Word Embeddings
What works, what doesn't, and how to tell the difference for applied research*

Arthur Spirling† Pedro L. Rodriguez‡

Abstract

We consider the properties and performance of word embeddings techniques in the context of political science research. In particular, we explore key parameter choices—including context window length, embedding vector dimensions and the use of pre-trained vs locally fit variants—with respect to efficiency and quality of inferences possible with these models. Reassuringly, with caveats, we show that results are robust to such choices for political corpora of various sizes and in various languages. Beyond reporting extensive technical findings, we provide a novel crowdsourced "Turing test"-style method for examining the relative performance of any two models that produce substantive, text-based outputs. Encouragingly, we show that popular, easily available pre-trained embeddings perform at a level close to—or surpassing—both human coders and more complicated locally-fit models. For completeness, we provide best practice advice for cases where local fitting is required.

9,984 words

*First version: March 8, 2019. This version: September 16, 2019. Comments from Elena Labzina, Brandon Stewart and Jacob Montgomery, and audiences at MPSA 2019, Polmeth 2019 and APSA 2019, are gratefully acknowledged.

†Associate Professor of Politics and Data Science, New York University (arthur.spirling@nyu.edu)

‡Postdoctoral Fellow, Data Science Institute (joint with Political Science), Vanderbilt University and Adjunct Professor, Instituto de Estudios Superiores de Administración (pedro.rodriguezsosa@gmail.com)

1 Introduction

The idea that words and documents can be usefully expressed as numerical objects is at the core of much modern political methodology. The exact method one uses to model "text as data" has been debated. But in recent times, so called "word embeddings" have exploded in popularity both inside and outside our discipline. The premise of these techniques is beguilingly simple: a token of interest ("`welfare`" or "`washington`" or "`fear`") is represented as a dense, real-valued vector of numbers. The length of this vector corresponds to the nature and complexity of the multidimensional space in which we are seeking to "embed" the word. And the promise of these techniques is also simple: distances between such vectors are informative about the semantic similarity of the underlying concepts they connote for the corpus on which they were built. Applications abound. Prosaically, they may be helpful for a 'downstream' modeling task: if consumers search for "`umbrellas`", they may also want to purchase "`raincoats`", though not "`picnic`" equipment. Or the similarities may be substantively informative *per se*: if the distance between "`immigrants`" and "`hardworking`" is smaller for liberals than for conservatives, we learn something about their relative worldviews.

Exploiting the basic principles behind these examples, word embeddings have seen tremendous success as feature representations in well-known natural language processing problems. These include parts-of-speech tagging, named-entity-recognition, sentiment analysis and document retrieval. Given the generality of those tasks, it is unsurprising that word embeddings are rapidly making their way into the social sciences (e.g. Kozlowski, Taddy and Evans, 2018), political science being no exception (e.g. Rheault and Cochrane, 2019; Rodman, 2019). But as is often the case with the transfer of a technology, there is a danger that adoption will outpace understanding. Specifically, we mean comprehension of how well the technology performs—technically and substantively—on specific problems of interest in the domain area of concern. The goal of this paper is to provide that understanding for political science, enabling practitioners to make informed choices when using these approaches.

This broad aim stated, we now clarify our particular focus. As conveyed in our examples above, word embeddings serve two purposes. First they have an instrumental function, as feature representations for some other learning task. So, crudely, while we care that advertising "`raincoats`" to those interested in an "`umbrella`" improves the user experience, we don't much care *why* this is. That is, we don't have a deep linguistic interest in these terms, or what their nearness tell us about society or its development. Second then, embeddings are a direct object of interest for studying word usage and meaning—i.e. human semantics. Good performance in the former need not, indeed often does not, correlate with good performance in the latter (Chiu, Korhonen and Pyysalo, 2016).

In this paper we focus on this second purpose: embeddings as measures of meaning. The reasoning is simple. First, we cannot pretend to foresee all the downstream use cases to which political scientists will apply embeddings. Moreover, given a well-defined downstream task, how to think about performance is trivial—these are usually *supervised* tasks with attendance metrics measuring accuracy, precision and recall. Second, word usage, including differences between groups and changes over time, is of direct and profound interest to political scientists. There are, however, no well-defined validation metrics beyond those used in the computer science literature which need not apply well to political science and indeed have important limitations (Faruqui et al., 2016).

With this in mind, our specific contribution goes beyond (what we consider) a useful series of results. We propose the framework used to generate them, that will guide researchers through the maze of choices that accompany word embeddings. These include whether to use cheap pre-trained or (more) expensive 'local' corpus trained embeddings. And, within models, we demonstrate the effects of altering core parameters such as context *window size* and *embedding vector length*. In addition to standard predictive performance and computational cost metrics though, we present two novel approaches to model comparison and validation. First, framing the task as an information retrieval one, we show how models may be mechanically compared in terms of the words they place close to others—including politics-specific tokens. As a second "gold-standard" approach, we propose a new take on the classical "Turing test" wherein human judges must choose between computer generated nearest neighbors and human generated nearest neighbors. While we necessarily make certain choices in terms of embedding architecture and which parameters to focus on, we stress the framework herein developed is completely general and not beholden to these choices. It is easily adaptable to evaluate new models—including non-embedding models of human semantics—and other parameter variations.

Our findings are ultimately reassuring for practitioners. In particular, (cheap, readily available) pre-trained embeddings perform extremely well on a multitude of metrics relative to human coding and (expensive) locally trained models for political science problems. This is true beyond our focus *Congressional Record* corpus, and extends even to non-English collections. Separate to our intellectual contribution, we also provide the full set of all local models we fit (some of which were time-consuming and computationally expensive), so practitioners can use them "off-the-shelf" in their own work.

We will discuss the choices practitioners face momentarily. Before that, we provide a brief overview of the embeddings literature to clarify terms for what follows.

2 Word Embeddings in Context

The methods to implement word embeddings in a scalable way are new. The central theoretical concepts are not. Indeed, modern incarnations of these models find common ground in the distributional semantics literature dating back to at least the 1950s (e.g. Wittgenstein, 1953; Harris, 1954; Firth, 1957). They now go by various names: semantic vector space models, word space models or—our preferred nomenclature—distributional semantic models (DSMs).

2.1 Local Windows: The Distributional Hypothesis

The key insight of the early theoretical work was that we can "know a word by the company it keeps" (Firth, 1957, 11). More concretely, a word's meaning can be garnered from its contextual information: literally, the other words that appear near it in text. Formalizing this idea, the "distributional hypothesis" suggests that words which appear in similar contexts are likely to share similar meanings (Harris, 1970). A "context" here would typically mean a symmetric window of terms around the word of interest.

When DSMs for large corpora took off empirically in the 1990s, the distributional insight was applied in very different ways. Notable efforts include Latent Semantic Analysis (Landauer and Dumais, 1997), Hyperspace Analogue to Language (HAL) (Lund and Burgess, 1996) and Latent Dirichlet Allocation (LDA) (Blei, Ng and Jordan, 2003). Of these, LDA and its variants (e.g. Quinn et al., 2010; Roberts et al., 2014) have proved extremely popular in social science but the implementations of these techniques do not require (nor typically recommend) local windows of text within documents. Instead context is typically defined to be an entire document.

2.2 Embeddings: Neural Models

While the logic of local windows is straightforward to describe, systematic modeling of word sequences is extremely challenging. The key innovation, provided by Bengio et al. (2003), was conceiving of words as distributed representations within a neural language model. Here *neural* means based on a (artificial) neural network, a very flexible framework for learning relationships that has appeared in some political science contexts (e.g. Beck, King and Zeng, 2000). The Bengio et al. approach maps words to real-valued vectors. Intuitively, each element of those vectors represents some hypothetical characteristic of the word(s). The vector for a given word can be called a *word embedding*.

These vectors are obviously conceptually and practically different to those of the "vector space model" well-known to political scientists. For one thing, word embedding vectors are, per their name, for *words*. In vector space models it is *document* vectors that are of interest. In addition, the embedding vectors are the *result* of applying a model, whereas the document vectors are the *input* to one.

Building on Bengio et al. (2003), and key for our interests, Collobert and Weston (2008) (see also Collobert et al., 2011) demonstrated that while word embeddings are useful for downstream tasks, they also carry substantive syntactic and semantic information about language *per se*. Again this is a conceptual shift from the traditional vector space modeling in political science. There, words are discrete symbols. Their meaning is exogenous to the endeavor at hand, and we simply count (in various ways) their occurrence. In the embeddings literature, the meaning of words is itself a quantity that can be *learned*; furthermore their vector representations often allow for simple but informative mathematical operations. A textbook case is to note that (certain) embeddings can produce analogies like `king` - `man` + `woman` ≈ `queen`, where each term is represented as vector in D dimensional space. But there was a second advancement in this work: the authors alleviated a methodological problem that made earlier estimation (by e.g. Bengio et al., 2003) very slow.

2.3 The rise and rise of Word2Vec and GloVe

Mikolov et al. (2013) took the logic of the Bengio et al. (2003) model, but focused solely on producing accurate word representations. These authors reduced the complexity of the model, and allowed for its scaling to huge corpora and vocabularies. Released as a set of models called `Word2Vec`, this work is so popular that it has confusingly become almost synonymous with both embeddings and DSM. Beyond modeling improvements, `Word2Vec` included several preprocessing steps that are key to its performance (Levy, Goldberg and Dagan, 2015). Soon after its release, Pennington, Socher and Manning (2014) proposed a competing algorithm—Global Vectors, or `GloVe`—that showed improved performance over `Word2Vec` in a number of tasks. The most notable difference between the two is that `Word2Vec` follows an *online learning* approach, that is, the model is trained as the local window is moved from word to word along the corpus. `GloVe` also employs local windows, but does so to compute global co-occurrence counts prior to training the model. Despite this difference, the two approaches are not mathematically very different.

Both camps released software that allowed researchers to use pre-trained embeddings (fit to a corpus such as the English entries on Wikipedia) or estimate their own. This latter scenario implies fitting one of the models "locally" to a researcher's particular corpus. Regardless of the specific implementation, initial comparative studies suggested word embedding models resoundingly out-perform traditional DSMs in a range of tasks (Baroni, Dinu and Kruszewski, 2014), though these claims have been subsequently moderated (Levy, Goldberg and Dagan, 2015).

Our perception is that `GloVe` is more popular in social science applications, so most of our work is focused on this approach below. But we also include explicit comparisons with `Word2Vec`—where, with caveats, we find the models indeed generate similar results.

2.4 Beyond Feature Representations: Use Cases for Social Scientists

Although word embeddings have been most widely used as feature representations for downstream tasks, an increasing number of studies have used them to study word usage and meaning which the focus of our paper. Of these, several have used word embeddings to perform diachronic studies of language —the study of linguistic changes over time (Kulkarni et al., 2015; Hamilton, Leskovec and Jurafsky, 2016*b*; Garg et al., 2018; Rodman, 2019). These papers usefully evaluate a series of methods to make separately trained embeddings models—on data from different time periods—comparable, along with proposing ways to quantify uncertainty. Other studies have focused on methods to identify biases—e.g. common stereotypes—inherent in embedding models trained on large corpora (Islam, Bryson and Narayanan, 2016; Bolukbasi et al., 2016). While the priority of these studies is often to "correct" for these biases, for social scientists these biases are precisely the object of interest (Garg et al., 2018). Word embeddings have also been applied to study cultural differences (Hamilton, Leskovec and Jurafsky, 2016*a*; Kozlowski, Taddy and Evans, 2018). Rheault et al. (2016) provide an interesting innovation is the use of embedding models, simultaneously estimating word and party embeddings to study partisanship. Finally, another use case worth highlighting leverages pre-trained word embeddings to aid the construction of topic-specific dictionaries (Fast, Chen and Bernstein, 2016). This list of works is hardly exhaustive, but hopefully serves to give a sense of how these methods can and might be used.

3 Embedding Models and Parameter Choices

The application of any statistical model requires choices; embeddings are no exception. For political scientists downloading code (or indeed downloading pre-fit embeddings), at the very least, they need to decide:

1. how large a **window size** they want the model to use.
2. how large an **embedding** they wish to use to represent their words.
3. whether to fit the embedding models **locally**, or to use **pre-trained** embeddings fit to some other (ideally related) corpus.

We now discuss the nature of these choices. In addition, we explain some other important features of embeddings for researchers: namely, the fact that embeddings demonstrate *instability* in practice, and what one might do about this. In general, we note that there is often little guidance in the literature as to how decisions should be made—and virtually none at all for social science problems. A final caveat here is that, of course, there are many other parameter choices beyond the ones we specify in this section; for example, `Word2Vec` allows one to choose a learning rate for its backpropogation algorithm, and all models can use documents that have been preprocessed differently. Using our methods below, users can make decisions over them in the same way. But we keep our focus on the three above because they seem most central to empirical research.

3.1 Window-size

Window-size determines the number of words, on either side of the focus word to be included in its context.[1] The type of semantic relationship captured by embeddings has been found to vary with window-size, with larger window sizes (> 2) capturing more topical relations (e.g.

[1]Context windows can also be asymmetric, in which case window-size refers to the number of words on one side of the focus word to be included in its context. Asymmetric windows are better able to account for word order which may be useful for some tasks. Pennington, Socher and Manning (2014) for example find asymmetric windows produced embeddings better suited for syntactic tasks. Nevertheless, symmetric windows are the default option for most use cases.

Obama - President) and smaller window sizes (< 2) capturing syntactic relations (e.g. dance - dancing).[2]

For topical relationships, larger windows (usually 5 or above) tend to produce better quality embeddings although with decreasing returns—a result highlighted by Mikolov et al. (2013) and which we corroborate below. Intuitively, larger contexts provide more information to discriminate between different words.[3] Take, for example, the following two sentences: cows eat grass and lions eat meat. A window-size of 1 does not provide enough information to distinguish between cows and lions (we know they both eat, but we don't know what) whereas a window-size of 2 does.

3.2 Embedding Dimensions

This parameter determines the dimensions of the embedding vectors which usually range between $50 - 450$. We can think of these dimensions as capturing different aspects of "meaning" or semantics—hidden to the researcher—that can be used to organize words.[4] Too few dimensions—imagine the extreme of 1—and we will miss potentially meaningful relationships between words; too many—imagine the extreme of a full co-occurrence vector with every word in the vocabulary—and some dimensions are likely to be redundant (add no information). Factors such as vocabulary size and topical specificity of the corpus are likely to play a role, although theoretical work in this area remains scant.[5] Empirically, more dimensions generally improve performance across a wide variety of tasks but with diminishing returns. Interestingly, extant literature suggests that the point at which improvements become marginal differs depending on the problem. For downstream tasks optimal performance can sometimes be reached with as few as 50 dimensions (Melamud

[2]Though note that removing words from a corpus prior to processing into input-target pairs effectively enlarges the window-size (Levy, Goldberg and Dagan, 2015). This need only really be of concern when interested in syntactic relationships which requires smaller windows.

