

WMDecompose: A Framework for Leveraging the Interpretable Properties of Word Mover’s Distance in Sociocultural Analysis

Mikael Brunila

McGill University

Department of Geography

mikael.brunila@gmail.com

Jack LaViolette

Columbia University

Department of Sociology

jack.laviolette@columbia.edu

Abstract

Despite the increasing popularity of NLP in the humanities and social sciences, advances in model performance and complexity have been accompanied by concerns about interpretability and explanatory power for sociocultural analysis. One popular model that balances complexity and legibility is Word Mover’s Distance (WMD). Ostensibly adapted for its interpretability, WMD has nonetheless been used and further developed in ways which frequently discard its most interpretable aspect: namely, the word-level distances required for translating a set of words into another set of words. To address this apparent gap, we introduce WMDecompose: a model and Python library that 1) decomposes document-level distances into their constituent word-level distances, and 2) subsequently clusters words to induce thematic elements, such that useful lexical information is retained and summarized for analysis. To illustrate its potential in a social scientific context, we apply it to a longitudinal social media corpus to explore the interrelationship between conspiracy theories and conservative American discourses. Finally, because of the full WMD model’s high time-complexity, we additionally suggest a method of sampling document pairs from large datasets in a reproducible way, with tight bounds that prevent extrapolation of unreliable results due to poor sampling practices.

1 Introduction: The Paradox of Word Mover’s Distance

The present paper introduces WMDecompose, an iteration of the Word Mover’s Distance (WMD) (Kusner et al., 2015) model commonly used for determining the semantic distances between pairs of documents. Leveraging word vectors from models such as word2vec (Mikolov et al., 2013), fastText (Bojanowski et al., 2016) and GloVe (Pennington et al., 2014), WMD was presented as a method that was not only hyper-parameter free and thus

easy to use, but also highly interpretable (Kusner et al., 2015). Arguably, this still makes the model a viable alternative for document similarity tasks, despite the recent and rapid rise of contextual and rich embeddings from Transformer-type models (Vaswani et al., 2017) such as BERT (Devlin et al., 2019). However, WMD is computationally expensive, with the full model running at cubic time complexity (Kusner et al., 2015). This may explain why, despite its initial pitch as an interpretable alternative, WMD has mainly been further developed and applied in ways that focus on decreasing the model’s high runtime, while ignoring or undermining the inherent interpretability of the model (Atasu et al., 2017; Werner and Laber, 2020).

To our knowledge, no current WMD implementation provides an out-of-the-box means of retaining word-level information, despite its utility to many research agendas. To confront this paradoxical situation, we introduce a set of methods and Python code for retaining the individual word distances that make WMD interpretable while simultaneously suggesting a simple trick for efficiently estimating the distance between large sets of documents. Specifically, we propose using implementation of the “relaxed” WMD (Kusner et al., 2015) with linear time complexity (Atasu et al., 2017) to first estimate the distances between a full set of documents, then using optimal pairing with the Gale-Shapley algorithm (Gale and Shapley, 1962) on this estimate to find the pairs between two sets of documents that minimize the average pairwise distance. Next, full WMD is calculated between each pair, while progressively adding the contributions of individual words to the overall distance between document sets. Words are, in turn, grouped using K-means clustering and vector dimensionality reduction to decompose the distances not only by word, but by thematic cluster. Each cluster is defined by its constituent words, and hence highly interpretable.

Having introduced WMDecompose, we demonstrate its utility in a social scientific context by exploring interrelated trends over time between two social media corpora: `r/conspiracy` and `r/The_Donald`, the primary Reddit communities for self-identified conspiracy theorists and Donald Trump supporters, respectively. While this presents only a cursory and exploratory engagement with these thematically complex data, we hope that it demonstrates the analytic potential of WMDecompose for social science and humanities research.

Finally, we also provide a complementary analysis using a well-known Yelp review dataset, to act as a sanity check on the validity of our method. This analysis can be found in Appendix B.

2 Previous Work

2.1 Word Mover’s Distance

Originally proposed by [Kusner et al. \(2015\)](#), WMD has become a popular metric of document semantic distance in computational linguistics and related subfields. An innovation and special instance of Earth Mover’s Distance ([Rubner et al., 1998](#); [Pele and Werman, 2008, 2009](#)), WMD leverages word embeddings to compute the minimum distance required to “move” the words from one document to another, providing a measure of document-level semantic (dis)similarity as a sum of the distance required to move individual words from one document to another (see Figure 1). Since its introduction, many related algorithms and analytic approaches have been proposed for language engineering tasks (e.g. [Atasu et al., 2017](#); [Huang et al., 2016](#); [Ren and Liu, 2018](#)). More recently, WMD and its many variations have also been applied to sociocultural analyses of data ranging from survey response data ([Taylor and Stoltz, 2020](#)), to Ancient Greek literature ([Pöckelmann et al., 2020](#)), to dyadic conversational dynamics ([Nasir et al., 2019](#)).

