Measuring Document Similarity With Weighted Averages of Word Embeddings

Bryan Seegmiller
Kellogg School of Management

Dimitris Papanikolaou
Kellogg School of Management and NBER

Lawrence D. W. Schmidt
MIT Sloan School of Management

September 2022

Abstract

We detail a methodology for estimating the textual similarity between two documents while accounting for the possibility that two different words can have a similar meaning. We illustrate the method's usefulness in facilitating comparisons between documents with very different formats and vocabularies by textually linking occupation task and industry output descriptions with related technologies as described in patent texts; we also examine economic applications of the resultant document similarity measures. In a final application we demonstrate that the method also works well relative to alternatives for comparing documents within the same domain by showing that pairwise textual similarity between occupations' task descriptions strongly predicts the probability that a given worker will transition from one occupation to another. Finally, we offer some suggestions on other potential uses and guidance in implementing the method.

In recent years there has been a proliferation of newly available textual datasets that are of interest to economists, combined with improvements in computational capacity and more readily accessible open-source programming libraries for textual analysis. Together these trends have made possible the creation of new text-based quantitative measures of economic quantities which wouldn't have been feasible for many economists even just a few years ago. Although this new wealth of data and tools provides new opportunities, it also brings about new challenges as researchers seek the best methods to apply to their particular research questions.

In this paper we propose a simple method that researchers can use out-of-the box to compare textual similarity across documents, especially when the documents being compared come from different domains and have tend to have very dissimilar formats and vocabulary. This situation may be of particular interest to economic historians interested in making textual comparisons across different classes of historical documents. Since purely numerical measures of economic quantities become less readily available further back in time, the comparative breadth of information available in historical texts means textual analysis methods like ours offer a unique opportunity to create new and economically interesting historical measures. While we think of this 'cross-domain' comparison as being the typical use case, we also provide a simple application of our method, along with an accompanying code repository, which shows that our approach also appears well-suited for comparing documents that come from within the same domain.

Many economic applications require estimating a notion of textual 'similarity' between two documents. Examples of these can include: identifying novel and impactful patents by computing the similarity to earlier or subsequent innovations (Kelly, Papanikolaou, Seru, and Taddy, 2020); matching patent documents to occupation task descriptions (Webb, 2020; Kogan, Papanikolaou, Schmidt, and Seegmiller, 2022); evaluating the similarity between firms' business operations to identify close competitors (Hoberg and Phillips, 2016); or measuring the similarity between scientific articles and course syllabi (Biasi and Ma, 2020). In each of these examples, researchers need a quantitative metric that indicates whether two documents are close in terms of meaning. This quantitative similarity metric can then be aggregated to create economic measures of interest: valuable breakthrough patents, occupational exposure to displacing technologies, the amount of product market competition a firm faces, or the extent of knowledge diffusion between academic research and school curricula. Naturally, since computers and humans understand language differently, this objective has challenges. Our goal here is to highlight one approach that can successfully navigate some of these difficulties.

Our method can be succinctly described as follows. First, we quantitatively represent individual words with word embeddings (also called word vectors), which are geometric representations of word meanings, and were first introduced to the natural language processing literature by

Mikolov, Sutskever, Chen, Corrado, and Dean (2013). There are numerous ways of estimating word embeddings, but the common thread is that they result in vector representations of word meanings where similar or related words are geometrically 'close' to one another, in the sense that they would be expected to have high cosine similarity with one another. Next, to go from a quantitative representation of individual words to complete documents, we simply take a weighted average of the word embeddings from the constituent terms in a given document, resulting in a document vector. We then measure the textual similarity between two documents by taking the cosine similarity between their corresponding document vectors. Because word embeddings are dense vectors, and may share some common components, these similarity scores tend to nearly always be strictly positive. Accordingly, we suggest a few adjustments to raw cosine similarity scores that can be made if researchers want to impose sparsity in the distribution of document textual relatedness.

The key advantage of using word embeddings is that they allow for words to be similar, without requiring exact overlap. In principle researchers can estimate their own word embeddings.[1] This is likely to be more successful when comparing documents coming from the same class of documents, and when there are many documents available (for example, comparing a firm's 10K filings versus other firms' 10K filings). However, in this case even simpler routines, like counting word overlap, may also be employed successfully.[2]

By contrast, our method is especially useful when documents are being compared across disparate domains, where it's unlikely that exact overlap in terminology can be leveraged successfully. In this case, training word embeddings to fit one type of documents can generalize poorly when comparing across document types. Estimation of word embeddings can also be computationally intensive, and often takes a large amount of information to obtain a good fit, which may not be possible when a researcher has a relatively small set of documents at her disposal. Accordingly, we *intentionally* do not estimate our own word embeddings in this paper. Instead, we use pre-trained word embeddings. There are many examples of pre-trained word embeddings that are readily available for free, and have been trained by professional natural language processing researchers on billions of documents capturing words' common usage in the English language. An added advantage of this approach is that it reduces the barrier to entry for our method considerably, as estimating embeddings successfully can be a challenging task for an economist just getting started in textual analysis.

In the remainder of the paper we explain our method in more detail and contrast it with alternative approaches. We then discuss a few example applications of our method. These include comparing occupation tasks or industry descriptions with patent texts, which are the type of

[1] This has been done in economics research in different contexts than we focus on in this paper: see Atalay, Phongthiengtham, Sotelo, and Tannenbaum (2020) or Hansen, Ramdas, Sadun, and Fuller (2021), for example.

[2] For example, this simpler approach is taken in Kelly et al. (2020) and Hoberg and Phillips (2016), who respectively compare patents to other patents and firms' 10K product descriptions with other firms' product descriptions.

"cross-domain" comparisons where we expect our measure to be most fruitful. The measure clearly sorts occupations or industries into their most related patents very well. This sorting property of our measure implies that our approach could be fruitfully applied when a researcher wants to find other documents that are related to a central document which summarizes a concept of interest.