[3]See Supporting Information C for a quick empirical verification of this claim for real data.

[4]Although the dimension per se have no specific meaning, embedded words can be subsequently projected onto a meaningful—researcher defined—dimension (see Kozlowski, Taddy and Evans (2018)).

[5]Of the few that we could find, Patel and Bhattacharyya (2017) posit that the number of pairwise equidistant words of the corpus vocabulary measured using the term co-occurrence matrix provides a lower bound on the number of dimensions. It is not entirely clear what this equidistant metric means substantively.

et al., 2016). Semantic tasks, on the other hand, continue to show significant improvements until around 200 – 300 dimensions (Pennington, Socher and Manning, 2014).[6] This difference is likely a result of downstream tasks leveraging specific aspects of meaning—for example, a sentiment classification task will likely benefit from embeddings that focus on discriminating words along affect-related dimensions.

3.3 Pre-Trained Versus Going Local

Embedding models can be data hungry, meaning they need a lot of data to produce 'useful' results. Consequently, researchers with small corpora often use generic pre-trained embeddings trained on much larger corpora. Pre-trained embeddings also help avoid the overhead cost associated with estimating and tuning new embeddings for each task. However, there are trade-offs. The training corpus used to estimate these embeddings need not accurately capture the semantics of domain-specific texts. Intuitively, we would want to use pre-trained embeddings trained on a corpus generated by a similar "language model" to that which generated our corpus of interest. The more similar the two language models, the more similar the underlying semantics. For a highly specific local corpus—a corpus in Old English for example—generic pre-trained embeddings may not be all that useful.

In what follows we compare the set of embeddings from a set of locally trained models using political corpora to one of the most popular pre-trained embeddings—`GloVe`. Our results show high correlations between both models, suggesting pre-trained embeddings may be appropriate for certain political corpora. However, we stress that researchers need be transparent about the implied assumptions when deciding to use pre-trained embeddings.[7]

[6] Coincidentally, in their evaluation of LSA Landauer and Dumais (1997) found that 300 dimensions produced the best results on several validations tasks.

[7] We are aware of an alternative new approach, wherein researchers "retrofit" generic (pre-trained) embeddings using additional domain-specific semantic information (e.g. Faruqui et al., 2014; Dingwall and Potts, 2018; Khodak et al., 2018)). We consider this beyond our current scope.

3.4 Non-Convexity and Instability

Both `Word2Vec` and `GloVe` have non-convex objective functions. As such the solution space is likely to be multi-modal. In practice this means that the embedding space of two models trained on the same corpus and with the same parameter choices may differ substantially—a fact observed in previous work (Wendlandt, Kummerfeld and Mihalcea, 2018) and which we confirm empirically below. This "instability" can be particularly problematic when drawing qualitative inferences from the embeddings themselves, with equivalent models producing widely different nearest neighbor rankings —words most semantically proximate to a target word. Magnifying this instability are various sources of randomness in the estimation of word embeddings, most notably random initialization of the embedding vectors and random order of training documents. While all words are affected, some are more affected than others (Pierrejean and Tanguy, 2018). It is worth noting that `GloVe` has been found to be more stable than `Word2Vec`, probably because of its use of a global co-occurrence matrix rather than an online local window context (Mimno and Thompson, 2017; Wendlandt, Kummerfeld and Mihalcea, 2018).[8]

To account for the inherent instability in the estimation process we recommend researchers estimate a given model over *multiple initializations* of the corpus—we use ten—and use the average of the metric of interest. We accept that variation between realized embeddings is simply a fact of life; nonetheless, for what follows we presume that researchers want to know how stability correlates with model specification.

4 Evaluating Embedding Models for Social Science

To evaluate which choices are optimal we need evaluation tasks. For word embeddings, tasks fall into one of two categories: *extrinsic* and *intrinsic*.

Extrinsic tasks include various downstream NLP problems such as parts-of-speech tagging,

[8]Separate to instability, it is reasonable to expect embeddings to differ as a result of *sampling variability*. If we view any given corpus as a particular instantiation of a superpopulation of linguistic entities, then we should adjust for this with the equivalent of a standard error. See Antoniak and Mimno (2018) for bootstrapping ideas pertaining to this problem.

named-entity-recognition, classification and document retrieval. These are usually supervised, and have well-defined performance metrics. For this paper we considered evaluating embeddings this way. However, it was not immediately obvious to us which tasks, if any, represented good baselines for political scientists.[9] As noted by Denny and Spirling (2018), there has been very low take up of supervised learning problems in political science relative to unsupervised learning problems. Moreover evidence of good performance need not generalize. How much should a researcher interested in information retrieval update when informed that a given embedding model performs well in a classification task of congressional speeches? Given a well-defined downstream task, we recommend users first consider pre-trained embeddings if reasonably appropriate—unlikely if the corpus of interest is in Old English—before proceeding to tune a locally trained model.

Intrinsic tasks evaluate embeddings as models of semantics, which is our focus. These include word analogy—algebraic operations are performed using word vectors to answer questions such as "France is to Paris as Germany is to…"; word similarity—pairs of words along with their human provided similarity ratings are compared to similarity ratings computed using word embeddings; synonym tests—TOEFL multiple-choice synonym questions; noun-clustering—a similarity measure is used to assign words to a pre-defined number of semantic classes; sentence completion (specific to the Skip-Gram architecture)—select from multiple choices to fill in the missing word in a sentence. These tasks require human generated data. Researchers tend to rely on existing datasets that are either freely available online or can be requested from the original authors. However, this can be problematic as existing datasets may be ill-suited to a particular corpus or for a particular semantic relation of interest. For example, word similarity datasets often do not differentiate between the various ways in which two words can be related (Faruqui et al., 2016).[10] Moreover, semantic relationships are likely to vary as a function of demographics (Halpern and Rodriguez, 2018; Garimella, Banea and Mihalcea, 2017), yet few datasets have information on the background characteristics of the subjects that generated them. The role of demographics or other

[9] The lack of consensus extends beyond political science (Nayak, Angeli and Manning, 2016).

[10] For example Agirre et al. (2009) distinguish between similarity—as in coffee and tea—and relatedness—as in cup and coffee.

background characteristics, including partisanship, is of particular relevance to social scientists. Indeed, these differences are often precisely what we are interested in! Below we make the case for crowdsourcing as a flexible alternative allowing researchers to tailor the tasks to specific objectives and gather demographic information when appropriate (Benoit et al., 2016; Schnabel et al., 2015).

We compare models using four criteria:

1. technical criteria —model loss and computation time;

2. query search ranking correlation—Pearson and rank correlations of nearest neighbor rankings;

3. model variance (stability)—within-model (holding parameters constant) Pearson correlation of nearest neighbor rankings across multiple initializations;

4. human preference— a "Turing test" assessment and rank deviations from human generated lists

Criteria 2 and 4 can also be used to compare pre-trained embeddings with locally-trained embeddings, which we do. To illustrate this framework, we compare pre-trained embeddings to a set of locally trained embedding models varying in two parameters: embedding dimensions and window-size. Before proceeding with our estimation framework we discuss each criteria in greater depth.

4.1 Technical Criteria

The most straightforward metric to compare different models is prediction loss at the point of convergence (i.e. when training stops). `GloVe`'s objective minimizes the weighted difference between the dot product of a given word pair's embeddings and the log of their global co-occurrence count.[11]

We consider window-size a tuning parameter. As noted above, the choice of window-size may be informed by the type of semantic relation of interest—syntactic or topical. If a specific window-size is chosen on theoretical grounds—whatever they may be—then it would no longer be a tuning parameter and it would be unreasonable to compare models of different window-sizes. Instead we opt to choose window-size as a function of model performance. If the intuition motivating `GloVe` is correct, namely that meaning is strongly connected to co-occurrence ratios, then the window-size that optimizes the correspondence between the embedding vectors and the global co-occurrence statistics should produce the more "meaningful" embeddings. Generally speaking, larger window sizes and more dimensions both translate into longer computation times, resulting in a performance vs computation time tradeoff. We therefore also compare the set of locally-trained models in terms of computation time in minutes.

4.2 Query Search Ranking Correlation

While prediction loss is informative, it is not obvious how to qualitatively interpret a marginal decrease in loss. Ultimately, we are interested in how a given embedding model organizes the semantic space relative to another. To evaluate this, we appeal to the information retrieval literature. A common objective in information retrieval problems is to rank a set of documents in terms of their relevance to a given query. In our case we are interested in how two models rank words in a common vocabulary in terms of their semantic similarity—defined by some distance metric such as cosine similarity— with a given query term. To do so we use both Pearson and rank correlations. The higher these correlations, the more similar the embedding spaces of both models. Below we

[11] Specifically:

$$J = \sum_{i,j=1}^{V} f(X_{ij}) \left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log(X_{ij}) \right)^2$$

where X_{ij} is the co-occurrence count of words w_i and w_j, f is a weighting function and b_i and b_j are word-specific bias parameters (Pennington, Socher and Manning, 2014).

discuss how we went about choosing the query terms. [12]

4.3 Stability

As we discussed above, embedding models are unstable due to the non-convexity of their objective function. The magnitude of this instability is likely to vary for different parameter choices. To quantify this we estimate the same model multiple times and compute the average pairwise Pearson correlation of nearest neighbor rankings for a random set of query terms (described below). Given ten separately estimated models for a given parameter pair, we have 45 pairwise correlations for each model ($\frac{n(n-1)}{2}$, or the lower diagonal of the 10×10 correlation matrix). We compare the average of these pairwise correlations across models.

4.4 Human Preferences

The output of distributional models with strong predictive performance need not be semantically coherent from a human standpoint. This point was illustrated by Chang et al. (2009) in the case of topic models. For this reason we make a clear distinction between predictive performance and semantic coherence, and propose separate metrics to evaluate both.

4.4.1 Turing Assessment

To evaluate semantic coherence we draw inspiration from the fundamental principles laid out by Turing (1950) in his classic article on computer intelligence. In that context, a machine showed human-like abilities if a person engaging in conversation with both a computer and a human could not tell which was which. We use that basic intuition in our study. In particular, an embedding model achieves "human" performance if human judges—crowd workers—cannot distinguish between the output produced by such a model from that produced by independent human coders. In our case, the idea is not to "fool" the humans, but rather to have them assert a preference for one

[12] We also include results using the intersect over the union—the Jaccard Index— for several values of N (see Supporting Information B).

set of outputs over another. If a set of human judges are on average indifferent between the human responses to a prompt and the model's responses, we say we have achieved human performance with the model. By extension, a model can achieve *better than human* performance by being on average preferred by coders. Naturally, models may be *worse than human* if the judges like the human output better.

Before getting into specifics, it is helpful to clarify some aspects of the intuition. First, there is a superficial similarity between our approach and more conventional supervised learning problems. This is misleading. In those arrangements, the researcher employs humans to hand-code a training set. Then they use a model to learn the relationships between the features of the data and the class labels given by the humans. After this, the analyst sees how well the machine can predict "held out" human labels in a test set. The machine's performance can then be directly assessed in terms its ability to replicate the human judgments for each case. But this is not what we are doing. Instead, we ask whether humans themselves, on seeing a statistical model's best attempt to describe a concept, find that representation reasonable relative to one produced by other humans. Second, while the Turing test connotes a human versus machine contest, the approach here is more general. Indeed, any output can be compared to any other—including where both sets are produced by a model or both by humans—and conclusions drawn about their relative performance as judged by humans.

The steps we take to assess the relative Turing performance of the models are as follows:

1. **Human generated nearest neighbors:** For each of ten political prompt words (described below) have humans—crowd workers on Amazon MTurk—produce a set of nearest ten neighbors—we have 100 humans perform this task. Subsequently rank "human" nearest neighbors for each prompt in terms of the number of mentions and choose the top 10 for each prompt.

2. **Machine generated nearest neighbors:** For the embedding model under consideration—pre-trained or some variant of the locally fit set up—produce a list of ten nearest neighbors

for each of the ten given prompt words above.[13]

3. **Human rating:** Have a separate group of humans perform a Triad task —135 subjects on average for each model comparison— wherein they are given a prompt word along with two nearest neighbors —a computer and a human generated nearest neighbor—and are asked to choose which nearest neighbor they consider better fits the definition of a context word.[14]

4. **Compute metric:** For each prompt compute the expected probability of the machine generated nearest neighbor —our *candidate model*— being selected vis-a-vis a *baseline model* —humans in our gold-standard. We divide this number by 0.5, as such the index will range between 0 and 2. A value of 1 implies the machine is on par with human performance (i.e. a human rater is equally likely to choose a nearest neighbor generated by the embedding model as one generated by another human) while a value larger (smaller) than 1 implies the machine performs better (worse) than humans.

In most cases there is some overlap in the set of nearest neighbors being compared. The comparisons we show subjects never include the same nearest neighbor for both models; in these cases we assume either model has 50% chance of being selected. This requires we adjust the expected probability of a machine generated nearest neighbor being selected by the probability of the triad task showing the same nearest neighbor for both machine and human. For both tasks above—collecting human generated nearest neighbors and the triad task—we created specialized `RShiny` apps that we deployed on MTurk. We restrict the set of workers to U.S. based workers with at least 100 previously approved HITs and a "Masters" qualification on Amazon Mechanical Turk. For the triad task we paid workers $1 to perform 13 such comparisons—one for each of our political prompt words, one trial run and two quality checks; for the word generation task we paid workers $3 to generate 10 associations for each of ten political prompts. Workers were not allowed to perform both tasks. The code for both apps is available from our `GitHub`.

[13]It is common in the literature to focus on the *top ten* nearest neighbors. See for example McCarthy and Navigli (2007) and Garimella, Banea and Mihalcea (2017).

[14]See Appendix for the exact wording of the task.

The quality of our approach is predicated on our human coders being able to make reasonable judgments about contexts in the way we described. Work by others that relies on similar services reassures us that this is plausible (Benoit et al., 2016; Benoit, Munger and Spirling, 2019). Unlike evaluations based on predefined databases of word associations, the Turing assessment allows us to target specific "crowds" (e.g. experts on a subject, partisans etc.) or, if using a random selection of crowd-workers, collect their demographic information. Naturally, the tradeoff is the added cost of crowdsourcing.