WMD is typically parameterized in the following manner ([Kusner et al., 2015](#); [Huang et al., 2016](#)), using words represented as embeddings produced with algorithms such as word2vec ([Mikolov et al., 2013](#)). Let $x_i \in \mathbb{R}^d$ be the i th embedding in d -dimensional space, drawn from a word embedding matrix $X \in \mathbb{R}^{d \times n}$ representing a vocabulary of n words. Let \mathbf{d}^a and \mathbf{d}^b be two n -dimensional, normalized bag-of-words (nBOW) vectors for a pair of documents where d_i^a is the normalized num-

ber of times word i occurs in vector d^a . WMD then attempts to find a transportation matrix $\mathbf{T} \in \mathbb{R}^{n \times n}$ that minimizes the total distance required to move all words in the first document to the second document, where $\mathbf{T}_{i,j}$ describes how much of the normalized word vector d_i^a should be transported to the normalized word vector d_j^b . Formally, WMD returns the minimum distance to move from document \mathbf{d}^a to document \mathbf{d}^b , given by summing the product of the optimal “flow” $\mathbf{T}_{i,j}$ from all words in the two documents with the “cost” $c(i,j)$ of moving between each word vector in the two documents:

$$\text{WMD}(\mathbf{d}^a, \mathbf{d}^b) = \min_{\mathbf{T} \geq 0} \sum_{i,j=1}^n \mathbf{T}_{i,j} c(i,j) \quad (1)$$

Furthermore, the equation is subject to the constraint that the entirety of the “mass” of \mathbf{d}^a should be distributed in the flows to \mathbf{d}^b and vice versa:

$$\begin{aligned} \sum_{j=1}^n \mathbf{T}_{i,j} &= d_i^a \quad \forall i \in \{1, \dots, n\} \\ \sum_{i=1}^n \mathbf{T}_{i,j} &= d_j^b \quad \forall j \in \{1, \dots, n\} \end{aligned} \quad (2)$$

Even though the original implementation of the WMD uses Euclidean distance for the metric $c(i,j)$, the similarity between word embeddings in general and with WMD in particular ([Yokoi et al., 2020](#)) is better captured using the cosine distance¹:

$$c(i,j) = 1 - \frac{\mathbf{x}_i \cdot \mathbf{x}_j}{\|\mathbf{x}_i\| \|\mathbf{x}_j\|} \quad (3)$$

Furthermore, \mathbf{d}^a and \mathbf{d}^b can be normalized using other techniques, such as Term frequency-Inverse document frequency (Tf-Idf), combined with L1-normalization, which allocate more mass to words that are more common in a specific document than the overall vocabulary.

While there is a well-established literature on solving this linear program using the EMD algorithm, doing so is computationally prohibitive. [Kusner et al. \(2015\)](#) note that “the best average time complexity of solving the WMD optimization

¹The use of the word “distance” here can be slightly confusing, as it is not quite the same as the “distance” in Word Mover’s Distance. For the latter, the distance between two documents is composed of the cosine distance *and* the “mass” of the nBOW representation of the documents to be moved to one another.

problem scales” $O(V^3 \log V)$, with V denoting the number of unique words in the corpus. Consequently, calculating the distances for large datasets will often prove insurmountable for WMD. [Kusner et al. \(2015\)](#) suggest working around this issue by calculating an approximation of WMD, where the distance from the two pairs are calculated with relaxed constraints, so that $\mathbf{T}_{i,j}$ must contain the mass or flow from the source document, but not the target. This “relaxed” WMD or R-WMD is then instead parameterized with only one constraint:

$$\begin{aligned} \text{R-WMD}(\mathbf{d}^a, \mathbf{d}^b) &= \min_{\mathbf{T} \geq 0} \sum_{i,j=1}^n \mathbf{T}_{i,j} c(i, j) \\ \text{subject to: } \sum_{j=1}^n \mathbf{T}_{i,j} &= d_i \quad \forall i \in \{1, \dots, n\} \end{aligned} \quad (4)$$

Repeating the process, so that R-WMD is instead calculated from \mathbf{d}^b to \mathbf{d}^a , gives us an estimate of the bounds within which the full WMD must be located. While R-WMD gives only an estimate of WMD within certain bounds, it has been shown to be a good approximation of the full WMD ([Kusner et al., 2015](#)). Notably, instead of the cubic time complexity of WMD, R-WMD can be performed with quadratic time complexity $O(V^2)$. Furthermore, [Atasu et al. \(2017\)](#) have demonstrated how to calculate the R-WMD so that the time complexity is reduced from quadratic to linear with the Linear Complexity Relaxed WMD (LC-RWMD) algorithm. However, this solution does *not* retain the distances contributed by individual words to the R-WMD.