We also include a simple example, with an accompanying code repository, which shows that our method can also be used for within-domain comparisons (here, comparing task descriptions of occupations with other occupations). Finally, we discuss potential extensions on the method and other settings of economic interest where our approach might be successfully applied.

A challenge for unsupervised textual similarity measures like ours is that there is not usually a well-defined criterion to determine if similarity scores are "correct" in some clear, objective way. Instead, we provide these examples to give a picture of how our measure organizes the data, and we find that the results are substantially intuitive and economically meaningful. The within-domain comparison of occupation-by-occupation textual similarity also provides a natural setting for analyzing our measure's performance in a somewhat "objective" way. In particular, we find that our measure of occupation-by-occupation task description similarity can predict the probability that a worker switches between two given occupations as well or even a bit better than a benchmark bag-of-words method—despite the fact that we view this type of within-domain comparison as not necessarily being our measure's comparative advantage. Our measure also performs substantially better than document similarities computed from a Latent Dirichlet Allocation (Blei, Ng, and Jordan, 2003) topic model directly trained on this data, underscoring the difficulties of training models to fit to a narrow setting when the amount of text information is not sufficiently large. Since it seems natural to assume that workers can more readily move between occupations with related job tasks, this stands as a nice sanity check on our method's performance; it also suggests that our method can be readily applied in other settings beyond the cross-domain use case that is our primary focus.

1 Measuring Document Similarity While Accounting for Synonyms

To illustrate why our method can be useful, we first contrast it with probably the simplest and most common approach for measuring textual similarity between documents, which is to count the number of words they share in common. One way to do this is to convert each document into a sparse 0/1 vector, where element of the vector indicates whether a particular word appears in the document. Slightly more sophisticated approaches will weight words differently, depending on how frequently they appear within this particular document and how often they show up across all documents.

Using this vector representation of document term counts, we can then compute a distance measure between two documents as the cosine similarity between each vector of (potentially weighted) term counts:

$$\text{Sim}_{i,j} = \frac{V_i}{||V_i||} \cdot \frac{V_j}{||V_j||} \tag{1}$$

Here V_i and V_j denote the vector of (potentially weighted) terms counts for documents i and j. It is common to remove from the document a pre-defined set of terms, known as "stop words", that carry little semantic information (like prepositions and conjunctions) before forming V_i and V_j. This approach is often referred to as the 'the bag-of-words' approach, and has been used successfully in many settings.[3] This approach can be successful if the documents i and j use similar vocabulary. For example, Kelly et al. (2020) use a variant of this approach to construct measures of patent novelty and impact based on pairwise distance measures between patent documents. Since patent documents have a structure and a legalistic vocabulary that is reasonably uniform, this approach works quite well for patent-by-patent comparisons.

However, this approach is less suited when the two documents i and j come from different sources and often use different vocabulary. In this case, if we were to use the bag-of-words approach, then the resulting vectors V_i and V_j are likely to be would highly sparse with most elements equal to zero, and therefore bias the distance measure (1) to zero. The root cause of the problem is that the distance measure in (1) has no way of accounting for words with similar meanings. For example, consider a set of two documents, with the first document containing the words 'dog' and 'cat' and the other containing the words 'puppy' and 'kitten'. Even though the two documents carry essentially the same meaning, the bag of words approach will conclude that they are distinct: the representation of the two documents is $V_1 = [1, 1, 0, 0]$ and $V_2 = [0, 0, 1, 1]$, which implies that the two documents are orthogonal, $\rho_{1,2} = 0$.

To overcome this challenge, we leverage recent advances in natural language processing that allow for synonyms.

1.1 Word Embeddings

The main idea behind this approach is to represent each *word* as a dense vector. The distance between two word vectors is then related to the likelihood these words capture a similar meaning. Researchers can estimate these word embeddings themselves, or they can use readily available word embedding vectors (e.g. Pennington, Socher, and Manning, 2014; Mikolov et al., 2013; Mikolov, Grave, Bojanowski, Puhrsch, and Joulin, 2018). To gain an intuition on how word embeddings work,

[3] See Gentzkow, Kelly, and Taddy (2019) for a summary of the bag of words approach and other text analysis methods.

we next briefly summarize how the (Pennington et al., 2014, commonly referred to as "GloVe") word embeddings are estimated. We focus on Pennington et al. (2014) because their objective function is easily recognized by economists as a non-linear weighted least squares optimization problem. Other commonly used embeddings methods, such as Mikolov et al. (2013)[known as "word2vec"] or Mikolov et al. (2018)["known as FastText"] also share the feature that related words will be geometrically close to one another, but they are estimated with a more complicated neural network architecture that is perhaps less instructive from an expositional standpoint. That being said, our approach does not require researchers to use one version of word embeddings versus another. Although we primarily focus on using the Pennington et al. (2014) GloVe embeddings in this paper, the method could apply equally to word2vec, FastText, or any other embedding procedure.

Denote the matrix X as a $V \times V$ matrix of word co-occurence counts obtained over a set of training documents, where V is the number of words in the vocabulary. Then $X_{i,j}$ tabulates the number of times word j appears in the context of the word i. Here, one must make a choice of how many words away from word i the algorithm should consider. For example, Pennington et al. (2014) GloVe embeddings use a symmetric 10 word window to determine context and then down-weight words that occur further away from the focal word—one word away receives weight 1, two words away receives weight 1/2, etc.