4.4.2 Log Rank Deviations

Using the set of human generated lists we can compare the aggregate human ranking of each nearest neighbor—as determined by token counts—with their equivalent rank on a given embedding space. So for example, if for the query `democracy` the word `freedom` is ranked 3rd according to human counts and 7th according to a given embedding space, we say it's log rank deviation is $\log((7-3)^2)$. We compute this deviation for every token mentioned by our subjects for each of our politics queries and compute an average over the set of queries for every model.[15]

5 Estimation Setup

Obviously, we need a data set on which to operate, and a particular way to model the embeddings. For the latter, as noted above, we choose `GloVe` simply because it seems more popular with social scientists,[16] though we have no reason to believe our results below would differ much under `Word2Vec`. For the data we focus on a medium sized corpus of around 1.4 million documents from American politics—though as we will see our findings are portable to other political contexts and indeed other languages.

[15] It may be worth limiting this to tokens mentioned by at least N subjects, but here we avoid making additional parameter choices.

[16] In particular, the `GloVe` pre-trained with window size 6 and embedding dimensions 300, available on February 2, 2019 from `https://nlp.stanford.edu/projects/glove/`, for which the training corpus is Wikipedia 2014 and Gigaword 5.

Below we will extend our analysis to other corpora and other languages, but for now we focus in detail on a collection we deem somewhat representative of political science efforts in this area. In particular, the set of *Congressional Record* transcripts for the 102^{nd}–111^{th} Congresses (Gentzkow, Shapiro and Taddy, 2018). These contain all text spoken on the floor of both chambers of Congress. We further restrict our corpus to the set of speeches for which party information is available.[17] We do minimal preprocessing: remove all non-text characters and lower case. Next we subset the vocabulary. We follow standard practice which is to include all words with a minimum count above a given threshold—between 5-10 (we choose 10). This yields a vocabulary of $91,856$ words.[18]

5.1 Implementing Choices

We focus our analysis on two hyperparameter choices and all 25 combinations, though to reiterate the framework we lay out is not specific to these parameter pairs:

1. window-size—1, 6, 12, 24 and 48 and
2. embedding dimension —50, 100, 200, 300, 450

To account for estimation-related instability we estimate 10 sets of embeddings for each hyperparameter pair, each with a different randomly drawn set of initial word vectors. In total we estimate 250 different sets of embeddings. The only other hyperparameter choices we make and leave fixed are the *number of iterations* and *convergence threshold*. For each model we set the maximum number of iterations to 100 and a use a convergence threshold of 0.001 such that training stops if either the maximum number of iterations is reached or the change in model loss between the current and preceding iterations is below the convergence threshold. None of our models reached the maximum number of iterations before meeting the convergence threshold. We set all remaining hyperparameter values at their default or suggested values in the `Glove` software.[19]

[17] Focusing on this subset reduces our corpus by around a third.
[18] The pre-trained `GloVe` vocabulary consists of $400,000$ tokens.
[19] We use the `text2vec` R package to run all our models.

5.2 Query Selection

Above we explained that a natural auxiliary quantity of interest is the set of nearest neighbors of a given word in the embeddings space. These form the core of our comparison metric in the sense that we will want to know how similar one set of nearest neighbors from one model specification is to another. And, by extension, how "good" one set of nearest neighbors is relative to another in terms of a quality evaluation by human judges. We use two sets of queries: a random sample of 100 words from the common vocabulary and a set of 10 curated political terms.[20]

For the politics-specific queries, we hand-picked 10 terms prior to performing any evaluations. First, there are series of concept words that we suspected would be both easily understood, but also exhibit different meanings depending on who is asked: `democracy`, `freedom`, `equality`, `justice`. Second, there are words pertaining to policy issues that are debated by political parties and motivate voting: `immigration`, `abortion`, `welfare`, `taxes`. Finally, we used the names of the major parties, which we anticipated would produce very different responses depending on partisan identification: `republican`, `democrat`. Obviously, these words are somewhat arbitrary; we could have made other choices. And indeed, we would encourage other researchers to do exactly that. Our prompts are intended to be indicative of what we expect broader findings to look like, and to demonstrate the utility of our generic approach.

6 Results: Performance Compared

This section reports the results for the evaluation metrics outlined in section 4. We begin with the technical criteria.

[20] A more systematic approach would compare the entire vocabulary (see for example Pierrejean and Tanguy (2017)). We found this prohibitively expensive, and we use a random sample of 100 words to approximate the comparisons of interest.

6.1 Technical Criteria

Figure 1a displays the mean—over all ten initializations—minimum loss achieved for sixteen (of the twenty-five) parameter pairs we considered.[21] Consistent with previous work, more dimensions and larger window-sizes both unconditionally improve model fit albeit with decreasing returns in both parameter choices. Except for very small window-sizes (< 6), improvements become marginal after around 300 dimensions. If we take loss seriously, then researchers ought avoid combining few dimensions (< 100) with small window-sizes (< 6).

But there are two important caveats here. First, it is ambiguous whether comparing *different* models on the same fitting criteria is an ideal way to make the determination about "bestness." As noted above, models with different window sizes represent qualitatively different notions of context, and presumably the match between that and the substantive problem at hand is more important than comparing relative fit. We return to this point below in giving advice and in our discussion.

Our second caveat is more prosaic: using more dimensions and/or a larger window-size comes at a cost—longer computation time (see Figure 1b). The largest of our models ($48-450$) took over three hours to compute parallelizing over eight cores.[22] This seems reasonable if only computing once and having access to several cores, but can become prohibitive when computing over several initializations as we suggest.[23] In this light, the popular parameter setting $6-300$ (window size 6, vector length 300) provides a reasonable balance between performance and computation time.

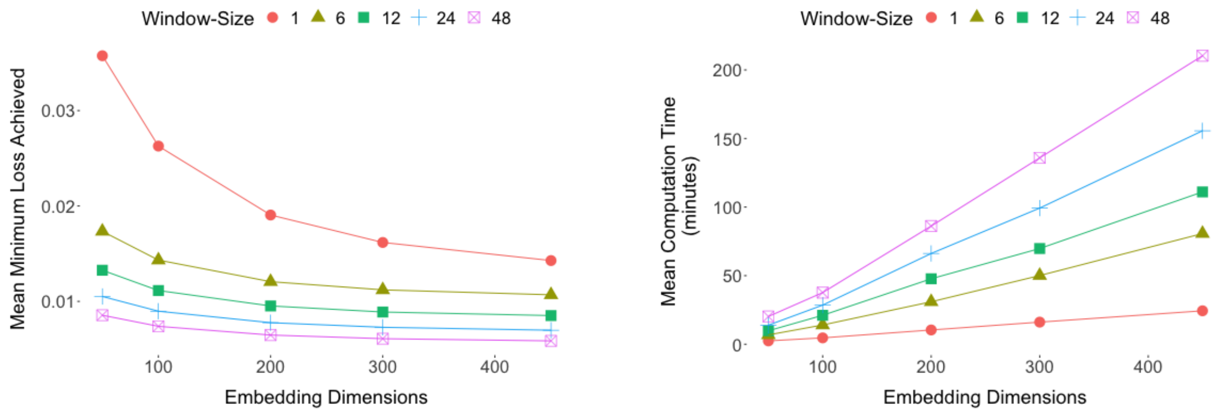
6.2 Query Search Ranking Correlation

Clearly different parameter choices produce different results in terms of performance, but what do these differences mean substantively? To answer this question we turn to comparing models with

[21] We plotted sixteen of the twenty-five parameter pairs to avoid clutter. The left-out parameter pairs follow the same trend.

[22] At the time of writing, a standard laptop has 4 cores available. Keep in mind computation time will be a function of the stopping conditions specified—number of iterations and convergence threshold. 100 iterations and a convergence threshold of 0.001 may be considered too conservative.

[23] Unless the researcher has access to a high-performance cluster (as we did) and is able to parallelize.



(a) Mean Minimum Loss Achieved (b) Computation Time (minutes)

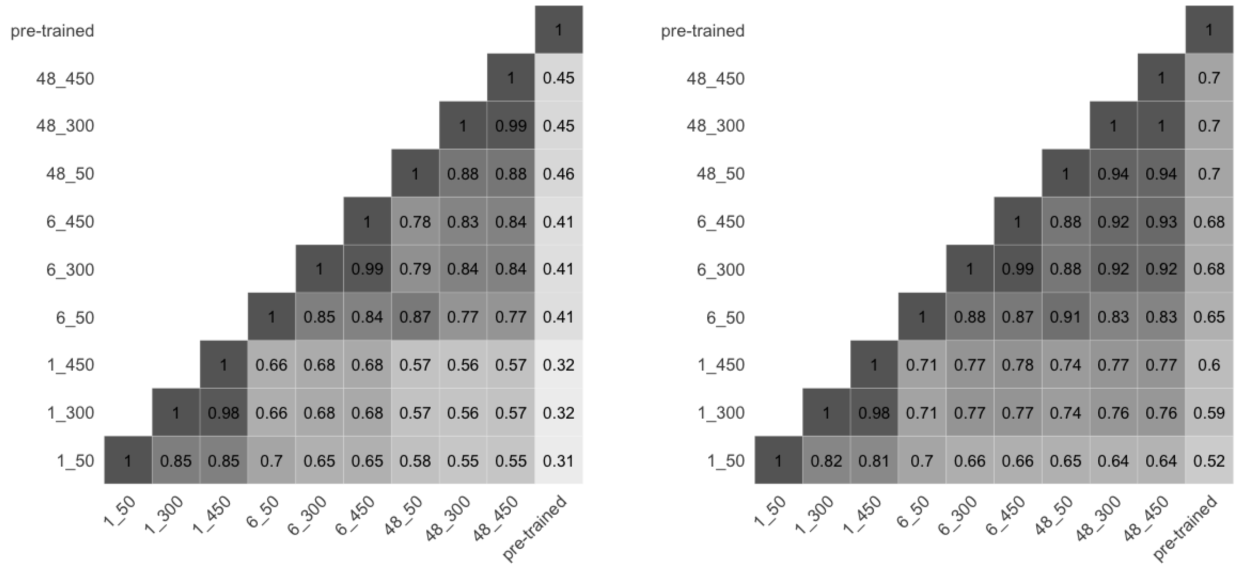
Figure 1: Technical Criteria

respect to how they rank query searches. Figure 2a displays a heatmap of pairwise correlations for all models, including `GloVe` pre-trained embeddings, for the set of random queries.[24] We observe high positive correlations (> 0.5) between all local models. Correlations are generally higher between models of the same window-size, an intuitive result, as they share the underlying co-occurrence statistics. Somewhat less intuitive, comparing models with different window-sizes, correlations are higher the larger the window-size of the models being compared (e.g. 6 and 48 vis-a-vis 1 and 6). Correlations are larger across the board for the set of political queries (see Figure 2b). These results suggest the organization of the embedding space is most sensitive to window-size but this decreases quickly as we go beyond very small window-sizes (i.e. models with window-size of 6 and 48 show much higher correlation than models with window-size of 1 and 6).

The last column of Figures 2a and 2b compare `GloVe` pre-trained embeddings with the set of local models. For this comparison we subsetted the respective vocabularies to only include terms common to both the local models and the pre-trained embeddings.[25] As would be expected, correlations are lower than those between local models, yet they are still surprisingly large—especially for local models with larger window-sizes and for the set of political queries (all above 0.5). Our

[24] As pre-trained embeddings we use the 6-300 `GloVe` embeddings.

[25] In the appendix we include additional comparisons without subsetting the vocabularies.



(a) Random Queries

(b) Politics Queries

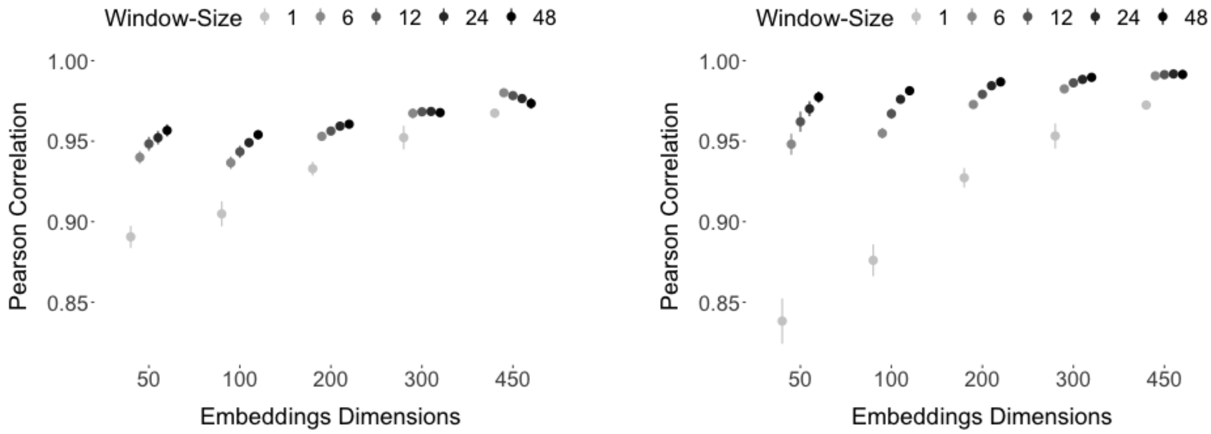
Figure 2: Query Search Ranking Criteria

reading is that `GloVe` pre-trained embeddings, even without any modifications (Khodak et al., 2018), may be a suitable alternative to estimating locally trained embeddings on present-day political corpora. This is good news for political scientists who have already relied on pre-trained embeddings in their work.

As a final check, we looked at whether pre-trained embeddings might do a 'worse' job of reflecting highly specific local embeddings for our focus corpus. In this case, we mean party: it could in principle be the case that while pre-trained embeddings do well in aggregate for the *Congressional Record* they do poorly for Democrats or Republicans specifically. To evaluate this we estimate a set of additional local models (again, 10 for each group and using 6-300 as parameter settings) for subsets—by party—of the aggregate corpus. We find no statistically significant differences in correlations (see Supporting Information D).

