**APPENDIX A: Some Methodological Notes on Computational Text Analysis**

This methodological appendix provides further detail on the different components of the computational text analysis presented in the paper.

***A.1 Pre-processing***

The corpus contains 9,842 transcripts: 4,498 from before the *Lawrence* and *Shaputis* California Supreme Court rulings and 5,344 from after. This includes transcripts for every hearing held between January 1, 2007 and March 31, 2010 that resulted in a grant or denial of parole. I received the transcripts in PDF form directly from the California Board of Parole Hearings. I converted the PDFs to plain text and removed formatting such as line numbers, page numbers, and headings. I then ran the transcripts through the NLTK tokenizer package in Python to separate each word from surrounding whitespace and punctuation. I used consistent textual markers to split each transcript into two sections: the ‘proceedings’ where commissioners question potential parolees and review relevant materials, and the ‘decision’ where the commissioners announce and explain their parole decision. The subsequent computational analysis focuses on the decisions, which contain 25,419,909 tokens (this is primarily words, but includes punctuation). I opted against common techniques of lemmatizing or stemming words because I wanted to preserve the specificity of word use – for example, how ‘insightful’ may be different that ‘insight’ in the context of the hearings. I also retained common ‘stop words’ (such as ‘and’, ‘but’, and ‘the’) because for word embedding models, ‘stop words’ provide significant information about word use and context.

***A.2 Word Frequencies***

To analyze word frequencies, I wrote a Python script to count both whether and how many times a target word appears in the transcripts. For each month in the study period, I calculated the percentage of transcripts where the target word appeared. I used this to create **Figure 1**. I completed a similar analysis, except counting how many sentences within a transcript that contain the target word, to create **Figure 3**. Recent research using massive corpora suggests that word frequency can be a powerful indicator of linguistic and cultural change (Michel et al. 2011; Pagel, Atkinson, and Meade 2007). Unlike those studies, which used millions of heterogeneous texts, my approach is situated within a specific organizational context, which allowed me to link words to hearing outcomes. I used the frequencies to analyze how word use varied with decisions before and after the rulings. **Table 1** presented these results.

***A.3 Word Embeddings***

I follow recent advances in using word embeddings to look at changing language use over time (Hamilton et al. 2016a; 2016b; Tahmasebi, Borin, and Jatowt 2019). To create the word embeddings, I used the word2vec algorithm in the Gensim Python package (Rehurek and Sojka 2010). Word2vec is an algorithm that uses machine learning techniques to represent words in a multi-dimensional vector space (Mikolov et al. 2013). The approach I use in this paper trains a neural network to predict a word’s context within a defined window, an approach called skip-gram with negative sampling (SGNS) (Goldberg and Levy 2014). To illustrate the basic mechanics, if we take the word ‘network’ from the previous sentence as our target word and look at a context window of 2, the word’s context would be: “a neural \_\_\_\_ to predict”. In turn, the algorithm is trained to predict the context words ‘a,’ ‘neural,’ ‘to,’ and ‘predict’ given the target word ‘network’. The algorithm uses a shallow neural network for this task, starting with randomly initialized coefficients in a fixed-length “hidden layer” between input and output. Given the prediction task, the algorithm finds an optimal configuration of coefficients in the hidden layer for each word in the vocabulary. When I refer to word embeddings, I am referring to these vectors of coefficients that uniquely describe each word in the vocabulary of the corpus. (For more on word embeddings see Chapter 6 in Jurafsky and Martin (2020), and on specifically word2vec see Chapter 6 section 8.)

Implementing SGNS required several important decisions. Hamilton et al. (2016a, 1494) find that, compared with embeddings created through alternative Positive Pointwise Mutual Information and Single Value Decomposition approaches, embeddings created via SGNS tend to fair best on discovering semantic shifts, and are relatively conservative when it comes to identifying shifts in smaller datasets. I experimented with the width of the context window and the length of the hidden layer, as these are consequential modelling parameters (Spirling and Rodriguez 2019). After testing multiple models, I used a model with a context window of 10 and an embedding layer length of 300 in the final analysis. I compared models of different embedding sizes and context windows with the widely used wordsim353 and simlex word pair test data as a check of the validity of the word embeddings for generic words that do not have unique corpus-specific meanings and uses. All models performed relatively similarly on both tasks. However, scholars have suggested that larger context windows more accurately capture the topical meaning of words, while smaller context windows better reflect syntactic dimensions of words (Chiu, Korhonen, and Pyysalo 2016; Levy, Goldberg, and Dagan 2015). This led me to use a window size of 10 words. Using windows of 5, 15, and 20 words produced substantively similar results to those presented in this paper.

