and metadata from the district court. One could also include a vector that embeds the stock of precedents in court c at time t, but a simpler approach is to include jurisdiction-time fixed effects.

We are interested in understanding

$$Y_i \sim F(D_i, X_j, c, t)$$

where $F(\cdot)$ is some distribution over opinion features we can approximate using deep neural nets or some other machine learning procedure (e.g. Hartford et al., 2016).

One could further enrich the model in a few obvious ways. First, one could allow for time varying judge characteristics X_{jt}. Second, one could allow for impacts of other judges on the panel, p.

This model could then be used to simulate counterfactuals. For example, how would the decision in a case change by switching out the authoring judge j? How would the style of language change for a different circuit c?

3 Applications to Federal Appellate Courts

3.1 Data

This paper utilizes a dataset collected by one of the authors on all 380,000 cases and a million judge votes in Circuit Courts. We process the text of the written opinions to represent each opinion as a vector of phrases. We further make use of a large set of biographical features of the 268 judges in our sample, as well as 400 hand-coded features in a 5% random sample of cases, and 6000 cases hand-coded for meaning in 25 legal areas. The latter two data sets help serve to validate that we can begin to textually measure what is salient in the reasoning of written opinions.[1]

To build our judge vectors, we took 212,101 opinions from 1,106 unique judges in U.S. Circuit Court cases from 1970-2013. We excluded judges with fewer than 5 opinions. Of the 212,101 opinions, 12,930 were discretionary

[1]For example, phrases like, "influenc outcom vote" and "disclosur sourc", predict liberal campaign finance decisions, while phrases like, "inform elector" and "buckley court limit", predict conservative decisions. Buckley held that limits on election spending are unconstitutional. For capital punishment, liberal decisions use "involuntari" and "mental health" while conservative decisions use "attack". For the EPA, liberal decisions refer to "hazard wast" and conservative ones refer to "statut silent ambigu".

opinions. Second, we included 3,647 Supreme Court case opinions from 1970-2013. Because dissents occur much more frequently in the Supreme Court than in appellate courts, we took special care to parse the text between majority and dissenting opinions. Overall, there were 2,315 majority opinions and 1,332 dissents, so the distribution (63.5 percent to 36.5 percent) was much more balanced.

3.2 Vectorization and Prediction using Opinions

First, we load in the opinion corpus and acquire vector representations of opinion texts using the doc2vec model (Le and Mikolov, 2014), implemented in the Python package gensim. As a preliminary test of the informativeness of these vectors, we used them as inputs into a case-level prediction task. Inputting document vectors into a Support Vector Machine with Radial Basis Kernel, we concluded parameter search after achieving a classification accuracy of 70.5%. To build the document vectors, we chose the distributed bag-of-words model over the distributed memory model, with 200 dimensions per document vector. Other parameter choices include a sliding window of size 20, using words appearing at least 50 times, and documents above 40 words in length. The sliding window parameter determines the length of the preceding word context for the word prediction task in the Doc2Vec model.

3.3 Judge Vectors

In vein of recent works about temporal dynamics of language (Rudolph and Blei, 2017), our model constructs vectors for each judge over a finite time window. For most of our experiments, the time window was one year. In our experiments with the Supreme Court cases, we instead used a time window of 5 years in order to deal with the fact that a Supreme Court justice writes relatively fewer opinions in a year compared to an Appellate Court justice.

One of the constraints when building our model is that we wanted all of the $(judge, year)$ tuples to be close to one another for each judge. While evaluating this experiment, we found that topic normalization has led to greater consistency of judge-year tuples being close for the same judge. With our initial (j, y) tuples, within ten closest neighbors of a given tuple (j, y), there are on average 1.6 tuples which belong to the same judge j.

The appellate court case dataset is organized into 8 topic categories. Each case is labelled with up to two of those categories. To construct the judge

vectors, we first compute a topic vector for each year, $t_{\alpha_y} = \frac{1}{N}\sum_{i=1}^{N} d_{y_i}$ where $T_y = d_{y_1}, d_{y_2}, \cdots$ are all of the documents associated with a given topic and $|D_t| = N$. From there we construct the topic normalized document vectors for each topic. Let d_i be a document in our corpus with topics α and β from year y. Then the topic-normalized document vector

$$\hat{d_i} = d_i - \frac{t_{\alpha_y} + t_{\beta_y}}{2} \tag{3.1}$$

In the case where a document only has a single topic α,

$$\hat{d_i} = d_i - t_{\alpha_y} \tag{3.2}$$

To construct a vector for judge x in year y, j_{x_y} we simply take the mean of all the topic-normalized document vectors written by that judge in a given year.