6.3 Stability

We next compare all parameter pairs with respect to the stability of the resulting embeddings. Figures 3a plots the distribution of Pearson correlations for the 100 random queries. Correla-



(a) Random Queries

(b) Politics Queries

Figure 3: Stability Criteria

tions are high —above 0.85— across the board, suggesting GloVe is overall rather stable —i.e. the organization of the embeddings space does not vary dramatically over different initializations. Nevertheless, models with larger window-sizes produce, on average, more stable estimates. As the number of dimensions increase, the difference in stability between different window sizes decreases and eventually flips—larger window sizes result in greater instability. This parabolic relationship between window-size, number of dimensions and stability is likely a function of corpus size —larger more generic corpora will require a greater number of dimensions to allow for multiple word senses— and token frequency —infrequent tokens are likely to be more unstable.[26] For the set of 10 politics queries we observe the same trends although do not reach the point at which the relationship reverses (see Figure 3b).

6.4 Human Preferences

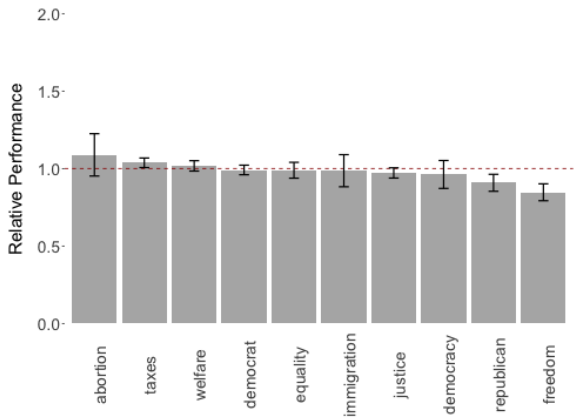
Recall that human raters represent our gold-standard evaluation metric; we assess performance here on two different types of tasks.

[26]For the State of the Union corpus, a much smaller corpus, we find the flip occurs after 100 dimensions (see Supporting Information E).

6.5 Turing Assessment

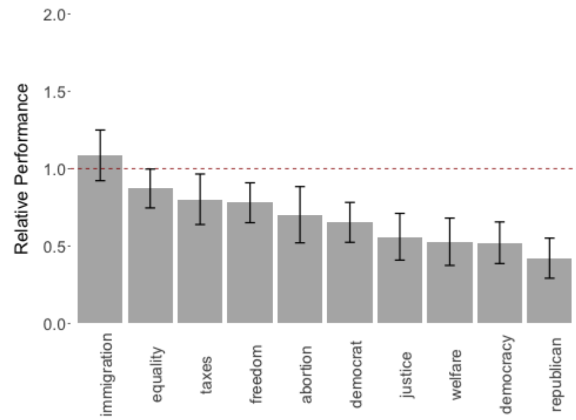
Figures 4a– 4d measure performance of a "candidate" model relative to a "baseline" model. Recall, values above (below) 1 mean nearest neighbors from the "candidate" model were more (less) likely to be chosen by human raters. A value of 1 means human raters were on average indifferent between the two models. Figure 4a compares two local models: $48-300$ (candidate) and $6-300$ (baseline). There is no unqualified winner. We see this as consistent with previous metrics—these models have a 0.92 correlation (see Figure 2b).

How do local models fare against human generated nearest neighbors? Except for one query (`immigration`), the local model of choice—6-300—shows *below-human* performance for all but two of the queries. On average, for the set of ten political queries, the local model achieves 69% (std devn= 0.20) of human performance. Turning to pre-trained `GloVe` embeddings, we observe that they are generally preferred to locally trained embeddings (see Figure 4c). Moreover, pre-trained embeddings are more competitive against humans—albeit with greater variance—achieving an average of 86% (std dev = 0.23) of human performance.



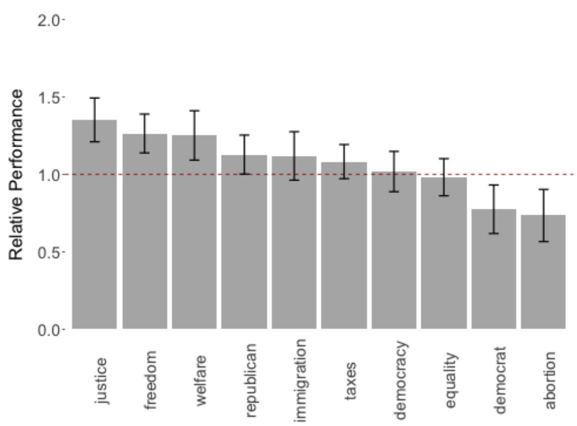
(a) Candidate: Local 48-300

Baseline: Local 6-300



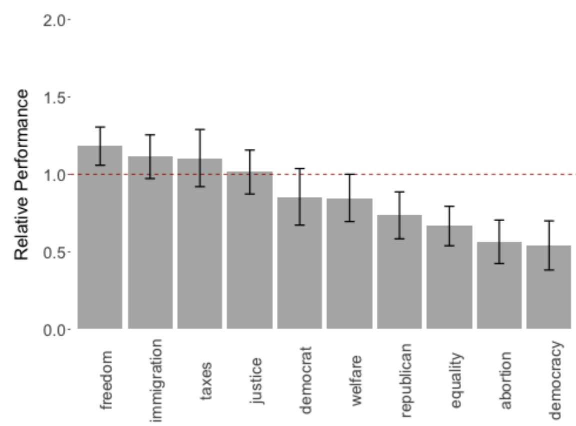
(b) Candidate: Local 6-300

Baseline: Human



(c) Candidate: `GloVe`

Baseline: Local 6-300



(d) Candidate: `GloVe`

Baseline: Human

Figure 4: Human Preferences-Turing Assessment

6.6 Log Rank Deviations

Using the log rank deviation measure, we can compare all models given our set of human generated lists (see Figure 5). Results generally mirror those obtained using our technical loss criterium, barring the large confidence intervals. Models with larger windows and more dimensions show lower log rank deviations, indicating better performance but with decreasing returns. This suggests a strong correspondence between predictive performance and semantic coherence as hypothesized by the distributional hypothesis.

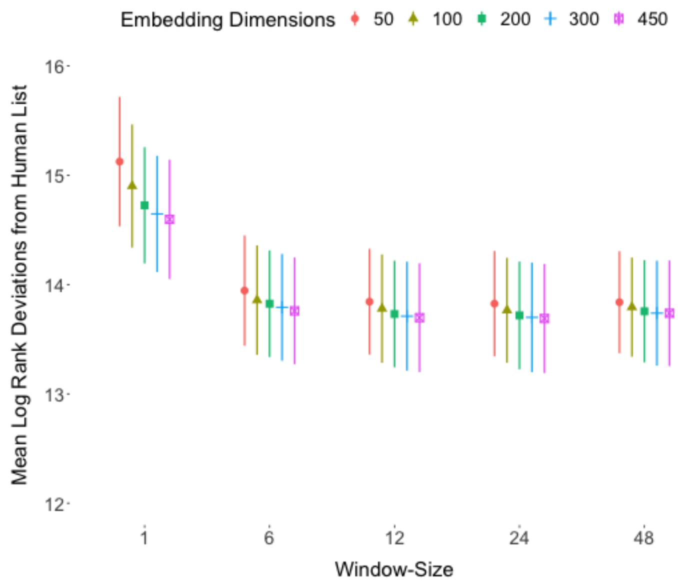


Figure 5: Human Preferences-Log Rank Deviations

7 Other Corpora, Other Languages

Our core results presented, we now extend our evaluation to four other corpora, varying in size and language. These are:

1. the full set of speeches from the UK Parliament for the period 1935 – 2016 obtained from Rheault et al. (2016)

2. all State of the Union (SOTU) speeches between 1790 and 2018

3. the full set of speeches from both chambers of the Spanish Legislature —*Cortes Generales*— for the V - XII legislatures.[27] As political queries we use: `democracia`, `libertad`, `igualdad`, `equidad`, `justicia`, `inmigracion`, `aborto`, `impuestos`, `monarquia`, `parlamento`.

[27] As the XII was ongoing at the time of writing we used all speeches available up until Oct-18, 2018.

4. the full set of speeches from the German Legislature—*Deutscher Bundestag*— for the election periods 14 - 19.[28] The political queries in this case are: `demokratie`, `freiheit`, `gleichberechtigung`, `gerechtigkeit`, `einwanderung`, `abtreibung`, `steuern`, `cdu` and `spd`.

We did not find readily available `GloVe` pre-trained embeddings in German, as such all our comparisons in this case are between locally trained embeddings. Both the Spanish and German corpora are original datasets collected for the purposes of this paper.[29]

Table 1 provides summary statistics for these corpora and the *Congressional Record* corpus. We can see that the SOTU corpus is substantially smaller than all the other corpora and also encompasses a much longer time period.

Corpus	Period	Num. of Docs.	Num. of Tokens	Avg. Tokens/Doc.	Vocab. Size
Congressional Record	1991 - 2011	1,411,740	3.4×10^8	238	91,856
Parliamentary Speeches	1935 - 2013	4,455,924	7.2×10^8	162	79,197
State of the Union	1790 - 2018	239	2.0×10^6	8143	11,126
Spanish Legislature	1993 - 2018	1,320,525	3.0×10^8	224	94,970
German Legislature	1998 - 2018	1,193,248	0.8×10^8	69	108,781

Table 1: Corpora Summary Statistics

In Supporting Information E, we provide the same results plots as we gave for our *Congressional Record* corpus. Perhaps surprisingly, but no doubt reassuringly, these are almost identical to the ones above. That is, when we look at the embedding models we fit to these very different corpora, the lessons we learn in terms of hyperparameter choices, stability and correlations across search queries (i.e. on the issue of whether to fit local embeddings, or to use prefit ones) are the same as before. Of course, there are some exceptions: for example, we do find models of window-size equal to one perform well in the case of the SOTU corpus and for the German corpus—though to a lesser extent.

[28] As the 19th *Wahlperiode* was ongoing at the time of writing we used all speeches available up until Oct-18, 2018.
[29] We have made these publicly available, and these may be downloaded via the project's github page.

8 GloVe vs Word2Vec: some differences

In contrast to GloVe, which approximates global co-occurrence counts, Word2Vec follows an *online learning* approach—the model is progressively trained as we move the context window along the corpus. Word2Vec at no point sees the global co-occurrence counts. Despite this difference, Pennington, Socher and Manning (2014), the authors of GloVe, show that GloVe and Word2Vec's skip-gram architecture are mathematically quite similar. We might then conclude that both algorithms will produce similar embeddings when trained on the same corpus. We find this is not the case.

To compare both algorithms, we implemented the same estimation setup with Word2Vec—skip-gram architecture—as we did with GloVe. That is, for each parameter pair we estimated ten different sets of embeddings, each starting from a different random initialization. We again restricted the vocabulary to words with a minimum count of 10 and ran each model for 5 epochs. Otherwise we set all parameters to their default values in the Python gensim module.[30]

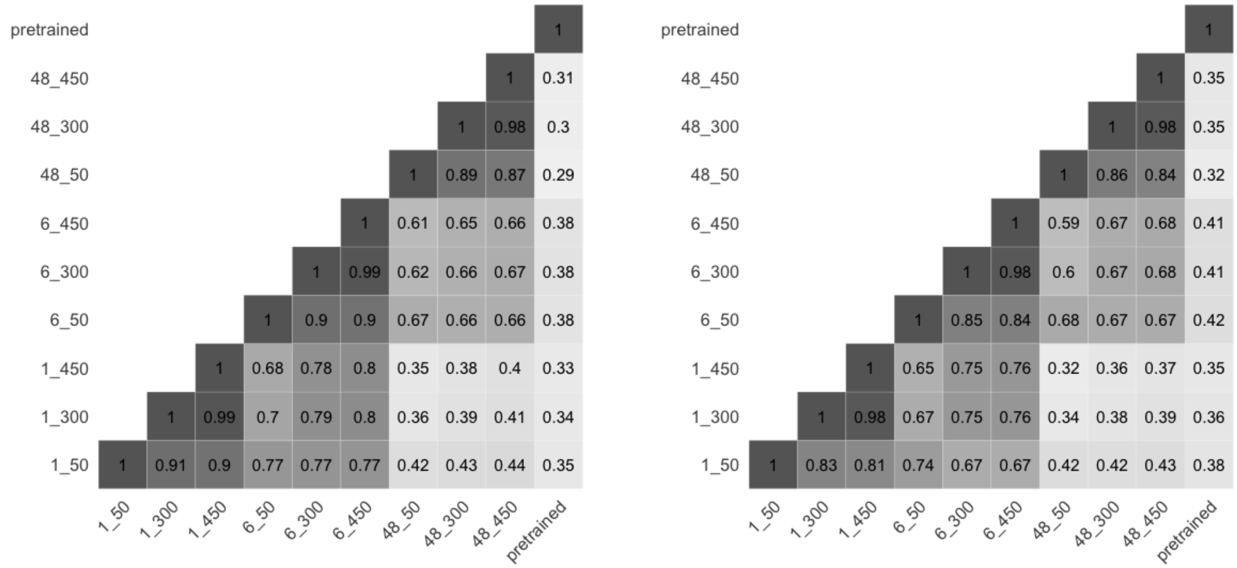
Figures 6a and 6b display the correlations between the set of locally trained models as well as between those and the pre-trained Word2Vec embeddings.[31] The results differ quite markedly from those obtained using GloVe. On the one hand, models with window size 6 are now more highly correlated with the smaller—window size 1—than with the larger models—window size 48. More importantly, Word2Vec pre-trained embeddings exhibit much lower correlations with the set of local models than was the case with GloVe.

In Figure 7 we directly compare both algorithms using a subset of the local models along with both sets of pre-trained embeddings. The correlation between both algorithms increases as we increase window size, yet it is never particularly high (for our set of parameter values). Moreover, and surprising to us, Word2Vec and GloVe pre-trained embeddings are themselves not all that highly correlated at 0.29.[32] One potential explanation for this result is that they are trained

[30]We run gensim in R using the reticulate package.

[31]We use pre-trained embeddings with a window size of 6 and embeddings dimension 300.

[32]For this comparison we subsetted both vocabularies to the intersection of the two. The Word2Vec vocabulary consists of 3 million words whereas the GloVe vocabulary consists of 400,000 words.