In order to quantify model uncertainty and address issues with model instability, I averaged cosine similarity scores over 50 embedding sets trained on bootstrapped variations of the corpus (Antoniak and Mimno 2018). SGNS models, because they are randomly initialized, can produce random variation across different sets of embeddings trained on the same corpus. Furthermore, for small corpora, individual documents can exercise outsized effects on the embeddings. To counter these issues, I follow Antoniak and Mimno’s (2018) recommendation to ‘bootstrap’ by training multiple sets of embeddings on random samples of the corpus. For each set I first trained a model on the full set of transcripts. I then fit that model to ‘before’ and ‘after’ corpora that were sampled with replacement from the original corpora. My re-sampled corpora had the same number of transcripts as the original corpora (4,498 from before the rulings and 5,344 from after the rulings). The results in **Table 2** report the 15 nearest neighbors by average cosine similarity score across all 50 embedding pairs for words that appear in the vocabulary of every set of embeddings.

To formally compare neighborhood change in the embedding space over time, presented in **Figure 2**, I implemented a technique developed by Hamilton et al. (2016b) that compares the distance from each target word to its 15 closest neighbors across time periods to assess how a word shifts in embedding space. More specifically, I made a combined neighborhood of words that appeared in the 15 nearest neighbors to the target word before *or* after the rulings. For each set of embeddings in a before-after pair, I measured the cosine distance from the target word to each word in the combined neighborhood vocabulary. I then organized these cosine similarity scores into two secondary vectors and calculated the cosine similarity of these secondary vectors as a measure of neighborhood change. I did this for each of the 50 bootstrapped pairs and then averaged across the results to present the means and standard deviations presented in **Figure 2**.

I further used the word embedding model to determine the neighborhoods for ‘insight’, ‘heinous’, and ‘assessment’ for **Figure 3.** I counted how many sentences in each transcript decision contained any word in the neighborhood, divided by the total number of sentences in the transcript, and used this to calculate the monthly averages reported in the figure. As noted in footnote 6, for this analysis I removed the word ‘risk’ from the neighborhood of ‘assessment’ and added the 16th closest neighbor. I did this because the word frequency for ‘risk’ greatly increases following the rulings, related to the court deciding that the commissioners had to make decisions based on a person’s ‘continuing danger and risk to public safety.’ Based on the case study close readings, it appears that the commissioners increasingly began to use this language in the decisions unrelated to the risk assessments. If the increased use of ‘risk’ were tied to assessments, I would expect to see a similar trend in the frequency of the other words in the ‘assessment’ neighborhood. Figure 3 shows that this is not the case.

**APPENDIX B: Further Robustness Checks for Computational Word Analysis**

In this appendix I show that using an alternate approach to building word embeddings provides substantively similar findings to the word2vec approach presented in the main analysis. I also use the word2vec model to further analyze whether commissioners substituted ‘insight’ for other words and provide further evidence that they did not.

**B.1 Another Approach to Word Embeddings: Co-Occurrence Matrices**

In this analysis I created a co-occurrence matrix for the 10,000 most common words in the corpus vocabulary. The matrix contains the counts of how often other words in the vocabulary appear within a 10-word window of the target word, for each of the 10,000 most common words (this uses the same window size as the Word2vec analysis). This created a sparse 10,000-by-10,000-word matrix, where each word is represented by a one-by-10,000-word vector of co-occurrence frequencies. Because some words, such as ‘the’, frequently appear in the context of most words, I further weighted the frequencies by term frequency-inverse document frequency (TF-IDF). TF-IDF is represented by the following equation:

Where *i* is the context word, *j* is the target word, is how many times *i* appears in the 10-word context window of *j*, *N* is the total number of target words (in this case 10,000), and is the number of target words with context word *i* in their context. This is a common technique in information theory for identifying context words that appear frequently in the context of one word but rarely in other contexts (Jurafsky and Martin 2020, Chapter 6). The intuition is that this gives a measure of how distinctive a particular context word is for a particular target word. To analyze how language use changed following the rulings, I create two co-occurrence matrices from the transcripts before and after the rulings.

First, the co-occurrence matrices allow me to see the context words with the highest TF-IDF scores for any given target word in both time periods. For ‘insight,’ the fifteen words with the highest TF-IDF score before the rulings are: *remorse*, *lack*, *into*, *develop*, *gain*, *factors*, *causative*, *crime*, *insight*, *why*, *responsibility*, *clinical*, *understanding*, *level*, and *help*. Following the rulings, the fifteen words with the highest TF-IDF scores for ‘insight’ are: *causative, factors, lack, into, lacks, conduct, remorse, insight, understanding, responsibility, gain, evidenced, crime, why,* and *develop.* Again, these words tend to appear frequently in the contexts where ‘insight’ is discussed in the transcripts, and less frequently in other contexts. The words show substantial overlap with the nearest neighbors presented in Table 2, which suggests that words in the neighborhood of ‘insight’ in the word2vec model may appear not only in *similar* contexts to ‘insight,’ but often the *same* context.