To clarify, denote $X_i = \sum_k X_{i,k}$ as the number of times any word appears in the context of word i, and the probability of word j occuring in the context of word i is $P_{i,j} \equiv X_{i,j}/X_i$. The goal of the word embedding approach is to construct a mapping $F(\cdot)$ from some d-dimensional vectors x_i, x_j, and $\tilde{x}_k$ such that

$$F(x_i, x_j, \tilde{x}_k) = \frac{P_{i,k}}{P_{j,k}} \tag{2}$$

Imposing some conditions on the mapping $F(\cdot)$, they show that a natural choice for modeling $P_{i,k}$ in (2) is

$$x_i^T \tilde{x}_k = \log(X_{i,k}) - \log(X_i) \tag{3}$$

Since the mapping should be symmetric for i and k they add "bias terms" (essentially i and k fixed effects) which gives

$$x_i^T \tilde{x}_k + b_i + b_k = \log(X_{i,k}) \tag{4}$$

Summing over squared errors for all pairwise combinations of terms yields the weighted least squares objective

$$\text{Min}_{x_i, \tilde{x}_k, b_i, b_k} \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{i,j}) \left(x_i^T \tilde{x}_k + b_i + b_k - \log(X_{i,j}) \right)^2 \tag{5}$$

Here the observation-specific weighting function $f(X_{i,j})$ equals zero for $X_{i,j} = 0$ (and does so rapidly

as $X_{i,j}$ gets small) so that the objective is well defined for zero values of $X_{i,j}$, and is constructed to avoid over-weighting rare occurrences or extremely frequent occurrences. The objective (5) is a highly-overidentified least squares minimization problem. Since the solution is not necessarily unique, the model is trained by randomly instantiating x_i and $\tilde{x}_k$ and performing gradient descent for a pre-specified number of iterations, yielding d-dimensional vector representations of a given word. Here, the dimensionality of the vector space d is a hyper-parameter; Pennington et al. (2014) find that $d = 300$ works well on word analogy tasks.

Since the objective function (5) is symmetric, it yields two vectors for word i, x_i and $\tilde{x}_i$, so the final word vector is taken as the average of the two. The ultimate output is a dense 300-dimensional vector for each word i that has been estimated from co-occurence probabilities and occupies a position in a word vector space such that the pairwise distances between words (i.e. using a metric like the cosine similarity) are related to the probability that the words occur within the context of one another and within the context of other similar words. Note that the basis for this word vector space is arbitrary and has no meaning; distances between word embeddings are only well-defined in relation to one another and a different training instance of the same data would yield different word vectors but very similar pairwise distances between word vectors.

1.2 From words to documents

At this point, we either obtained, or potentially estimated, a set of word vectors for each word in our dictionary (the word embeddings). The next step consists of aggregating these word vectors at the document level and then using them to construct measures of document similarity.

We represent the document as a dense vector X_i by computing the weighted average of the word vectors that belong in each document $x_k \in A_i$,

$$X_i = \sum_{x_k \in A_i} w_{i,k}\, x_k. \tag{6}$$

Just like when using the 'bag of words' approach, it is sensible to choose appropriate weights $w_{i,k}$ in order to emphasize important words in the document.

A common approach in selecting these weights is what is termed the 'term-frequency-inverse-document-frequency' (TF-IDF),

$$w_{i,k} \equiv TF_{i,k} \times IDF_k. \tag{7}$$

The first component of the weight, term frequency (TF), is defined as

$$TF_{i,k} = \frac{c_{i,k}}{\sum_j c_{i,j}}, \tag{8}$$

where $c_{i,k}$ denotes the count of the k-th word in document i—a measure of its relative importance within the document.

The inverse-document frequency is

$$IDF_k = \log\left(\frac{\text{\# of documents in sample}}{\text{\# of documents that include term } k}\right). \tag{9}$$

Thus, IDF_k measures the informativeness of term k by under-weighting common words that appear in many documents, as these are less diagnostic of the content of any individual document.

In brief, equation (7) overweighs words that are important for a document (terms that occur relatively frequently within a given document) and relatively uncommon (terms that do not occur commonly across all documents). Hence, two documents will be similar if they use many closely related but relatively uncommon words. If documents come from two distinct sources, and hence use different vocabulary, it may make sense to estimate the IDF weights separately within each document group. For example, Kogan et al. (2022) estimate the inverse-document-frequency for the set of patents and occupation tasks separately. As a result, the word 'abstract' or 'invention' receives very low weight if it originates in a patent document, since it is quite frequent.

1.3 Measuring document similarity

Armed with a vector representation of the document that accounts for synonyms, we next use the cosine similarity to measure the similarity between patent i and occupation j,

$$\text{Sim}_{i,j} = \frac{X_i}{||X_i||} \cdot \frac{X_j}{||X_j||} \tag{10}$$

This is the same distance metric as the bag of words approach in equation (1), except now X_i and X_j are dense vectors carrying a geometric interpretation akin to a weighted average of the semantic meaning of all nouns and verbs in the respective documents.

In addition to accounting for synonyms, this method has some clear computational advantages relative to the standard 'bag of words' methodology: the dimensionality of the vectors is much smaller than the 'bag of words' approach. The reason is we have represented each document as a vector of length $d = 300$. By contrast, in the 'bag of words' approach, the dimensionality of the vector is equal to the union of words between the two documents. If vocabulary is sufficiently large, this difference is meaningful and allows for faster computation of (10) relative to (1).

2 Alternative Choices for Measuring Document Similarity and When our Method is Likely to be Most Useful

Before moving onto applications of our method, we discuss alternative methods for measuring document similarity.

Standard Bag of Words

We already briefly discussed TF-IDF–weighted bag words before, which represents documents geometrically by sparse vectors of weighted counts of individual terms. The resulting document vectors are inherently high-dimensional because a set of documents will tend to contain many different words, and each word requires a separate entry in a bag of words document vector.

Clustering Methods for Dimensionality Reduction

While several types and variants of clustering routines exist, they all have the same basic feature of reducing the inherent high dimensionality of textual data that, as commonly occurs with bag of words approaches. The most popular clustering algorithms are latent dirichlet allocation (Blei et al., 2003) and latent semantic analysis (Deerwester, Dumais, Furnas, Landauer, and Harshman, 1990), both of which can be used to cluster words into groups or 'topics', where topics represent words that tend to co-occur with one another. Documents can then be represented by a much more low-dimensional vector of topic weights instead of a vector of weighted word counts.

Doc2vec

The doc2vec approach is introduced in Dai, Olah, and Le (2015), and is an extension on Mikolov et al. (2013) which estimates word embeddings directly for documents instead of individual words. Acemoglu, Yang, and Zhou (2021) use this technique to compare the textual similarity of Chinese academic research papers.