For this experiment, since opinions which belong to the same topic are more likely to be similar, we found topic normalization to be helpful. The construction of topic vectors is detailed above. For each document vector, we subtract the topic vector corresponding to its primary topic. For judge vectors constructed with these topic-normalized document vectors, Within ten closest neighbors of a given tuple (j, y), there are on average 2.8 tuples belonging to judge j.

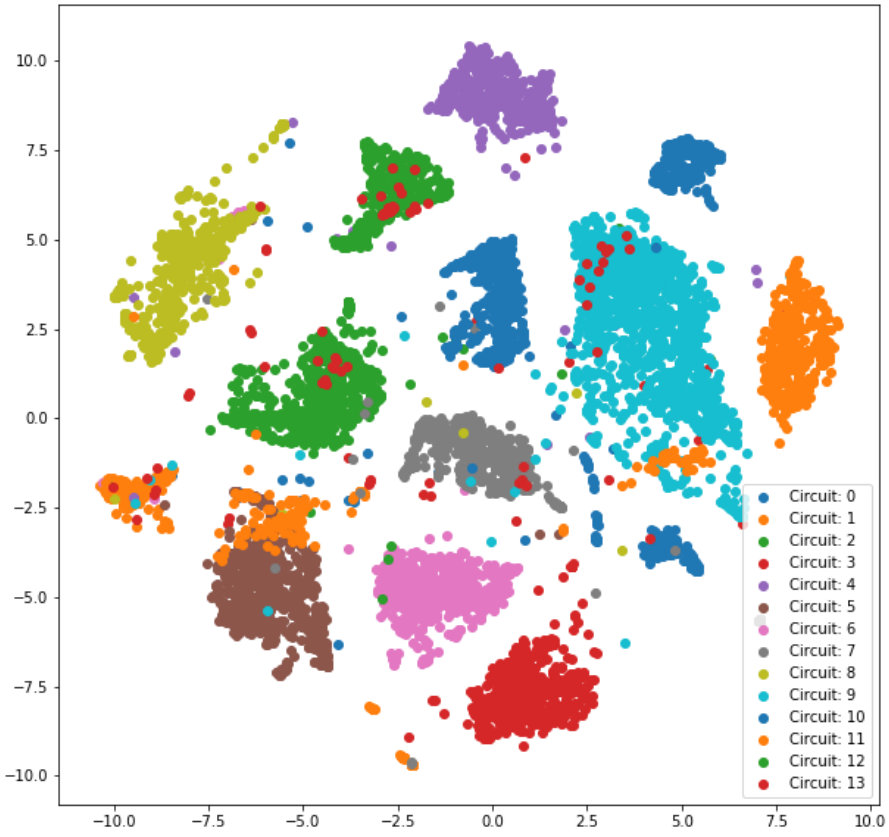
3.4 Visual Structure of Judge Vectors

In Figure 3.1, we visualize the yearly judge vectors using a t-SNE plot (Maaten and Hinton, 2008). This plot projects judge-year vectors down to two dimensions for visualization purposes. Each dot is a judge-year, and the coloring is given by the circuits. Circuit 0 is the DC Circuit, Circuit 12 is the Federal Circuit, and Circuit 13 is the U.S. Supreme Court.

This plot demonstrated a handful of interesting features. First, it is clear that the circuit in which the judges reside is a highly significant component of our belief model. Normalizing this factor away would make sense in improvements to the model.

One possible approach for eliminating this variability would be to apply substantial pre-processing to the text before feeding it into our model, controlling for circuit specific features. Possible approaches here include replacing those n-grams that have high tf-idf scores in certain circuits, and

Figure 3.1: t-SNE plot of the topic-normalized judge vectors



removing person or location references in the text corpus via named entity recognition.

Another approach would be to do the transformation at the circuit level by computing vectors representative of each circuit, then subtracting it from the judge vectors. The core difficulty with the vector subtraction approach is building a vector for the geographic component of the vectors while separating circuit-specific components of judicial belief. Alternatively, we may control for circuit/topic components of each judge vector by subtracting only circuit/topic projections while at the document level (each document's vector that are orthogonal to the projection is then preserved).

Second, it is interesting that there is much more spread of Supreme Court judges (red dots) into other clusters. Ideology might be more diverse (less constrained) in the Supreme Court than the Circuit Courts. That said, we still found that the Supreme Court does cluster together. For example, when we tried to assess which judges had vectors closest to Ruth Bader Ginsburg, the list was dominated by Supreme Court justices, regardless of their ideology. For example, Scalia appears near the very top of the similarity ranking with Ginsburg, which does not make sense to lawyers and legal scholars familiar with their ideological differences.