(a) Random Queries (b) Politics Queries

Figure 6: Query Search Ranking Criteria

on different corpora.[33] We postulate however that the main source of differences lies in the implementation details. In particular, whereas `GloVe` explicitly underweights relatively rare terms, `Word2Vec` explicitly underweights high frequency terms. Consequently, `Word2Vec` often picks out relatively rare terms (including misspellings) as nearest neighbors as evidenced in Table 2.[34] In practice this means `Word2Vec` is likely to be less "robust," i.e. embeddings will tend to be more corpus specific, than `GloVe`.

democracy		**freedom**		**equality**		**justice**		**immigration**	
W2V	**GloVe**	**W2V**	**GloVe**	**W2V**	**GloVe**	**W2V**	**GloVe**	**W2V**	**GloVe**
pluralism	freedom	liberty	liberty	equal	equal	justices	rehnquist	naturalization	naturalization
freedom	democracies	freedoms	democracy	enfranchisement	racial	rehnqulst	scalia	ins	illegal
democracyand	democratic	democracy	freedoms	racial	fairness	nowchief	owen	immigrations	ins
democracies	liberty	freedomthe	expression	liberty	gender	rehnquist	ginsburg	aliens	reform
liberty	promoting	freedomfreedom	equality	faimess	freedom	justiceand	court	immigrants	customs
democracythe	capitalism	freedoma	free	egalitarianism	liberty	scalia	souter	asylum	border
democratization	stability	pluralism	speech	suffrage	struggle	brennan	oconnor	undocumented	nationality
pluralistic	promote	freedomand	religious	nonviolence	justice	bablitch	brennan	border	immigrants
selfgovernment	pluralism	freedomour	enduring	fairness	tolerance	antonin	department	illegal	laws
democracys	peace	tyranny	prosperity	inclusiveness	harmony	justicethat	supreme	immigrant	aliens

Table 2: Nearest Neighbors `Word2Vec` (local 6-300) and `GloVe` (local 6-300): note that `Word2Vec` selects rarer terms, including typos and misspellings.

[33] `Word2Vec` is trained on a Google News corpus while `GloVe` is trained on Wikipedia 2014 and Gigaword 5.

[34] This is not wrong per se —it makes sense for a word's misspellings to be it's nearest neighbors— but is something researchers ought to keep in mind when prioritizing nearest neighbors.

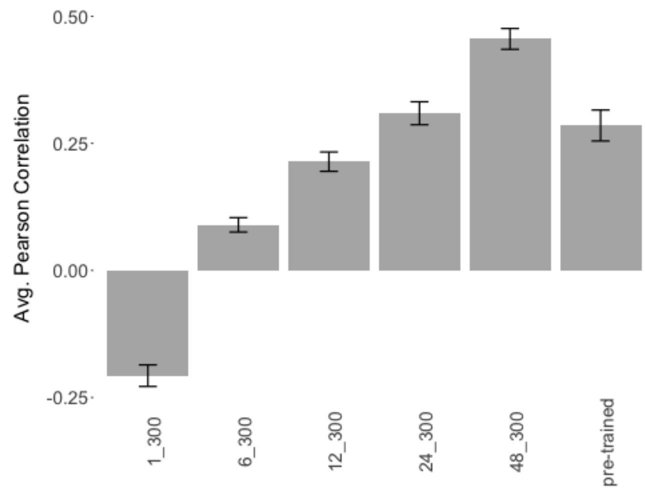


Figure 7: Avg. Pearson Correlation `GloVe` v `Word2Vec` (politics queries)

Additionally, we applied our Turing assessment to compare the two sets of pre-trained embeddings. For this exercise, we subsetted—post-estimation— the vocabularies to the intersection of the two. The latter greatly improved the quality of `Word2Vec`'s nearest neighbors by eliminating relatively rare terms (often typos). Figure 8 displays the results. Clearly, at least for our set of politics queries, our human raters are on average indifferent between the two models.

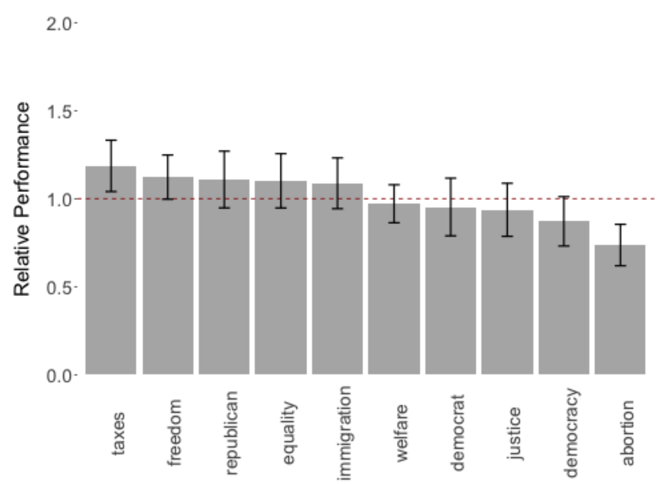


Figure 8: Human Preferences-Turing Assessment
Candidate: `Word2Vec` Baseline: `GloVe`

9 Advice to Practitioners

In this section we summarize our results in terms of what we deem to be the main takeaways for practitioners looking to use word embeddings in their research. First, in terms of *choice parameters* in applied work:

- **Window-size and embedding dimensions:** with the possible exception of small corpora like the State of the Union speeches, one should avoid using very few dimensions (below 100) and small window-sizes (< 5), especially if interested in capturing topical semantics. If one cares about syntactic relationships, then the model choice should be based on that criterion first (i.e. small windows may be desirable). While performance improves with larger window-sizes and more dimensions, both exhibit decreasing returns—improvements are marginal beyond 300 dimensions and window-size of 6. Given the tradeoff between more dimension/larger window-size and computation time, the popular choice of 6 (window-size) and 300 (dimensions) seems reasonable. This particular specification is also fairly stable meaning one need not estimate multiple runs to account for possible instability.

- **Pre-trained vs local embeddings:** `GloVe` pre-trained embeddings generally exhibit high correlations (> 0.4 for the set of random queries and > 0.65 for the set of curated queries) with embeddings trained on our selection of political corpora.[35] At least for our focus *Congressional Record* corpus, there is little evidence that using pre-trained embeddings is problematic for subdivisions of the corpus by party—Republican vs Democrat speech.

  Human coders generally prefer pre-trained representations, but not for every term, and it is quite close for many prompts. Specifically, `GloVe` pre-trained word embeddings achieve on average—for the set of political queries—80% of human performance and are generally preferred to locally trained embeddings.

  These results suggest embeddings estimated on large online corpora (e.g. Wikipedia and

[35] This is lower in the case of small corpora like the State of the Union, and in the case of random queries for the Spanish corpus.

Google data dumps) can reasonably be used for the analysis of contemporaneous political texts.

Further, if one does wish to train locally, the computational overheads are (not especially) severe, at least for a medium size corpus, so this is probably not a reason *per se* to use pre-trained embeddings.

Second, in terms of methodology lessons on *how* to evaluate models:

- **Query search:** in the absence of a clearly defined evaluation metric—a downstream task with labeled data—embeddings can be compared in terms of how they "organize" the embedding space. We propose doing so using query search ranking correlations for a set of randomly selected queries and—given a specific domain of interest— a set of representative domain-specific queries. To discriminate between models resulting in very different embedding spaces, both can be compared to a baseline, either a model known to perform well or, as we do, a human baseline.

- **Crowdsourcing:** Crowdsourcing provides a relatively cheap alternative to evaluate how well word embedding models capture human semantics. We had success with a *triad task* format, a choice-task with an established track-record and solid theoretical grounding in psychology.

- **Human "Turing" test:** a given embeddings model—or any model of human semantics for that matter—can be said to approximate human semantics well if, on average, for any given cue, the model generates associations (nearest neighbors) that a human cannot systematically distinguish from human generated associations.

  Specifically, we define human performance as the point at which a human rater is indifferent between a computer and a human generated association.

Third, in terms of *instability*

- **Stability:** along with a non-convex solution space, word embeddings methods have a lot of moving parts many of which introduce an element of randomness into the estimation. This produces additional variability beyond sampling error which, if unaccounted for, can lead to mistaken and non-replicable inferences.[36] To account for estimation-related instability we endorse estimating the same model several times, each with different randomly drawn initial word vectors and use an average of the distance metric of choice.[37] The good news, from our results at least, is that embeddings that perform well on the technical and human metrics tend to also be the most stable. Finally as an aside, the embeddings themselves should *not* be averaged as they lie in different spaces.

Fourth, in terms of algorithm (`GloVe` or `Word2Vec` (skip-gram))

- **`GloVe` vs. `Word2Vec` (skip-gram):** although `GloVe` is mathematically very similar to `Word2Vec`'s skip-gram architecture, in practice they will diverge, often quite substantially, in their mapping of the semantic space. `Word2Vec` benefits from a more careful filtering of the vocabulary (e.g. increasing the minimum count or setting a lower maximum number of words in the vocabulary) as it tends to *over* weight relatively rare terms (often misspellings). This is less of an issue with `GloVe` given it's modified weighting function. According to our human raters, `Word2Vec`, with an appropriately filtered vocabulary, performs as well as `GloVe`.

10 Discussion: Why do we get these results? And does any of it matter?

Why do we get the results we do? That is, why are pre-trained embeddings sometimes preferred to locally fit ones given that the latter are domain specific? And why do humans sometimes prefer human created neighbors, but sometimes prefer those generated by a statistical model? Answering these questions are beyond the scope of the current paper, but we can speculate a little.

[36] For replication purposes, it is possible to set a seed when estimating embeddings however, this comes at the cost of not being able to parallelize which will significantly increase computational costs.

[37] Note, all packages initialize word vectors randomly so this simply amounts to estimating the same model several times.

On the issue of poor local fits, one possibility is simply a lack of data. That is, corpora being used for such fits are too small to exhibit the helpful smoothing that a very large corpus (like Wikipedia) would allow. Thus, even with weighting down rare terms, small corpora have idiosyncratic co-occurrences (perhaps even typos) that are unappealing to our human coders.

As to the core Turing issue—that humans sometimes prefer model output rather than that of other humans—we suspect this is fundamentally connected to issues of sampling. In particular, even though we remove outlier human suggestions, it may nonetheless be the case that a model aggregating over millions of words is more reasonable, on average. Meanwhile, one pathology of embeddings is that they can quickly become out of date (e.g. until recently "`Trump`" would be a word with nearest neighbors pertaining to real estate or casinos, rather than the presidency).

A broader question with these results is a frank one: does any of this matter? That is, when we say that a given model specification is undesirable, is there any evidence that an end user would suffer in terms of the merits of their study should they go down that route? Again this is beyond the scope of the current paper, but in Supporting Information F we give an example of "negative" consequences.

11 Conclusions

Word embeddings in their modern scalable form have captured the attention of academia and industry, with the papers introducing `Word2Vec` and `GloVe` accumulating tens of thousands of citations since publication just five or six years ago. Early indications are that their influence will soon be felt in social science. As always, more methodological options are better, but it is important that we understand what they can do for us and what they cannot.

Here, we focused on "optimal" specifications, for which we used multiple criteria both technical and substantive on what we deem to be a representative corpus—*The Congressional Record*. This included a new "Turing"-style test, which pits models (including cheap, pre-trained ones) against humans, to discover what (other) humans prefer. For the domain of political science, we have good news: by all the criteria we used, off-the-shelf pre-trained embeddings work very well relative to—and sometimes better than—both human coders, and more involved locally trained models. Furthermore, locally-trained embeddings perform similarly—with noted exceptions—across specifications which should reduce end-user angst about their parameter choices. The general form of these findings extend to historical and non-English texts. Lastly and with caveats, human coders do not systematically prefer one of the two popular models, `GloVe` and `Word2Vec`, with both showing similar performance.

Our efforts here have dealt with a broad but necessarily limited number of possible options. Of course, other researchers will care about different substantive concepts and technical specifications. Irrespective of those particularities however, our work-flow here will be useful. Finally, of course, we have focused on *relative* performance: we have not studied whether embeddings are interesting or useful *per se* for understanding behavior, events and so on. We leave such questions for future work.

References

Agirre, Eneko, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca and Aitor Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics pp. 19–27.

Antoniak, Maria and David Mimno. 2018. "Evaluating the stability of embedding-based word similarities." *Transactions of the Association for Computational Linguistics* 6:107–119.

Baroni, Marco, Georgiana Dinu and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vol. 1 pp. 238–247.

Beck, Nathaniel, Gary King and Langche Zeng. 2000. "Improving quantitative studies of international conflict: A conjecture." *American Political Science Review* 94(1):21–35.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent and Christian Jauvin. 2003. "A neural probabilistic language model." *Journal of machine learning research* 3(Feb):1137–1155.

Benoit, Kenneth, Drew Conway, Benjamin E Lauderdale, Michael Laver and Slava Mikhaylov. 2016. "Crowd-sourced text analysis: Reproducible and agile production of political data." *American Political Science Review* 110(2):278–295.

Benoit, Kenneth, Kevin Munger and Arthur Spirling. 2019. "Measuring and explaining political sophistication through textual complexity." *American Journal of Political Science* 63(2):491–508.

Blei, David M, Andrew Y Ng and Michael I Jordan. 2003. "Latent dirichlet allocation." *Journal of machine Learning research* 3(Jan):993–1022.

Bolukbasi, Tolga, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*. pp. 4349–4357.

Chang, Jonathan, Sean Gerrish, Chong Wang, Jordan L Boyd-Graber and David M Blei. 2009. Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*. pp. 288–296.

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*. pp. 1–6.

Collobert, Ronan and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*. ACM pp. 160–167.

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu and Pavel Kuksa. 2011. "Natural language processing (almost) from scratch." *Journal of Machine Learning Research* 12(Aug):2493–2537.

Denny, Matthew J and Arthur Spirling. 2018. "Text preprocessing for unsupervised learning: why it matters, when it misleads, and what to do about it." *Political Analysis* 26(2):168–189.

Dingwall, Nicholas and Christopher Potts. 2018. "Mittens: an extension of glove for learning domain-specialized representations." *arXiv preprint arXiv:1803.09901* .

Faruqui, Manaal, Jesse Dodge, Sujay K Jauhar, Chris Dyer, Eduard Hovy and Noah A Smith. 2014. "Retrofitting word vectors to semantic lexicons." *arXiv preprint arXiv:1411.4166* .

Faruqui, Manaal, Yulia Tsvetkov, Pushpendre Rastogi and Chris Dyer. 2016. "Problems with evaluation of word embeddings using word similarity tasks." *arXiv preprint arXiv:1605.02276* .

Fast, Ethan, Binbin Chen and Michael S Bernstein. 2016. Empath: Understanding topic signals in large-scale text. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM pp. 4647–4657.

Firth, John Rupert. 1957. *Studies in linguistic analysis*. Wiley-Blackwell.

Garg, Nikhil, Londa Schiebinger, Dan Jurafsky and James Zou. 2018. "Word embeddings quantify 100 years of gender and ethnic stereotypes." *Proceedings of the National Academy of Sciences* 115(16):E3635–E3644.

Garimella, Aparna, Carmen Banea and Rada Mihalcea. 2017. Demographic-aware word associations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pp. 2285–2295.