Contextual Word Embeddings

New advancements in embeddings methods (Devlin, Chang, Lee, and Toutanova, 2018; Liu, Ott, Goyal, Du, Joshi, Chen, Levy, Lewis, Zettlemoyer, and Stoyanov, 2019)—often referred to as BERT and RoBERTa, respectively—allow for word embeddings to be generated depending on the context of surrounding words. Because words sometimes have ambiguous meaning, this has the potential to improve the fit of embeddings-based methods. There are also contextual sentence embeddings

methods. While we don't apply contextual word embeddings in this paper, in principle this could be an extension on our basic procedure.

How do approaches compare?

Ash and Hansen (2022) show that variants of average word embeddings methods (pre-trained versus context specific training, different types of word embeddings, etc) tend to produce highly correlated document similarity scores when comparing firms' 10K risk disclosures. This suggests that in many contexts the embedding method chosen is not highly important. However, the correlation is lower for embeddings methods with clustering approaches or doc2vec. Ash and Hansen (2022) suggest that these differences may be unimportant if they are pure noise and measures can be aggregated to predict economic content similarly. In section 3.3 we include an example which shows that our measure performs well relative to other benchmark methods in a simple prediction exercise. However, simple inspections of linked documents from a given method can be highly informative, especially when there is not always an obvious right numerical metric for "correctly" measuring similarity. In sections 3.1 and 3.2 we provide a number of examples to highlight how our method sorts the data in a reasonable and intuitive ways, and show how it has been and can be applied in economically useful ways.

When is our method useful?

The above alternatives have strengths and disadvantages. The bag of words method is straightforward, but clearly suffers by not allowing for synonyms; the high dimensionality of the data can be problematic as well, as slight differences in document vocabulary can lead to very noisy measures of textual overlap. Meanwhile, doc2vec can be a powerful tool for creating geometric documents representations, but it requires large amounts of data and computing power. Another concern for doc2vec is its out-of-sample generalizability, since it is formed from neural networks which are prone to overfitting within the training sample and performance declines out of sample. LDA and LSA require a user to pre-commit to the number of topics; like doc2vec or in training one's own word embeddings, topic fits require a lot of data to be successful, and are also highly subject to the set of documents in which they are trained, and hence can likewise struggle with out-of-sample generalizability.

Contextual word embeddings like BERT or RoBERTa could potentially be a useful extension on our method. However, they are very computationally intensive for large scale applications because generating a contextual word embedding for a single word requires reading the context and making a forward and backward pass through the neural network; meanwhile, non-contextual word

embeddings, while perhaps giving up some accuracy of fit, can be accessed nearly instantaneously. In the next section we describe an application where we compare occupation patent descriptions with millions of patent texts using non-contextual Pennington et al. (2014) word embeddings. Parallelized on 16 cores, we can generate document vectors for about 10 million patents texts and 700 occupation task descriptions and create the matrix of patent-by-occupation pairwise similarity scores in a matter of minutes. Using a back-of-the-envelope calculation based on sample BERT contextual embeddings lookup times, we estimate that doing the same computation with BERT would take *several months*.

Considering the strengths and weaknesses of different approaches, we believe our methodology (especially when using pre-trained embeddings) is likely to be most useful when computational speed is required, when generalizability is to be preferred over precision of fit, and when researchers want a method that tends to work well immediately in a variety of settings without any model fine-tuning. This is especially likely to be the case in the example of cross-domain document comparisons, where similarity scores are being estimated on documents having very different styles, formats, and terminology. Our method will also be more applicable when a researcher has a smaller set of documents on which to train, or when the amount of training data available for one set of documents is drastically larger than the other. Our use of pre-trained embeddings has the added benefit of diminishing barriers to entry for implementing our approach, as fine tuning one's own embeddings can be a challenging task for any researcher without expertise in textual analysis.

Finally, while at times it may be useful for researchers to estimate their own word embeddings, we view the use of pre-estimated word as a feature rather than a bug in our canonical setting of cross-domain document comparisons. These have been pre-trained by professional NLP researchers on a massive set of documents capturing common English usage, and they act as a "neutral third party" when comparing disparate types of documents. Moreover, there are pre-trained embeddings readily available in other languages;[4] they can be used easily out-of-the-box by researchers with little to no experience in textual analysis; the computational burden is comparatively small; and, as we show in the sections that follow, the method produces document similarity metrics that are meaningful and can be employed for economically useful applications.

Our first two examples to follow illustrate that for a given document the measure consistently finds highly similar documents that are clearly related to one another. While false positives can of course occur for any notion of textual similarity, we have examined hundreds of example highly similar matches to a fixed document, and have found this accurate sorting property to be a highly robust feature of the method. Because of this, our approach can be especially useful for grouping

[4]For example, they are freely available for 157 languages here: https://fasttext.cc/docs/en/crawl-vectors.html

documents as being related or not related to a particular concept summarized within a central document. A researcher could employ our procedure to sort by relatedness to a corpus of text capturing the desired idea, select a similarity threshold under which matches no longer appear viable, and potentially remove a few false positives by inspection along the way. For example, our method could be utilized in this manner to retrieve newspaper articles related to a particular war or economic distress.

3 Example Applications

We now present a few applications of our method. Since the method is entirely unsupervised and there is no objectively correct notion of document-by-document textual relatedness, we emphasize that we can't provide formal performance evaluation metrics. Instead we illustrate how our approach sorts documents by similarity in a manner that is intuitive and reasonable, and we discuss ways in which it has been put to use in economic applications already. Our first two examples focus on the canonical cross-domain comparison, where we measure textual similarity of patent documents with either occupation tasks or industry descriptions.

Our final application features the nearest approximation to a proper performance evaluation that we can provide, and shows that method can predict worker flows between occupations based on task similarity at least as well as a TF-IDF weighted bag-of-words benchmark, and substantially better than an off-the-shelf LDA approach. We also show that results are similar for this exercise when using pre-trained word embeddings from Mikolov et al. (2018) instead of Pennington et al. (2014).