We tried to fix this issue by removing certain words from the opinion texts (such as "Justice", "Ginsburg", "Scalia", etc.), as well as some unusual footnote labels, but it still was not enough. The structure and format of the Supreme Court texts is different from the appellate court texts, and an important next step is to correct for that.

Third, note that the 11th Circuit (orange dots) is split into multiple clusters. And in particular note that these judges overlap with the 5th Circuit (brown dots). This makes good sense, as the 11th Circuit split off from the 5th Circuit in 1982 and uses pre-1982 5th Circuit cases as precedent.

3.5 Vectorization of Citation Networks

As noted, we had trouble recovering the distinctions between Ginsburg and Scalia using just a text-based metric. One way to address this issue is to incorporate information from the citation graph. The idea is that while Ginsburg and Scalia may use similar language in their opinions due to the institutional context of the Supreme Court, the set of precedents they select will reflect differences in legal beliefs.

We use a promising method for this type of task, node2vec (Grover and

Leskovec, 2016). This is a scalable, micro-and-macro-structure-aware approach to network representation that encodes the graph as an $\mathbb{R}^d \times |V|$ matrix, where $|V|$ is the cardinality of the graph vertices, and d is the output vector dimension of a node's vector representation.

The 5.8 million citations in the 1880-2013 opinion corpus constitute the edges of the citation graph G. The node2vec algorithm works by completing random walks in order to discover underlying node relationships. As parameters, we select 128 output dimensions, 16 workers, 7 SGD iterations per walk, 50 walks, and 80 nodes visited per walk.

3.6 Decision Prediction Using Language and Citation Vectors

By document-wise concatenation, we integrate our citation vector with our document vector, derived from node2vec and doc2vec models respectively. In this way, the vector representation of a document reflects its position in the citation graph, as well as its textual structure. With normalization, this representation is ready for document tasks. We evaluate the performance of constituent and concatenated vectors with Support Vector Machine, Random Forest, and Logistic Regression models.

The overall best performing model across input vectors is Support Vector Machine with Radial Basis Function (RBF) kernel. This is a noticeable improvement over the majority classifier (at 52% accuracy). Though the citation vectors have performed consistently across parameters, its citation features, when added to document vectors, have slightly worsened the performance of the concatenated vector (at a total of 328 dimensions). With certain C parameters and the Polynomial kernel (bold), we notice slight improvements by concatenation.

A possible direction for better concatenated vector performance may be in dimension reduction. Techniques such as fewer output dimensions during node2vec and doc2vec training, or inputting high dimensional vectors into a neural network, should be considered.

Table 1: Accuracy of Pro/Anti Gov classification amongst regulation cases, case level classification

Kernel, C param	Docs Only	Citations Only	Doc + Citation
RBF 1	0.678	0.635	0.670
RBF 2	0.688	0.638	0.680
RBF 2.5	0.693	0.639	0.680
RBF 3	0.697	0.638	0.681
RBF 4	0.705	0.633	0.687
Poly 1	0.521	0.618	0.537
Poly 2	0.538	0.628	0.632
Poly 2.5	0.600	0.629	0.637
Poly 3	0.639	0.632	0.639
Poly 4	0.651	0.633	0.645

Table 2: Accuracy of Pro/Anti Gov classification w/ Random Forest

Number of Trees	Docs Only	Citations Only	Doc + Citation
5	0.601	0.588	0.597
10	0.605	0.620	0.613
20	0.660	0.609	0.647
30	0.657	0.616	0.641

Table 3: Sentiment and Target Words for Measuring Implicit Language Bias

Sentiment Attribute Words	
"pleasant" ($\vec{w}_+$)	"unpleasant" ($\vec{w}_-$)
joy, love, peace, wonderful, pleasure, friend, laughter, happy	agony, terrible, horrible, nasty, evil, war, awful, failure

Implicit Sexism Target Words (X and Y)	
"male" ($\vec{w}_M$)	"female" ($\vec{w}_F$)
male, man, boy, brother, he, him, his, son	female, woman, girl, sister, she, her, hers, daughter

Implicit Racism Target Words (X and Y)	
"white" ($\vec{w}_W$)	"black" ($\vec{w}_B$)
european, white, caucasian	black, african, negro

3.7 Language-Based Metrics of Implicit Bias

A final application of these ideas is to follow in the steps of Caliskan et al. (2017) to measure implicit bias in the text of judicial opinions. Caliskan et al. (2017) start with an off-the-shelf word embeddings model, which as discussed represents words as vectors in a 300-dimensional geometric space. They then compute similarity, which means having the same direction in the word vector space, between groups of words.