Gentzkow, Matthew, J.M. Shapiro and Matt Taddy. 2018. "Congressional Record for the 43rd-114th Congresses: Parsed Speeches and Phrase Counts.".
URL: *https://data.stanford.edu/congress_text*

Halpern, David and Pedro Rodriguez. 2018. Partisan representations: Partisan differences in semantic representations and their role in attitude judgments. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*. pp. 445–450.

Hamilton, William L, Jure Leskovec and Dan Jurafsky. 2016*a*. Cultural shift or linguistic drift? comparing two computational measures of semantic change. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*. Vol. 2016 NIH Public Access p. 2116.

Hamilton, William L, Jure Leskovec and Dan Jurafsky. 2016*b*. "Diachronic word embeddings reveal statistical laws of semantic change." *arXiv preprint arXiv:1605.09096* .

Harris, Zellig S. 1954. "Distributional structure." *Word* 10(2-3):146–162.

Harris, Zellig S. 1970. Distributional structure. In *Papers in structural and transformational linguistics*. Springer pp. 775–794.

Islam, Aylin Caliskan, Joanna J Bryson and Arvind Narayanan. 2016. "Semantics derived automatically from language corpora necessarily contain human biases." *CoRR, abs/1608.07187* .

Khodak, Mikhail, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon Stewart and Sanjeev Arora. 2018. "A la carte embedding: Cheap but effective induction of semantic feature vectors." *arXiv preprint arXiv:1805.05388* .

Kozlowski, Austin C, Matt Taddy and James A Evans. 2018. "The geometry of culture: Analyzing meaning through word embeddings." *arXiv preprint arXiv:1803.09288* .

Kulkarni, Vivek, Rami Al-Rfou, Bryan Perozzi and Steven Skiena. 2015. Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee pp. 625–635.

Landauer, Thomas K and Susan T Dumais. 1997. "A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge." *Psychological review* 104(2):211.

Levy, Omer, Yoav Goldberg and Ido Dagan. 2015. "Improving distributional similarity with lessons learned from word embeddings." *Transactions of the Association for Computational Linguistics* 3:211–225.

Lund, Kevin and Curt Burgess. 1996. Hyperspace analogue to language (HAL): A general model semantic representation. In *Brain and Cognition*. Vol. 30 ACADEMIC PRESS INC JNL-COMP SUBSCRIPTIONS 525 B ST, STE 1900, SAN DIEGO, CA … pp. 5–5.

McCarthy, Diana and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*. Association for Computational Linguistics pp. 48–53.

Melamud, Oren, David McClosky, Siddharth Patwardhan and Mohit Bansal. 2016. “The role of context types and dimensionality in learning word embeddings.” *arXiv preprint arXiv:1601.00893* .

Mikolov, Tomas, Kai Chen, Greg Corrado and Jeffrey Dean. 2013. “Efficient estimation of word representations in vector space.” *arXiv preprint arXiv:1301.3781* .

Mimno, David and Laure Thompson. 2017. The strange geometry of skip-gram with negative sampling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. pp. 2873–2878.

Nayak, Neha, Gabor Angeli and Christopher D Manning. 2016. Evaluating word embeddings using a representative suite of practical tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*. pp. 19–23.

Patel, Kevin and Pushpak Bhattacharyya. 2017. Towards Lower Bounds on Number of Dimensions for Word Embeddings. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Vol. 2 pp. 31–36.

Pennington, Jeffrey, Richard Socher and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. pp. 1532–1543.

Pierrejean, Bénédicte and Ludovic Tanguy. 2017. Towards Qualitative Word Embeddings Evaluation: Measuring Neighbors Variation. In *Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. pp. 32–39.

Pierrejean, Bénédicte and Ludovic Tanguy. 2018. Predicting word embeddings variability. In *The seventh Joint Conference on Lexical and Computational Semantics*. pp. 154–159.

Quinn, Kevin M, Burt L Monroe, Michael Colaresi, Michael H Crespin and Dragomir R Radev. 2010. “How to analyze political attention with minimal assumptions and costs.” *American Journal of Political Science* 54(1):209–228.

Rheault, L, Beelen K, Cochrane C and Hirst G. 2016. “Measuring Emotion in Parliamentary Debates with Automated Textual Analysis.” *PLOS ONE* 11(12).

Rheault, Ludovic and Christopher Cochrane. 2019. “Word Embeddings for the Analysis of Ideological Placement in Parliamentary Corpora.” *Political Analysis* pp. 1–22.

Roberts, Margaret E, Brandon M Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson and David G Rand. 2014. “Structural topic models for open-ended survey responses.” *American Journal of Political Science* 58(4):1064–1082.

Rodman, Emma. 2019. “A Timely Intervention: Tracking the Changing Meanings of Political Concepts with Word Vectors.” *Political Analysis* pp. 1–25.

Sahlgren, Magnus. 2006. The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces PhD thesis.

Schnabel, Tobias, Igor Labutov, David Mimno and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pp. 298–307.

Turing, Alan. 1950. “Computing machinery and intelligence.” *Mind* 59(236):433.

Wendlandt, Laura, Jonathan K Kummerfeld and Rada Mihalcea. 2018. “Factors Influencing the Surprising Instability of Word Embeddings.” *arXiv preprint arXiv:1804.09692* .

Wittgenstein, Ludwig. 1953. “Philosophical investigations.” *London, Basic Blackw* .

Supporting Information

A Task Wording

Context Words

A famous maxim in the study of linguistics states that:

You shall know a word by the company it keeps. (Firth, 1957)

This task is designed to help us understand the nature of the ''company'' that words ''keep'': that is, their CONTEXT.

Specifically, for a CUE WORD, its CONTEXT WORDS include words that:

- Tend to occur in the vicinity of the CUE WORD. That is, they are words that appear close to the CUE WORD in written or spoken language.

AND/OR

- Tend to occur in similar situations to the CUE WORD in spoken and written language. That is, they are words that regularly appear with other words that are closely related to the CUE WORD.

For example, CONTEXT WORDS for the cue word COFFEE include:

1. *cup* (tends to occur in the vicinity of COFFEE).
2. *tea* (tends to occur in similar situations to COFFEE, for example when discussing drinks).

Click ''Next'' to continue

Next

(a) Context Words

Task Description

For each iteration of the task (13 in total including trial and screener tasks):

1. You will be given a cue word (top center of the screen) and two candidate context words (on either side of the cue word).
2. Please select the candidate context word that you find best meets the definition of a context word.
3. We are especially interested in context words likely to appear in **political discourse.**
4. If both are reasonable context words, please select whichever you find most intuitive.
5. You must select **one and only one** of the two candidate context words.

Keep in mind, some iterations are for screening purposes. These are tasks for which there is clearly a correct answer.

Wrong answers in these screening tasks will automatically end your participation so **be sure to read carefully.**

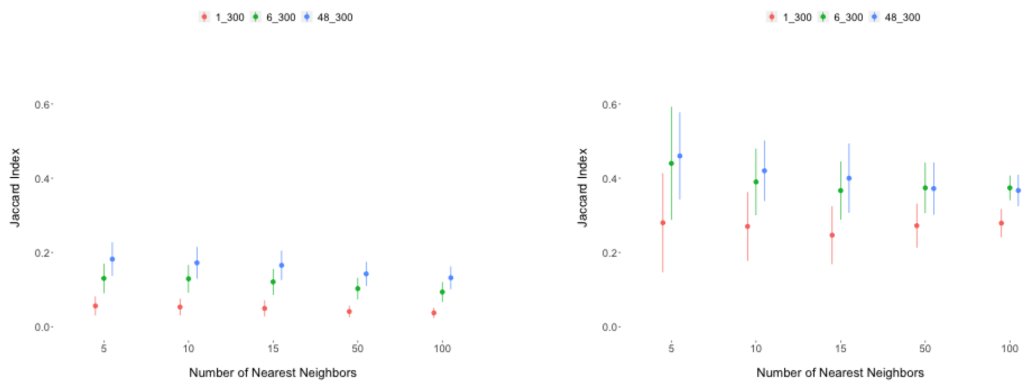
The trial task that follows is meant for you to practice. Like screening tasks, the trial task has a correct answer.

Click ''Next'' to continue to the trial runs

Next

(b) Task Instructions

Figure 9: Instructions



(a) Random Queries

(b) Politics Queries

Figure 10: Jaccard Index Between Pre-Trained and Local Models

B Jaccard Index

To further evaluate the correspondence between pre-trained embeddings and local models we use the average Jaccard-index —also known as the intersection over the union (IoU)—over the set of random and politics queries (Sahlgren, 2006; Pierrejean and Tanguy, 2017). The Jaccard-index between two models for a given query corresponds to the number of common nearest neighbors in the top N (the intersect of the two sets), over the union of the two sets. For example, take the following two sets of top 5 nearest neighbors for the query term `democracy`: $A = \{$ `freedom`, `democratic`, `ideals`, `vibrant`, symbol$\}$ and $B = \{$`freedom`, `democratic`, `dictatorship`, `democratization`, `socialism`$\}$. Given two nearest neighbors in common, the IoU is $\frac{|A \cap B|}{A \cup B} = \frac{2}{8} = 0.25$. Figure 10 plots the Jaccard-index, for various values of N, between GloVe pre-trained embeddings and several local models varying by window size. Unlike with the Pearson correlations we do not subset the respective vocabularies. As with the Pearson correlations, we observe larger values as window-size increases but with decreasing returns.

C Window Size and Discrimination for a real corpus

The claim is that larger windows mean that we can better discriminate between term meanings. We looked at the evidence for this on our *Congressional Record* corpus. To assess the claim we first set up a set of 'true negatives'—words that should be (fairly) unrelated. In particular for us, these are just random pairs of words from our corpus. We also evaluated how the average distance varies for 'true positives', that is words that are in fact the same. To assess this we sampled 100 words from the vocabulary. Suppose `congress` is one of those 100 words. We then...

1. tag half of the appearances (randomly selected) of `congress` in the corpus as `congress_tp`. So, if `congress` appears 10,000 times, in our transformed corpus it will appear as `congress` 5000 times, and `congress_tp` 5000 times.

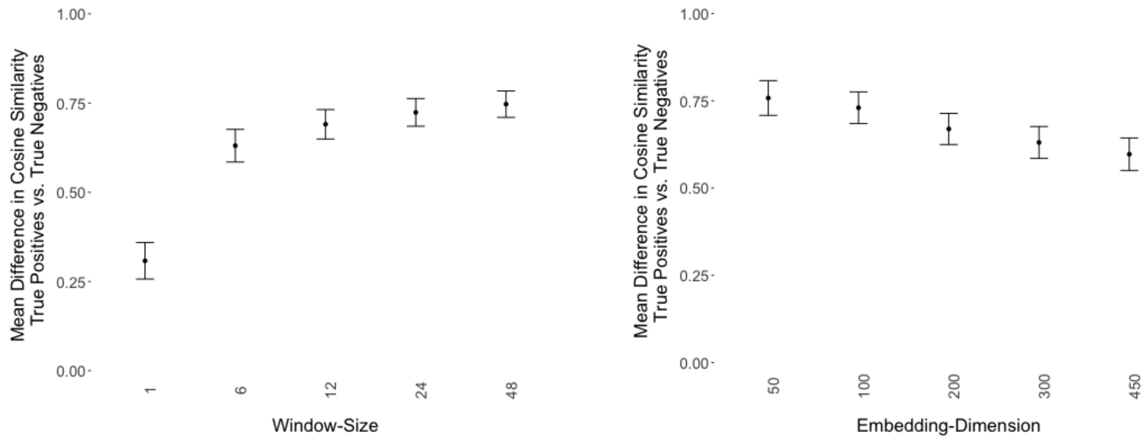
2. estimate a set of embeddings with the vocabulary including both `congress` and `congress_tp`.

Now we have an embedding for `congress` and `congress_tp`. These should be close in embedding space, since they are the same word albeit (randomly) half the incidences have been given a different token (hence we call them "true positives"). We interpret *how* close they are as measure of performance.

In Figure 11a we plot the mean difference in similarity terms between the true positives and the true negatives. When this number is large, we are saying similar words look much more similar to one another than random words (i.e. our model is performing well). When this number is smaller, the model is telling us it cannot distinguish between words that are genuinely similar and words that are not. On the left of the figure, fixing the embedding dimensions at 300, we see that larger windows translate to bigger differences—i.e. the model performs better in terms of discrimination. We call this *meaningful separability*. As an aside, on the right of the figure, we see that for a fixed window-size of 6, increasing the number of dimensions actually causes the model to do worse.

D Pre-trained embeddings perform equally across subgroups for *Congressional Record*

Above we showed that overall `GloVe` pre-trained embeddings correlate highly with locally trained embeddings. Next we ask whether these correlations differ by party. Such biases can be prob-



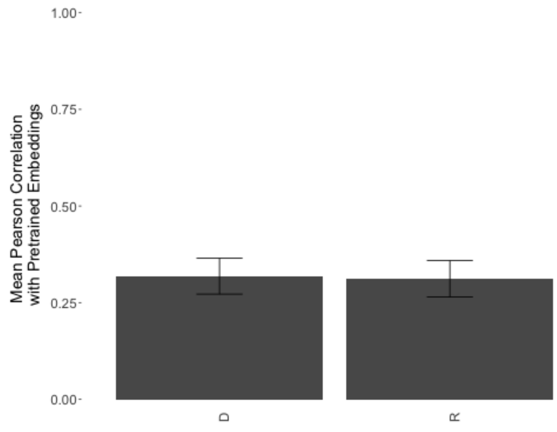
(a) Varying Window-Size (dimensions = 300) (b) Varying Dimensions (window-size = 6)

Figure 11: Mean Difference in Cosine Similarity True Positives vs. True Negatives

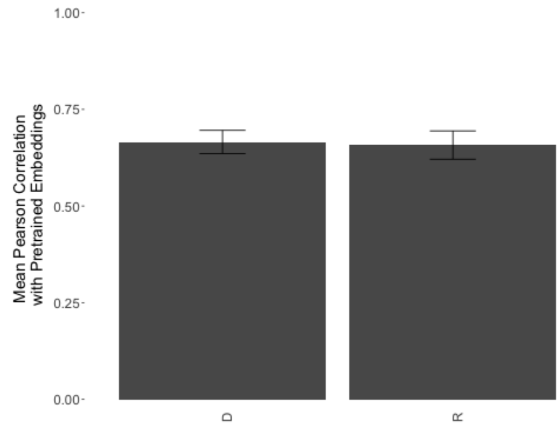
lematic if pre-trained embeddings are subsequently used to analyze texts and draw conclusions on the basis of party. To evaluate whether pre-trained embeddings exhibit bias we compare query search results based on pre-trained embeddings to results based on locally trained embeddings specific to each group (Democrat and Republican legislators). We say pre-trained embeddings exhibit bias—according to this metric—if query search results correlate significantly higher with the query search results of one group relative to the other.