3.1 Application: Linking Occupations with Patents

Kogan et al. (2022) apply the methodology described in the previous section to constructing an index of technology exposure of specific occupations over time. A key part of their calculation is the matrix of pairwise similarities (10) between a given patent document i and a description of the tasks performed by occupation j. Here we use occupation task descriptions from the 1991 Dictionary of Occupational titles and the full text of US patents downloaded from the USPTO website (for patents issued post-1976) and scraped from the Google Patents database. We follow Kogan et al. (2022) by retaining only nouns and verbs in the patent and occupation descriptions and lemmatizing the remaining terms (meaning we convert nouns to their singular form and verbs to their present tense).

To illustrate the effectiveness of this methodology in identifying links between technology and occupation task descriptions, we next reproduce some of the representative examples in the paper

in Figure 1. We consider three patents in the list of breakthrough patents identified by Kelly et al. (2020). Patent 276,146, titled "Knitting Machine", was issued in the height of the Second Industrial Revolution in 1883. The top part of the figure documents the five occupations whose task description is most related to this patent. The most related occupation is "Textile Knitting and Weaving Machine Setters, Operators, and Tenders"; the next most similar occupation is "Sewing Machine Hand Operators", followed by "Sewers, hand". The middle part of the figure considers the patent for "Metal wheel for vehicles (1,405,358), which was issued in 1922. The occupation most closely related to this patent is "Automotive Service Technicians and Mechanics", with other production and metal machine workers following. Finally, we examine a patent from a very different era and representing a very different technology. The bottom part of the figure examines the patent "System for managing financial accounts by a priority allocation of funds among accounts," which is U.S. patent number 5,911,135 and was issued in 1999 during the tech boom period. The top occupations related to this patent are Financial managers, credit analysts, and loan interviewers and clerks.

We next perform the reverse exercise, where we fix a particular occupation, and list the most relevant innovations. The occupations we choose are cashiers, loan interviewers and clerks, railroad conductors, and petroleum engineers. Table 1 lists the top five patents that are linked to each of these occupations. Examining the patent titles, we see that each one of these patents is directly related to the work performed by the given occupation. For example, one of the top patents for cashiers is "Vending type machine dispensing a redeemable credit voucher upon payment interrupt" (patent 5,055,657); the top patent for loan interviewers and clerks is titled "Automatic business and financial transaction processing system" (patent number 6,289,319). For rail road conductors, titled "Automatic train control system and method" (patent 5,828,979) is the top patent; and finally, for petroleum engineers the top patent is "Method and equipment for obtaining energy from oil wells" (patent 5117908). The remaining patents in the table are similarly all clearly related to the listed occupations. In general the patents showing up on this list represent technologies that (1) relate to the work performed by individuals in that the occupation; and (2) if adopted, appear likely to be able to change the way that an occupation performs its core work functions and/or substitute for work done by that occupation.

In Kogan et al. (2022), we explore the economic implications of our method for linking patents with occupation task descriptions by annually aggregating patent-occupation similarity scores to create a time-varying index of occupational exposure to technological change. Consistent with capturing technological displacement of labor, our measure of exposure strongly predicts occupational employment and wage declines—the predictability for employment declines even extends over a hundred year period. Examining person-level income data from confidential administrative records

in the post-1980 period, we show that the measure predicts average wage declines for individuals, as well as heightened probabilities of left tail income changes. The measure also gives a picture of time-series trends in exposure by Acemoglu and Autor (2011) occupation task types; we find that more routine and manual physical tasks tend to be most exposed in the 19th and 20th centuries, but there was a dramatic shift towards cognitive type tasks in the second half of the 20th-century. Similarly, while less-educated occupations have always been more exposed to technological change on average, there was a trend towards increased exposure for relatively more educated occupations over the same time period. See Kogan et al. (2022) for a detailed exposition of all these findings.

3.2 Application: Matching Industries with Related Patents

We now apply our method to match industries to their most similar patents. To do this we first scrape descriptions of the 2012 version of 4-digit NAICS industries from the NAICS manual website[5]. We then follow a similar procedure as with the previous application, replacing occupation texts with NAICS industry descriptions. To maximize textual information we append all the descriptions for all 5-digit and 6-digit NAICS codes within a 4-digit NAICS code into one document for each industry, including industry and sub-industry titles. We then follow the same procedure as in the previous section, except we replace occupation tasks with industry descriptions.

To illustrate our method's performance in this application we select a diverse group of industries and list the most similar patents: oil and gas extraction (NAICS code 2111); aerospace product and parts manufacturing (NAICS code 3364); software publishers (5112); and, restaurants and other eating places (NAICS code 7225). We restrict to US patents that are issued in the year 2000 and identified as important breakthrough technologies by Kelly et al. (2020). The top five patent matches for each of these industries are listed in 2. As before, the matches are intuitive and clearly related: all the top patents for oil and gas extraction involve extracting and disposing of fossil fuels; the top patents for software publishers are related to creating and distributing software, and so on for the other industries.

This application suggests that our approach could be leveraged to predict the likely industry of use of patents. To this end, Chen and Srinavasan (2022) have applied our methodology to predict the exposure of firms to AI activity based on textual overlap between firms' industry descriptions and AI patent abstracts.