We start with a set of attribute words, where A is a set of words with positive sentiment, and B is a set of words with negative sentiment. These are listed in the top panel of Table 3. Next, we have a set of target words, where X represents one social group and Y represents another social group. The group words for measuring gender bias are listed in the middle panel of Table 3, while the group words for measuring racial bias are listed in the bottom panel. We isolate the direction for "pleasant" ($\vec{w}_+$), "unpleasant" ($\vec{w}_-$), "male" ($\vec{w}_M$), "female" ($\vec{w}_F$), "white" ($\vec{w}_W$), and "black" ($\vec{w}_B$), respectively, by taking the average vector direction of the words in each group.

Then, implicit language association is measured by the cosine similarity between the averaged vectors,

$$s(\vec{v}, \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{\|\vec{v}\| \, \|\vec{w}\|}$$

which is equal to one minus the cosine of the angle between the vectors. It takes a value between -1 and 1, with similarities approaching one meaning that the words co-occur together relatively often, and similarities approaching -1 meaning the words rarely co-occur. Similarites near zero mean that the words appear together as often as two randomly selected words from the vocabulary.

The metrics for implicit language bias are given by

$$\begin{aligned}\text{Implicit Gender Bias} &= \frac{\text{Male-Pleasant Association}}{\text{Male-Unpleasant Association}} - \frac{\text{Female-Pleasant Association}}{\text{Female-Unleasant Association}} \\ &= \frac{s(\vec{w}_M, \vec{w}_+)}{s(\vec{w}_M, \vec{w}_-)} - \frac{s(\vec{w}_F, \vec{w}_+)}{s(\vec{w}_F, \vec{w}_-)}\end{aligned}$$

$$\begin{aligned}\text{Implicit Racial Bias} &= \frac{\text{White-Pleasant Association}}{\text{White-Unpleasant Association}} - \frac{\text{Black-Pleasant Association}}{\text{Black-Unleasant Association}} \\ &= \frac{s(\vec{w}_W, \vec{w}_+)}{s(\vec{w}_W, \vec{w}_-)} - \frac{s(\vec{w}_B, \vec{w}_+)}{s(\vec{w}_B, \vec{w}_-)}\end{aligned}$$

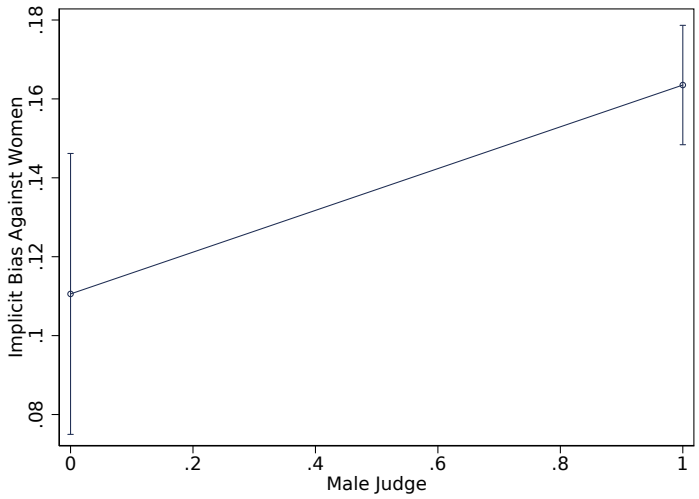
We computed $s(\cdot)$ and the corresponding bias measures for each judge in our corpus of opinions for the years 1970-2013. We constructed an auxiliary corpus of opinions for all of the judges on President Trump's Supreme Court nominees list, with seventeen judges total. Here we provide some initial descriptive analysis.

Figure 3 reports differences between judges on the basic metrics, based on the gender and race of the judge. In panel a, we see that male judges have higher implicit gender bias (that is, implicit male preference) than female judges. In panel b, we see that white judges have *lower* implicit racial bias (that is, implicit white-race preference) than non-white judges.