I can further use the co-occurrence matrices to recreate the neighborhood change analysis presented in the paper. As with the word2vec neighborhood change analysis, for each target words (‘insight’, ‘lawrence’, and ‘mother’) I first take the 15 context words with the highest TF-IDF scores for each time period to create a combined neighborhood. I then create a secondary vector for each time period with the TF-IDF scores of each context word in the combined neighborhood. I next take the cosine similarity score of these two secondary vectors as a measure of how much a word’s context has shifted between the two time periods. This creates a measure with greater spread than the word2vec model, but similar substantive results: ‘lawrence’ has a score of 0.356, ‘insight’ has a score of 0.794, and ‘mother’ has a score of 0.971. For reference, the 1,000 most-frequent words in the transcripts have an average score of 0.800. This suggests that the context of ‘insight’ shifted about as much as the 1,000 most-used words on average before and after the rulings. Meanwhile, the results provide further evidence that ‘lawrence’ took on a new meaning following the rulings, while the context of ‘mother’ remained unusually stable. Compared to the neighborhood change scores of the word2vec model, these findings are in line with other researchers’ observations that the Word2vec SGNS algorithm tends to be conservative in identifying semantic change relative to other approaches (Hamilton et al. 2016a).

**B.2 Further Word Substitution Analysis**

In this section, I return to the word2vec model to provide more detailed analysis of whether commissioners substituted ‘insight’ for another word that was already in circulation prior to the court rulings. Recall that **Figure 3** presented the number of sentences containing ‘insight’ or its 15 nearest neighbors from the word2vec model trained on the transcripts before the rulings. (The 15 nearest neighbors in the word2vec model represent words that appear in similar contexts.) **Figure 4** presents the Pearson correlation coefficients for ‘insight’ and each of those 15-closest neighbors at the decision level across the whole study period. To make this figure, I counted how many times each word occurs in every decision in the corpus and calculated the Pearson correlation between the occurrence counts to provide a more fine-grained analysis of co-occurrence patterns.

[FIGURE 4 ABOUT HERE]

The figure shows that ‘insight’ is either positively correlated or not correlated with its nearest neighbors at the level of individual decisions. Notably, ‘insight’ is most strongly correlated with ‘remorse’, ‘causative’, and ‘understanding’, suggesting that when commissioners use one of these terms in a decision, they are more likely to invoke ‘insight’ in the same decision. The figure also reveals further information about the neighborhood of ‘insight.’ For instance, ‘underlying’ also tends to co-occur in decisions with ‘causative’ and ‘understanding.’ Meanwhile, other words appear uncorrelated with the rest of the neighborhood, such as ‘characterological’ and ‘woefully.’ While the word2vec analysis suggests that those terms come up in similar textual contexts as ‘insight’ and the rest of the neighborhood before the rulings, this co-occurrence analysis suggests that the actual decisions that use those terms are not strongly correlated with discussion of ‘insight’ or other words in the neighborhood.

**WORKS CITED**

Chiu, Billy, Anna Korhonen, and Sampo Pyysalo. 2016. “Intrinsic Evaluation of Word Vectors Fails to Predict Extrinsic Performance.” In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, 1–6. Berlin, Germany: Association for Computational Linguistics.

Goldberg, Yoav, and Omer Levy. 2014. “Word2vec Explained: Deriving Mikolov et al.’s Negative-Sampling Word-Embedding Method.” *ArXiv:1402.3722 [Cs, Stat]*, February.

Hamilton, William L., Jure Leskovec, and Dan Jurafsky. 2016a. “Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change.” *ArXiv:1605.09096 [Cs]*, May.

———. 2016b. “Cultural Shift or Linguistic Drift? Comparing Two Computational Measures of Semantic Change.” *ArXiv:1606.02821 [Cs]*, June.

Jurafsky, Dan, and James H. Martin. 2020. *Speech and Language Processing*. 3rd Edition (draft).

Levy, Omer, and Yoav Goldberg. 2014. “Neural Word Embedding as Implicit Matrix Factorization.” In *Advances in Neural Information Processing Systems 27*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, 2177–2185. Curran Associates, Inc.

Levy, Omer, Yoav Goldberg, and Ido Dagan. 2015. “Improving Distributional Similarity with Lessons Learned from Word Embeddings.” *Transactions of the Association for Computational Linguistics* 3 (December): 211–25.

Michel, Jean-Baptiste, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, The Google Books Team, Joseph P. Pickett, et al. 2011. “Quantitative Analysis of Culture Using Millions of Digitized Books.” *Science* 331 (6014): 176–82.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. “Efficient Estimation of Word Representations in Vector Space.” *ArXiv:1301.3781 [Cs]*, January.

Pagel, Mark, Quentin Atkinson, and Andrew Meade. 2007. “Frequency of Word-Use Predicts Rates of Lexical Evolution throughout Indo-European History.” *Nature* 449: 717–20.

Rehurek, Radim, and Petr Sojka. 2010. “Software Framework for Topic Modelling with Large Corpora.” In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: Citeseer.

Spirling, Arthur, and Pedro L. Rodriguez. 2019. “Word Embeddings: What Works, What Doesn’t, and How to Tell the Difference for Applied Research.” *Working Paper*.

Tahmasebi, Nina, Lars Borin, and Adam Jatowt. 2019. “Survey of Computational Approaches to Lexical Semantic Change.” *ArXiv:1811.06278 [Cs]*, March.

**TABLES AND FIGURES**

**Chart, bar chart, histogram

Description automatically generated**