[5]Found here: https://www.census.gov/naics/?58967?yearbck=2012

3.3 Application: Predicting Cross-Occupation Employment Flows With Textual Similarity

In the previous section we showed that the method of representing documents as TF-IDF weighted averages of word embeddings does well in textual comparisons where documents come from very different domains (patent texts versus occupation task descriptions). We view this as a primary advantage of our method, because it is intended to allow for textual overlap between documents that may use very different vocabularies. However, our method also can be used fruitfully when comparing documents from the same domain. Here we examine whether occupation-by-occupation textual similarity predicts the probability that a given worker switches between two given occupations. For this task we use data from Schubert, Stansbury, and Taska (2021), who construct occupation transition shares using data from Burning Glass technologies.[6] Along with this application we include a publicly available GitHub repository with all code and data to implement our method on these documents and replicate these results.[7]

We let $\text{share}_{i \to j}$ denote the fraction of workers leaving occupation i who transition to occupation j. Using the method described in section 1, we compute $\rho_{i,j}$, the raw similarity score between occupations i and j. Because of the nature of using a weighted average of word embeddings, all of the similarity scores are non-zero. We experiment with a similar transformation to Kogan et al. (2022) in order to impose sparsity. Specifically, we transform all similarity scores so that all beneath the 80th percentile get a weight of zero, and scale linearly above this threshold so that a score at the maximum gets a weight of 1 and a score right at the 80th percentile gets a weight of 0. We call this transformed similarity score $\tilde{\rho}_{i,j}$. Finally, we also use the similarity score percentile instead of the raw score, which we call $\rho_{i,j}^{pctile}$. We denote similarity scores obtained with bag of words methods with "BOW" superscript (i.e. $\rho_{i,j}^{BOW}$ for the raw similarities).

We then run regressions of the form

$$\log(\text{share}_{i \to j}) = \alpha + \beta \rho_{i,j} + \alpha_i + \alpha_j + \epsilon_{i \to j} \tag{11}$$

Depending on the specification we include α_i and α_j, occupation i and j fixed effects. We weight observations by the total number of transitions observed between occupation i and j. Since it's likely that workers find it easier to transition to another occupation with similar skills, we would expect that $\beta > 0$ and should be highly statistically significant. In Panel A of Table 3 we confirm this result for all specifications. The t-statistics for estimates of β range from about 15 to about 23, and the within R^2 ranges from 14% to 21%.

[6] Thanks to Schubert et al. (2021) for making their data available.

[7] The repository can be found here: `https://github.com/bryanseegmiller/DocSimilarityWordEmbeddings`.

In panel B of Table 3 we compare results from the commonly used bag of words method, which requires exact overlap in terms. Despite the fact that our method is primarily intended for use where documents come from different domains, we find that our method outperforms bag of words at predicting occupation flows in terms of R^2 and statistical precision.

In this example and in the previous examples we have used Pennington et al. (2014) word embeddings.[8] However, other pre-trained embeddings also perform similarly. To see this, we re-do the same exercise using a different set of pre-trained embeddings, this time from Mikolov et al. (2018)–commonly called "FastText."[9] We report the results in Panel A of Table 4. The R^2 for occupation-occupation transitions goes down slightly, but by and large the performance is quite comparable.

To further illustrate the predictive power of our method in this application, we also try dimensionality reduction of the occupation texts via an LDA clustering algorithm. This requires pre-specifying the number of topics k; each document will then be represented by a k-dimensional vector of topic importance weights. To give LDA the best representation, we experiment with a number of topic values, and we find that 100 topics gives the highest predictability. We compute an LDA similarity score by using the cosine similarities of the document vectors of topic importance weights, which we call $\rho_{i,j}^{LDA}$. In Panel B of Table 4 we report results using just raw similarity scores and with occupation fixed effects for brevity. We re-report the regressions including our baseline measure $\rho_{i,j}$, the bag of words similarity $\rho_{i,j}^{BOW}$; then we add a specification with the LDA similarity $\rho_{i,j}^{LDA}$. In the last two columns we include all three measures simultaneously, where in the final column we again replace our baseline GloVe embeddings with FastText embeddings. Panel B of Table 4 shows that LDA performs quite poorly in terms of within R^2, and it is subsumed by the other measures–actually turning negative–when included in a head-to-head. Meanwhile, the strongest predictors are our measure (whether using GloVe or FastText), but the bag-of-words method does still retain strong predictability, suggesting it does contain some additional information.

The poor performance of LDA underscores the difficulty of training models on relatively small amounts of information—in this case, about 700 occupation task descriptions. The LDA routine clusters words into groups that tend to show up in documents together, but clearly the clustering struggles with noisy estimation in this setting. While the estimation approach is different, word embeddings are also trained based on context, and so training our own embeddings here would likely also result in a noisy fit that doesn't generalize well out-of-sample. The problem would likely be just as severe or worse for a doc2vec model, which requires estimating a more complicated neural

[8] Available here: https://nlp.stanford.edu/projects/glove/. We select the 300-dimensional embeddings estimated on 840 billion tokens from the common crawl dataset.

[9] These can be found here: https://fasttext.cc/docs/en/english-vectors.html. We select the vectors estimated on common crawl with subword information.

network than for training word embeddings. This underscores that our method is highly flexible and generalizes easily in different settings, even without much additional tweaking.

4 Some Notes on Implementation

In this section we provide some guidance to users on implementing our method. We first discuss programming implementation. With open source natural language processing libraries, computing measures of textual similarity has become much more accessible for researchers with a little programming knowledge. As mentioned in the previous section, we include a GitHub repository with code necessary to replicate these results. We rely on the Python programming language, which is fully free and open source, and we use the common Anaconda distribution of Python in our examples. For cleaning and processing texts we rely primarily on the Python package NLTK; for representing textual documents numerically and extracting word embeddings we rely on the Python package Gensim.

Users may consider adjusting similarity scores in sensible ways depending on their setting. For example, in Kogan et al. (2022) we note that the patent OCR quality gets worse over time, which leads to lower similarity scores on average for earlier patents. Accordingly, we remove year fixed effects from the similarity scores; we also previously experimented with removing patent tech class by year fixed effects, and noted that all results were essentially unchanged. In many contexts one might desire similarity scores to be sparse, where only sufficiently high scores are assigned a positive weight. Word embeddings methods result in dense word vectors, and so when a document is represented as a weighted average of dense vectors the similarity scores will tend to nearly always be positive. The ordering of the similarity scores is where the primary source of information lies. In both Kogan et al. (2022) and Autor, Salomons, and Seegmiller (2021) the authors impose a right tail cutoff in order for scores to be given positive weight.