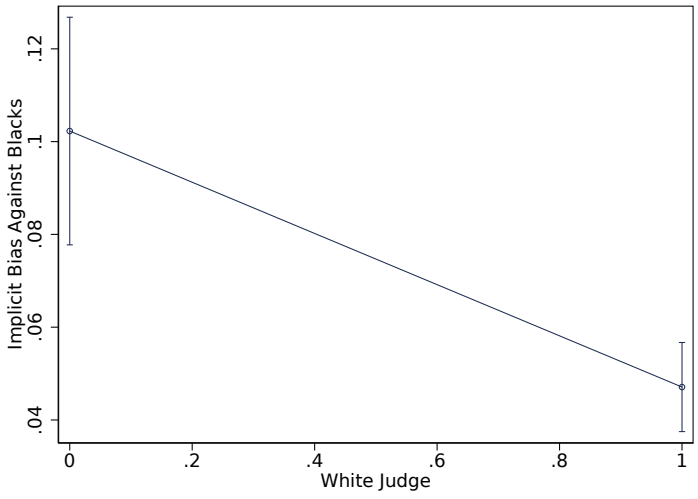
Figure 3.3 analyzes the descriptive relation between implicit bias and political variables. Panel A shows that relative to Democrat judges, Republican judges demonstrate higher implicit gender bias and higher implicit race bias. Panel B shows that relative to the rest of the population of circuit judges, Trump shortlist judges are about the same for gender bias, but score higher on implicit racial bias.

These are preliminary results and additional work is needed to assess their robustness. In future work, one can use richer vector representations (that use the citation network, for example) of judicial attitudes to develop more refined measures of implicit bias in the judiciary.

Figure 3.2: Differences in Implicit Bias of Circuit Judges, by Gender and Race

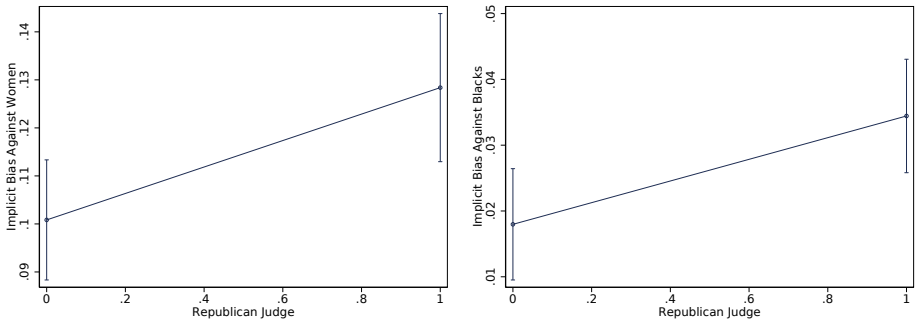


(a) Differences in Implicit Bias, Female vs. Male Judges

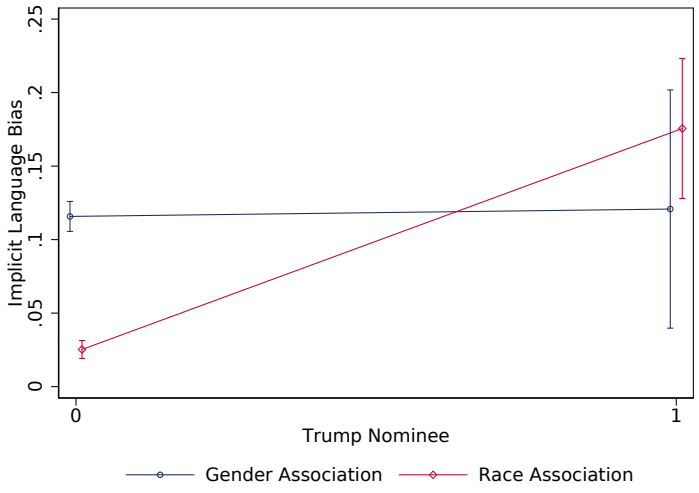


(b) Differences in Implicit Bias, Non-White vs. White Judges

Figure 3.3: Differences in Implicit Bias of Circuit Judges, by Political Party and Trump Shortlist Status



(a) Differences in Implicit Bias, Democrat vs. Republican Judges



(b) Differences in Implicit Bias, Other Judges vs. Trump Shortlist

References

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, 55(4):77–84. (document), 1

Caliskan, A., Bryson, J. J., and Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186. (document), 3.7

Grover, A. and Leskovec, J. (2016). Node2vec: Scalable feature learning for networks. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 855–864, New York, NY, USA. ACM. 3.5

Hartford, J., Lewis, G., Leyton-Brown, K., and Taddy, M. (2016). Counterfactual prediction with deep instrumental variables networks. *arXiv preprint arXiv:1612.09596*. 2

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*. 3.2

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605. 3.4

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119. (document), 1

Rudolph, M. and Blei, D. (2017). Dynamic bernoulli embeddings for language evolution. *arXiv preprint arXiv:1703.08052*. 3.3