This evaluation requires we estimate separate embeddings for each of these groups. To do so, we split the congressional corpus by party (Republican vs Democrat). We apply the same estimation framework as laid out in section 4 to each sub-corpora except we fix window-size and embedding dimension at 6 and 300 respectively.

Figures 12a and 12b display the main results of our evaluation for a random set of queries and our set of politics queries respectively. For neither set of queries do we find evidence of partisan bias—as defined here—in pre-trained embeddings. To be clear, this result does not mean that pre-trained embeddings do not exhibit common cultural biases —they do according to previous research Bolukbasi et al. (2016); Islam, Bryson and Narayanan (2016)— but rather that pre-trained embeddings —GloVe specifically— are equally correlated with party specific embedding models.



(a) Random Queries

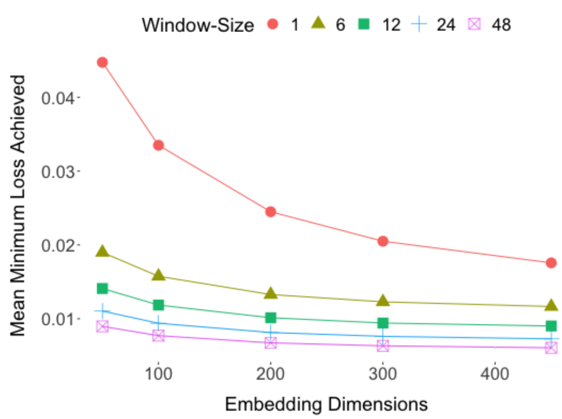


(b) Politics Queries

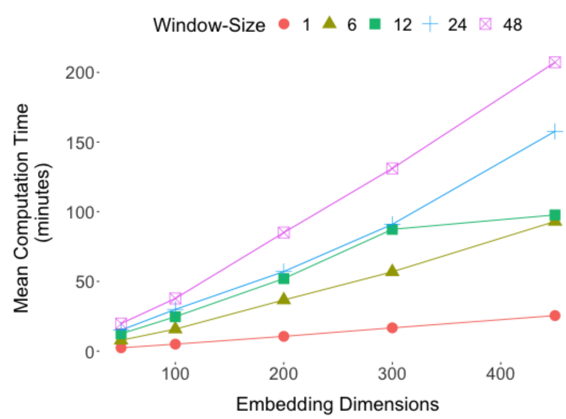
Figure 12: Pearson correlation of group embeddings with pre-trained `GloVe` embeddings.

E Other Corpora, Other Languages: Results

E.1 Technical Criteria

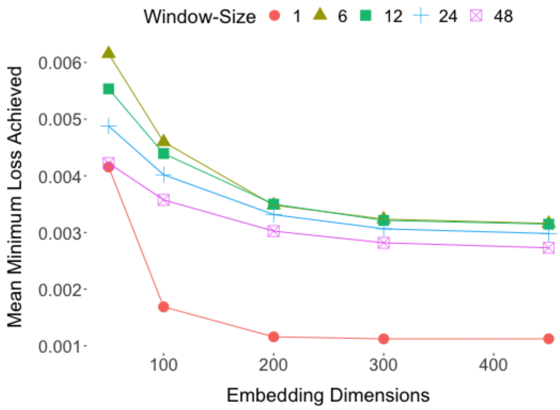


(a) Mean Minimum Loss Achieved

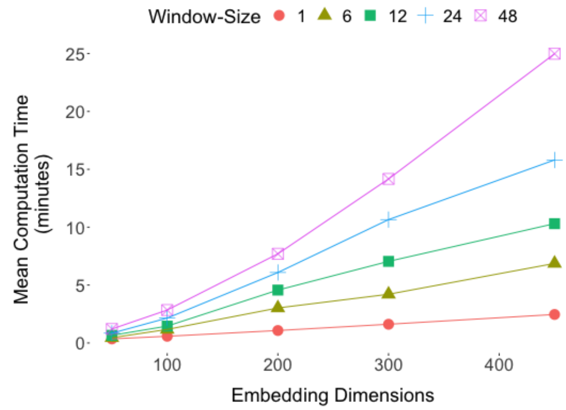


(b) Computation Time (minutes)

Figure 13: Technical Criteria: Parliamentary Speeches

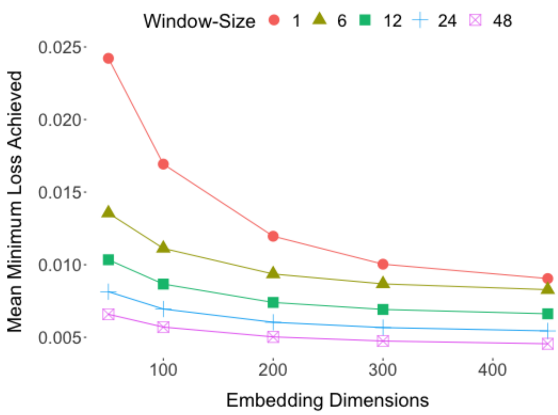


(a) Mean Minimum Loss Achieved

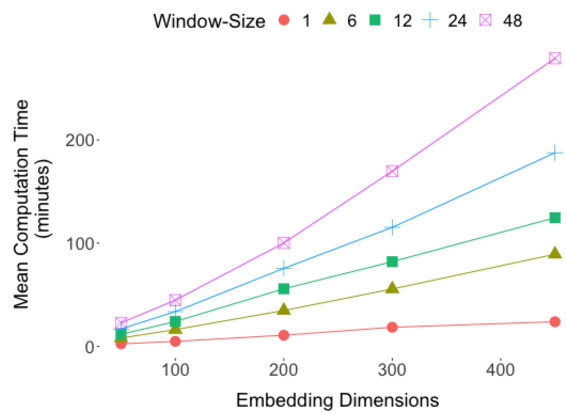


(b) Computation Time (minutes)

Figure 14: Technical Criteria: State of the Union Speeches

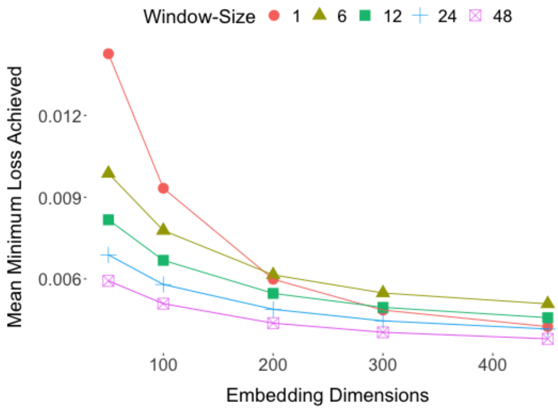


(a) Mean Minimum Loss Achieved

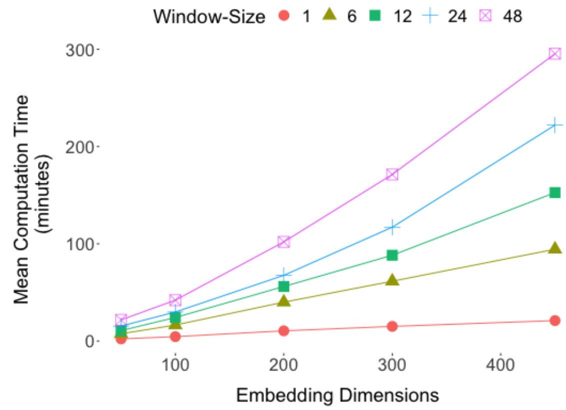


(b) Computation Time (minutes)

Figure 15: Technical Criteria: Spanish Corpus



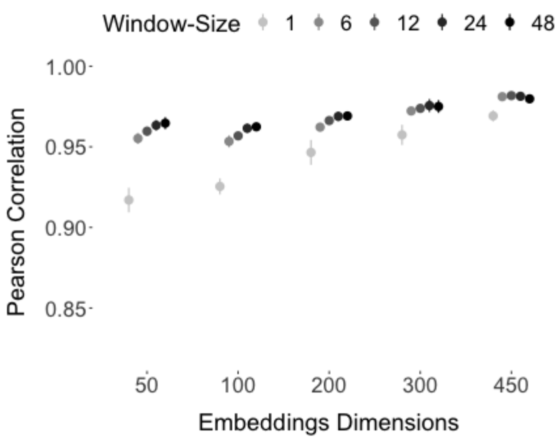
(a) Mean Minimum Loss Achieved



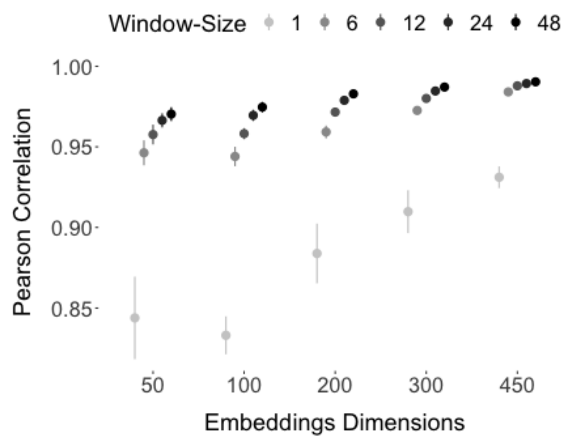
(b) Computation Time (minutes)

Figure 16: Technical Criteria: German Corpus

E.2 Stability

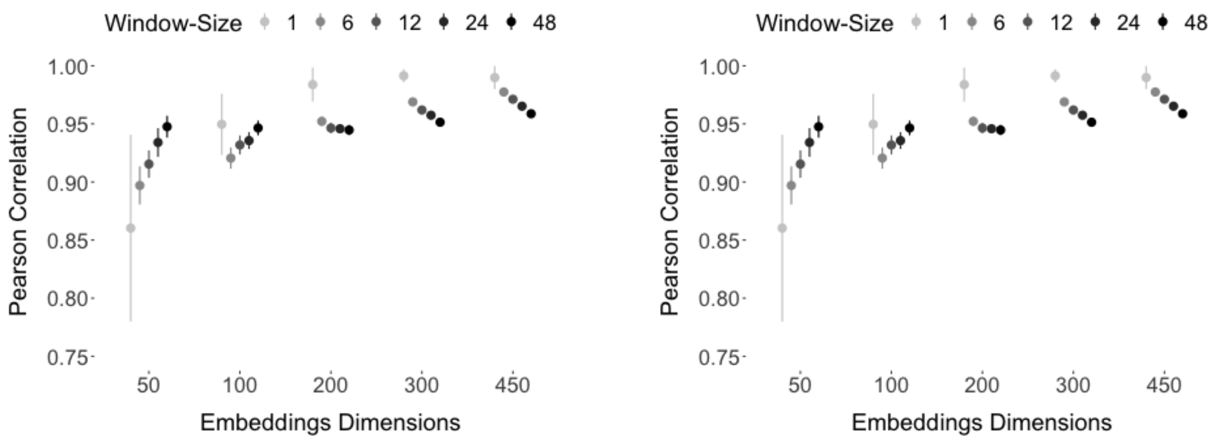


(a) Random Queries



(b) Politics Queries

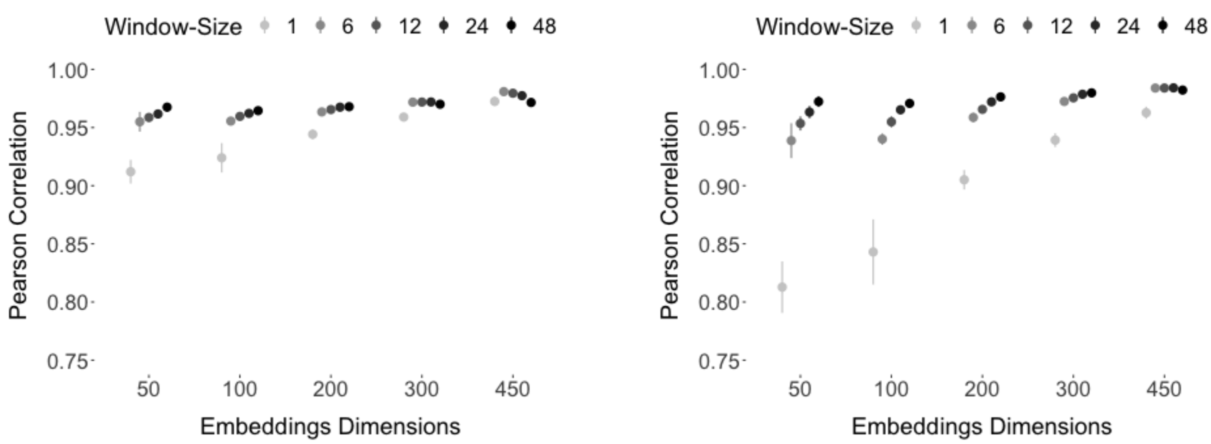
Figure 17: Stability Criteria: Parliamentary Speeches



(a) Random Queries

(b) Politics Queries

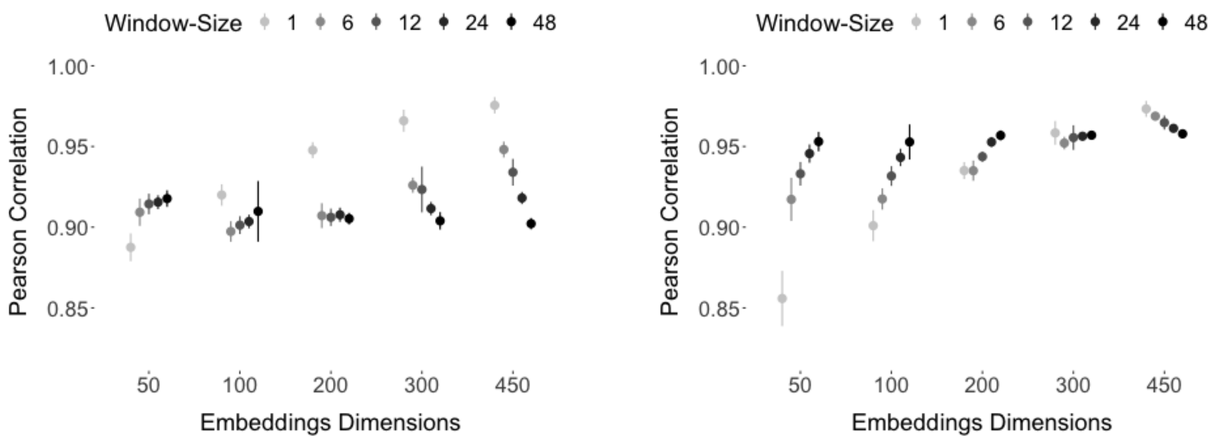
Figure 18: Stability Criteria: State of the Union Speeches