5 Conclusion and Potential for Further Applications

In this paper we illustrate how representing documents as weighted averages of word embeddings can create useful geometric representations of document meanings. The method is particularly well-suited to cross-domain comparisons between documents with disparate formats and terminology. We find that the method performs remarkably well in matching occupations or industries to their most similar patents, or patents to their most similar occupations. The method is particularly well-suited to cross-domain document comparisons In Kogan et al. (2022) an aggregated measure of patent-occupation task textual overlap predicts occupational employment and wage declines over a

century-long period, illustrating how the method can be useful for analyzing historical documents and predicting economic outcomes. Users may easily adapt the method to particular settings, and there are many potential additional applications. For example, the method could be useful for linking historical patents with their most likely industries of origin or use, comparing text in newspapers with other literature, or analyzing product descriptions, to name just a few possibilities. Moreover, because our measure does well at sorting the most related documents to a given document, it may be of special interest to economic historians who are interested in retrieving texts that are related to a central document considered to well capture or summarize an economic idea, concept, or object of interest.

References

Acemoglu, D. and D. Autor (2011). Skills, tasks and technologies: Implications for employment and earnings. In *Handbook of Labor Economics, Volume 4. Amsterdam: Elsevier-North*, pp. 1043–1171.

Acemoglu, D., D. Y. Yang, and J. Zhou (2021). Political pressure and the direction of research: Evidence from china's academia.

Ash, E. and S. E. Hansen (2022). Text algorithms in economics.

Atalay, E., P. Phongthiengtham, S. Sotelo, and D. Tannenbaum (2020). The evolution of work in the united states. *American Economic Journal: Applied Economics 12*(2), 1–34.

Autor, D., A. Salomons, and B. Seegmiller (2021). New frontiers: The origins and content of new work, 1940–2018. Working paper, MIT.

Biasi, B. and S. Ma (2020). The education-innovation gap. working paper, Yale.

Blei, D. M., A. Y. Ng, and M. I. Jordan (2003). Latent dirichlet allocation. *J. Mach. Learn. Res. 3*, 993–1022.

Chen, W. and S. Srinavasan (2022). Going digital: Implications for firm value and performance.

Dai, A. M., C. Olah, and Q. V. Le (2015). Document embedding with paragraph vectors.

Deerwester, S., S. Dumais, G. Furnas, T. Landauer, and R. Harshman (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, 391–407.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2018). Bert: Pre-training of deep bidirectional transformers for language understanding.

Gentzkow, M., B. Kelly, and M. Taddy (2019). Text as data. *Journal of Economic Literature 57*(3), 535–74.

Hansen, S., T. Ramdas, R. Sadun, and J. Fuller (2021). The demand for executive skills.

Hoberg, G. and G. Phillips (2016). Text-based network industries and endogenous product differentiation. *Journal of Political Economy 124*(5), 1423 – 1465.

Kelly, B., D. Papanikolaou, A. Seru, and M. Taddy (2020). Measuring technological innovation over the long run. *American Economic Review: Insights forthcoming.*

Kogan, L., D. Papanikolaou, L. D. W. Schmidt, and B. Seegmiller (2022). Technology-skill complementarity and labor displacement: Evidence from linking patents with occupations. Working paper.

Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov (2019). Roberta: A robustly optimized bert pretraining approach.

Mikolov, T., E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin (2018). Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Mikolov, T., I. Sutskever, K. Chen, G. Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. *CoRR abs/1310.4546*.

Pennington, J., R. Socher, and C. D. Manning (2014). Glove: Global vectors for word representation. In *EMNLP*.

Schubert, G., A. Stansbury, and B. Taska (2021). Employer concentration and outside options. Working paper.

Webb, M. (2020). The impact of artificial intelligence on the labor market.

Figures and Tables

Figure 1: Examples of Technology Exposure: By Innovations

Table 1: Most Similar Patents For Select Occupations

$\rho_{i,j}$	**Patent No.**	**Patent Title**
Panel A: Cashiers (SOC Code 412011)		
.968	5055657	Vending type machine dispensing a redeemable credit voucher upon payment interrupt
.967	5987439	Automated banking system for making change on a card or user account
.967	5897625	Automated document cashing system
.965	6012048	Automated banking system for dispensing money orders, wire transfer and bill payment
.958	5598332	Cash register capable of temporary-closing operation
Panel B: Loan Interviewers and Clerks (SOC Code 434131)		
.981	6289319	Automatic business and financial transaction processing system
.977	5611052	Lender direct credit evaluation and loan processing system
.979	6233566	System, method and computer program product for online financial products trading
.977	5940811	Closed loop financial transaction method and apparatus
.976	5966700	Management system for risk sharing of mortgage pools
Panel C: Railroad Conductors (SOC Code 534031)		
.964	5828979	Automatic train control system and method
.959	6250590	Mobile train steering
.954	3944986	Vehicle movement control system for railroad terminals
.958	6135396	System and method for automatic train operation
.958	5797330	Mass transit system
Panel D: Petroleum Engineers (SOC Code 172171)		
.944	5117908	Method and equipment for obtaining energy from oil wells
.941	4265309	Evaluation and production of attic oil
.936	4031956	Method of recovering energy from subsurface petroleum reservoirs
.935	5165235	System for using geopressured-geothermal reservoirs
.937	4458945	Oil recovery mining method and apparatus

Note: This table lists the top 5 most similar patents generated by our method for select occupations. $\rho_{i,j}$ refers to the raw similarity score between the patent and occupation. The sample covers patents issued between 1976 and 2002. See main text for details.

Table 2: Most Similar Breakthrough Patents Issued in 2000 to Select 4-Digit NAICS Industries

$\rho_{i,j}$	**Patent No.**	**Patent Title**
Panel A: Oil and Gas Extraction (NAICS Code 2111)		
.922	6030536	Disposal method for fuel oil and crude oil spills
.912	6037515	Process for producing ethylene from a hydrocarbon feedstock
.897	6074769	Method of generating electric energy from regenerative biomass
.895	6111154	High energy density storage of methane in light hydrocarbon solutions
.893	6033447	Start-up process for a gasification reactor
Panel B: Aerospace Product and Parts Manufacturing (NAICS Code 3364)		
.927	6065720	Manufacture of aircraft
.923	6056237	Sonotube compatible unmanned aerial vehicle and system
.911	6092763	Aircraft crash damage limitation system
.910	6044700	Aircraft equipment configuration identification interface guide
.899	6138945	Neural network controller for a pulsed rocket motor tactical missile system
Panel C: Software Publishers (NAICS Code 5112)		
.921	6044469	Software publisher or distributor configurable software security mechanism
.919	6167568	Method and apparatus for implementing electronic software distribution
.918	6073214	Method and system for identifying and obtaining computer software from a remote computer
.915	6151643	Automatic updating of diverse software products on multiple client computer systems by downloading scanning application to client computer and generating software list on client computer
.914	6029145	Software license verification process and apparatus
Panel D: Restaurants and Other Eating Places (NAICS Code 7225)		
.876	6024996	Packaged carbonated coffee beverage
.869	6165522	Processed food and a method for making a processed food product for mass distribution
.864	6096361	Method for non-frozen preservation of food at temperature below freezing point
.853	6098529	Display case for food items
.851	6062126	Beverage quality control apparatus and method

Note: This table lists the top 5 most similar patents generated by our method for select NAICS industries. $\rho_{i,j}$ refers to the raw similarity score between the patent and industry. We restrict to breakthrough technologies according to Kelly et al. (2020) that are issued in the year 2000. See main text for details.

Table 3: Predicting Probability of Occupation Transitions Using Textual Similarity

Panel A: Tf-Idf Weighted Average of Word Embeddings

	(1)	(2)	(3)	(4)	(5)	(6)
Raw Similarity, $\rho_{i,j}$	13.7***	13.1***				
	(15.46)	(19.71)				
Adjusted Similarity, $\tilde{\rho}_{i,j}$			4.81***	2.97***		
			(18.30)	(22.69)		
Similarity Pctile, $\rho_{i,j}^{pctile}$					0.034***	0.031***
					(15.70)	(18.42)
Occ FE		X		X		X
N	258118	258115	258118	258115	258118	258115
R^2 (Within)	0.17	0.21	0.14	0.14	0.16	0.18

Panel B: Bag of Words with Tf-Idf Weights

	(1)	(2)	(3)	(4)	(5)	(6)
Raw Similarity, $\rho_{i,j}^{BOW}$	13.5***	7.78***				
	(14.50)	(18.85)				
Adjusted Similarity, $\tilde{\rho}_{i,j}^{BOW}$			12.2***	6.97***		
			(12.15)	(17.76)		
Similarity Pctile, $\rho_{i,j}^{pctile,BOW}$					0.034***	0.020***
					(18.19)	(22.73)
Occ FE		X		X		X
N	258118	258115	258118	258115	258118	258115
R^2 (Within)	0.13	0.15	0.093	0.12	0.15	0.13

Note: This table shows estimates of regression equation (11) in the main text. The dependent variable is the log of the share of transitions out of occupation i that go to occupation j, and the main dependent variable are the measures of textual overlap between occupation i and j. In panel A shows results using our method of representing documents as Tf-Idf weighted averages of word embeddings; panel B instead represents documents using a bag of words approach, where words are also Tf-Idf weighted. Observations are weighted by the number of transitions from occupation i to occupation j and standard errors are double clustered by occupations i and j, and the "Occ FE" denote fixed effects for both occupation i and j are included. See section 3.3 for further details.

Table 4: Predicting Occupation Transitions: Alternative Word Embeddings and Comparison With Bag of Words and LDA Methods

Panel A: FastText Embeddings Instead of GloVe

	(1)	(2)	(3)	(4)	(5)	(6)
Raw Similarity, $\rho_{i,j}^{FastText}$	19.3***	17.5***				
	(13.97)	(18.35)				
Adjusted Similarity, $\tilde{\rho}_{i,j}^{FastText}$			5.13***	3.16***		
			(17.00)	(19.93)		
Similarity Pctile, $\rho_{i,j}^{pctile,FastText}$					0.033***	0.028***
					(15.03)	(17.24)
Occ FE		X		X		X
N	258118	258115	258118	258115	258118	258115
R^2 (Within)	0.16	0.19	0.12	0.13	0.16	0.17

Panel B: Comparison with Bag of Words and LDA

	(1)	(2)	(3)	(4)	(5)
Raw Similarity, $\rho_{i,j}$	13.1***			10.0***	
	(19.71)			(14.85)	
Raw Similarity, $\rho_{i,j}^{FastText}$					12.9***
					(13.20)
Raw Similarity, $\rho_{i,j}^{BOW}$		7.78***		3.48***	3.78***
		(18.85)		(10.26)	(9.89)
Raw Similarity, $\rho_{i,j}^{LDA}$			1.14***	-0.086*	-0.096*
			(13.35)	(-1.68)	(-1.89)
Occ FE	X	X	X	X	X
N	258115	258115	258115	258115	258115
R^2 (Within)	0.21	0.15	0.019	0.22	0.21

Note: This table shows estimates of regression equation (11) in the main text. The dependent variable is the log of the share of transitions out of occupation i that go to occupation j, and the main dependent variable are the measures of textual overlap between occupation i and j. In Panel A we replace the Pennington et al. (2014) ("GloVe") word embeddings used in Table 3 with Mikolov et al. (2018) ("FastText") embeddings. In the first two columns of Panel B we again report results using raw similarity scores with baseline GloVe embeddings and bag-of-words; in the third column we instead measure similarity by using Latent Dirichlet Allocation (Blei et al., 2003) to cluster words into topics, and then use the topics to measure document similarity; in the fourth of Panel B we include all three measures simultaneously; finally, in the fifth column we include all three measures but use FastText embeddings instead. Observations are weighted by the number of transitions from occupation i to occupation j and standard errors are double clustered by occupations i and j, and the "Occ FE" denote fixed effects for both occupation i and j are included. See section 3.3 for further details.