(a) Random Queries

(b) Politics Queries

Figure 19: Stability Criteria: Spanish Corpus

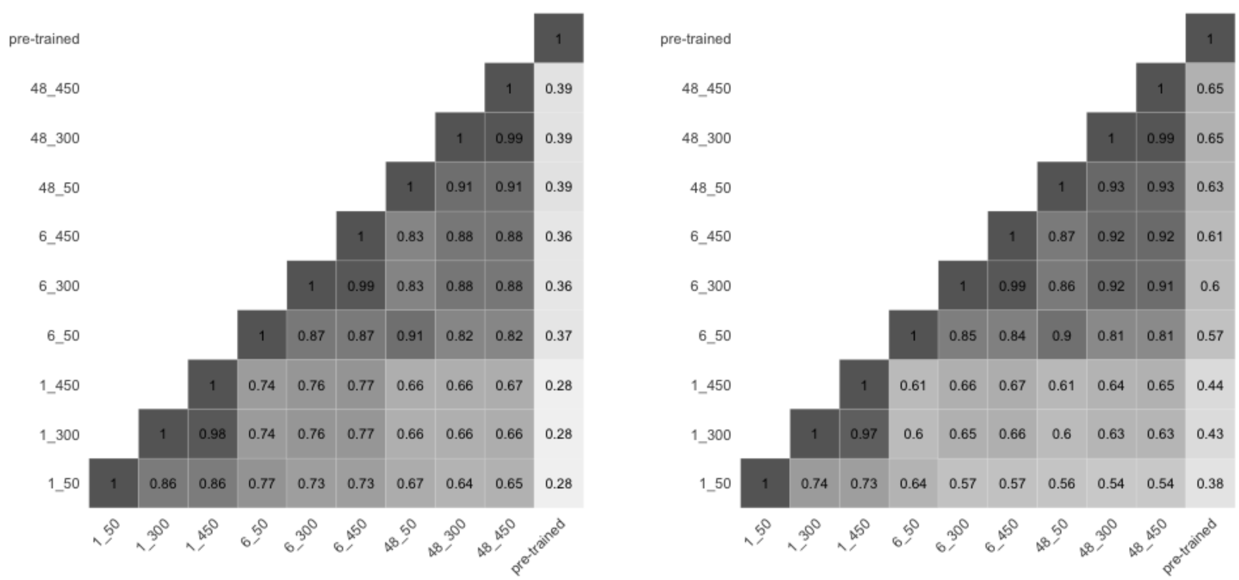


(a) Random Queries

(b) Politics Queries

Figure 20: Stability Criteria: German Corpus

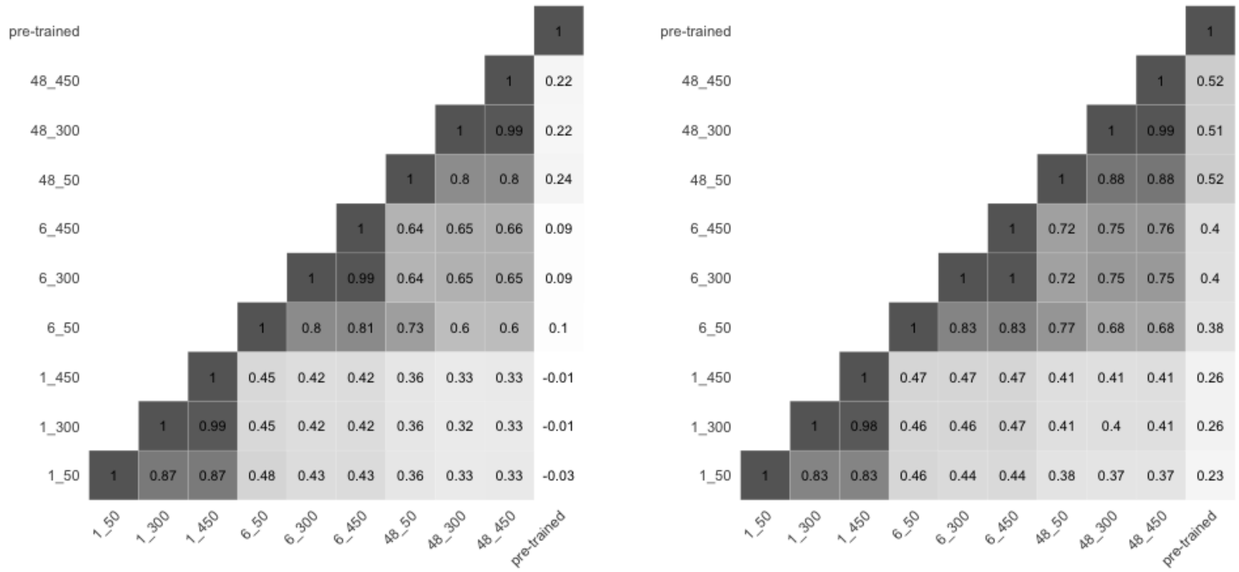
E.3 Query Search Ranking Correlation



(a) Random Queries

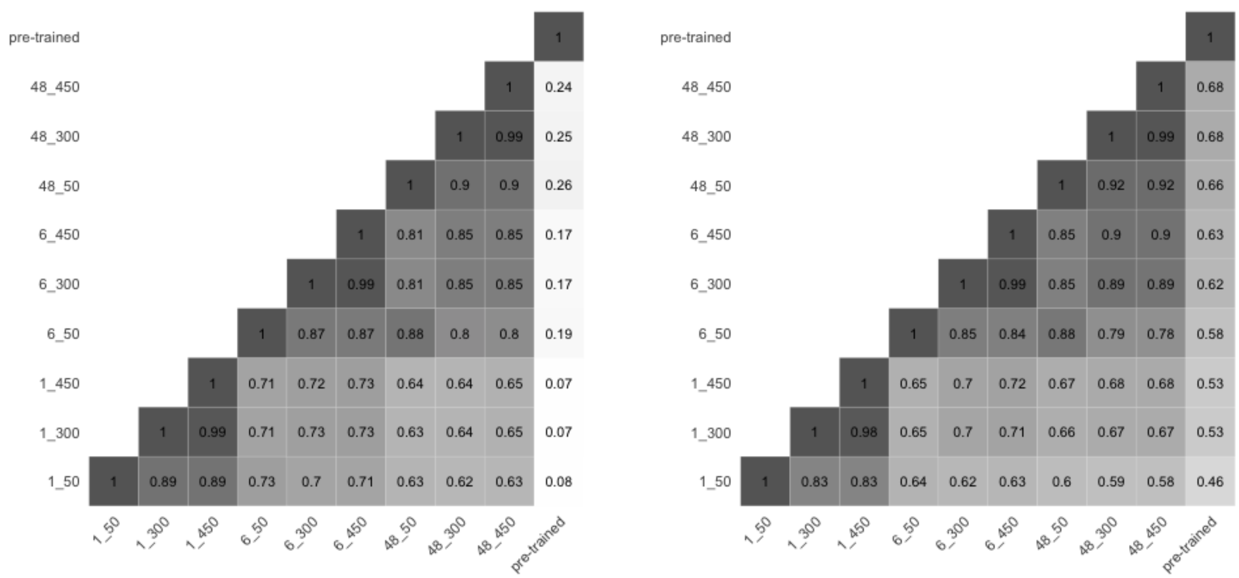
(b) Politics Queries

Figure 21: Query Search Ranking Criteria: Parliamentary Speeches



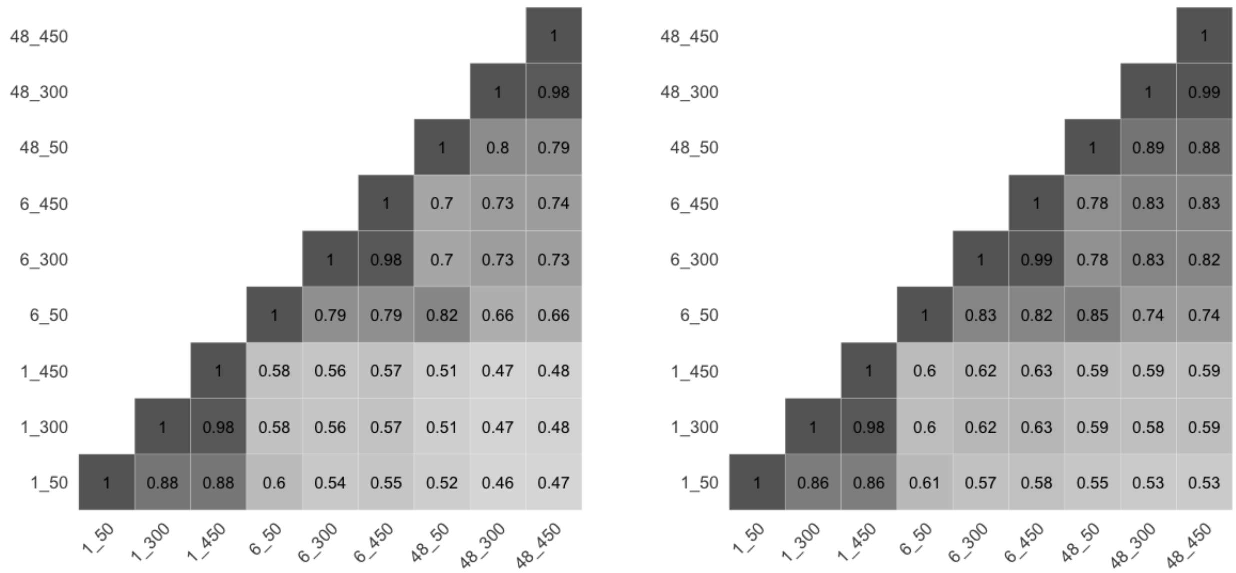
(a) Random Queries (b) Politics Queries

Figure 22: Query Search Ranking Criteria: State of the Union Speeches



(a) Random Queries (b) Politics Queries

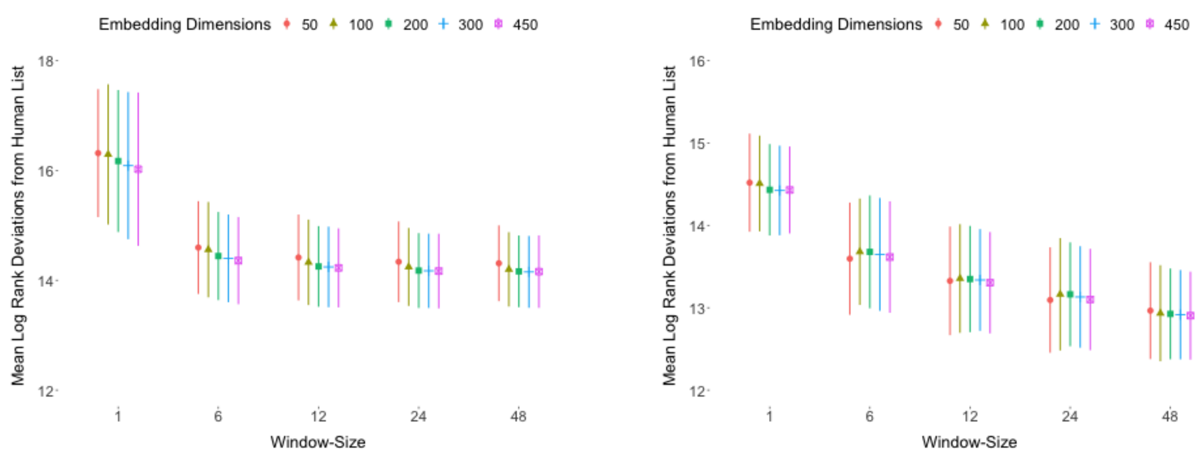
Figure 23: Query Search Ranking Criteria: Spanish Legislature



(a) Random Queries (b) Politics Queries

Figure 24: Query Search Ranking Criteria: German Legislature

E.4 Human Validation



(a) Parliamentary Speeches (b) State of the Union Speeches

Figure 25: Human Preferences-Log Rank Deviations

F What could possibly go wrong? Problems with using inappropriate embedding models

In Table 3 we report the top-5 nearest neighbors for two different specifications of `GloVe` models fit to two of our corpora. In the first two columns, we compare "immigration" for a model with a small window and short representation vector (1-50) with a more standard specification (6-300). This exercise is repeated for *Hansard* for the word "abortion".

The most immediate observation is that the representation and thus inference differs within corpus, depending on the specification. Thus, we see the 6-300 specification reports "naturalization", "illegal" and "INS" (Immigration and Naturalization Service) as the nearest neighbors for "immigration" in the *Congressional Record*, while the 1-50 reports "tort", "reform" and "laws". However, without reference to purpose, it is misleading to claim that one list is correct and one is incorrect. Topically, the 6-300 words do seem more appropriate; but that is what we would expect given previous results. Similarly, they might help us build a better dictionary, or locate search terms of interest. To reiterate though, the 1-50 neighbors are not "wrong" *per se*. It is more that the words are capturing a different sense of context. One possibility here, for example, is that the 1-50 context is about legislative issues arising at the same time (or pushed by the same legislators) as "immigration" was being discussed. Similarly, when we switch to *Hansard* we see that the topical context of "abortion" is best captured by the 6-300 model. But the 1-50 model perhaps captures some temporal context: the decriminalization of abortion in the UK occurred in 1968, and is approximately contemporaneous with ending "conscription" (1960) and the beginning of "fluoridation" of the public water supply, along with changes to "insolvency" law (1976).

Congressional Record "immigration" `1-50`	`6-300`	*Hansard* "abortion" `1-50`	`6-300`
tort	naturalization	extradition	abortions
reform	illegal	insolvency	contagious
laws	(ins) INS	arbitration	contraception
bankruptcy	reform	conscription	clinics
ethics	customs	fluoridation	pregnancy

Table 3: Comparing top-5 nearest neighbors across `GloVe` specifications. Note that 6-300 typically returns better topical context words.

Note that we have candidly "cherry-picked" our comparisons here. That is, for other words, the differences between specifications are minor. Nonetheless, as a *possibility* result, our findings stand.