2.2 WMD and interpretability

While these and other suggested tricks and improvements (e.g. [Tithi and Petrini, 2021](#); [Werner and Laber, 2020](#); [Yokoi et al., 2020](#)) for efficient WMD make the algorithm a feasible and powerful tool for comparing the semantic distance between large sets of documents, they have been introduced with little regard to the algorithm’s initial claims to intuitive and interpretable explanations for document (dis)similarities. Consider, for example, Figure 1, introduced by [Kusner et al. \(2015\)](#) as an example of how the distances between three documents could be decomposed into different parts. Indeed, for sophisticated NLP techniques such as WMD to be maximally useful for sociocultural analysis, the possibility to decompose document-level results

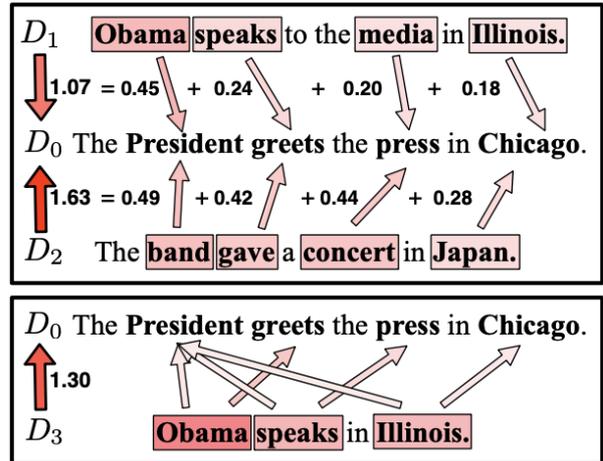


Figure 1: An illustration of WMD in action by [Kusner et al. \(2015\)](#). (Top:) The components of the WMD metric between a query D_0 and two sentences D_1 , D_2 (with equal BOW distance). The arrows represent flow between two words and are labeled with their distance contribution. (Bottom:) The flow between two sentences D_3 and D_0 with different numbers of words. This mismatch causes the WMD to move words to multiple similar words.

into interpretable lexical information is key. For example, if the mean WMD document distances between two longitudinal corpora sampled at some time t_0 and again at some later time t_1 shrink, it might indicate that the corpora have become more similar. However, the change in distance by itself would tell the curious analyst little about the particular lexical phenomena responsible for the semantic changes, and (crucially) how these might relate to extralinguistic social and symbolic processes. In order to tell the full story, fluctuations in distance must be decomposed into interpretable parts.

To invoke [Danilevsky et al.’s \(2020\)](#) classification scheme of explanation in NLP, WMD provides explanations that are *local* (i.e. the distance between any pair of documents can be decomposed to individual words) and *self-explanatory* (i.e., no *post hoc* processing of the model outputs is necessary). However, these explanations have, to the best of our knowledge, not been leveraged in applied research with WMD, most likely due to the prohibitive computational cost of calculating the full WMD and the neglect of interpretability in more efficient elaborations of the model. Moreover, applied WMD research has to date generally begun with an *a priori* interest in the relationship between documents and particular “concept” words ([Stoltz and Taylor, 2019](#), e.g.) or predefined topics

(Wu and Li, 2017, e.g.). Although such analyses are perfectly valid, a fully inductive relationship to words of interest can be useful when the analyst does not have or desire strong assumptions about what lexical changes underlie the phenomena of interest.

To these ends of inductive interpretability, we propose what we believe to be a novel analytic pipeline which combines LC-RWMD decomposed at the word level, optional document pairing based on the Gale-Shapley matching algorithm to ensure robustness, and t-SNE (Maaten and Hinton, 2008) reduced word vectors with K-means clustering (Lloyd, 1982; Elkan, 2003) to enhance interpretability by inducing higher-level thematic groupings. To assist future researchers, we additionally provide a Python package and example code notebooks available on Github.²

3 Data

3.1 The corpus: conspiracy theories and American conservatism online

Several researchers have identified conspiratorial thinking as a feature of the (post-)Trump era of American conservative politics (Bracewell, 2021; Hellinger, 2018; Polletta and Callahan, 2019). Many such studies have been theoretical and/or qualitative (e.g. Barkun, 2017; Stecula and Pickup, 2021), survey-based (e.g. Federico et al., 2018; Miller et al., 2016; Uscinski et al., 2020), focused on patterns of “misinformation” dissemination on social media (e.g. Benkler et al., 2017; Marwick and Lewis, 2017), or on the discourse of political elites (e.g., Hornsey et al., 2020; Neville-Shepard, 2019). The present paper chooses a different, though not unprecedented, approach of studying large user-generated text corpora from [reddit.com](https://www.reddit.com). Specifically, we analyze “self posts”³ from the `r/The_Donald` and `r/conspiracy` communities (“*subreddits*”); the former was the main subreddit for Donald Trump supporters before being banned in June 2020, while the latter remains the largest subreddit for self-identified conspiracy theorists. Others have examined both

²<https://github.com/maybemkl/wmdecompose>

³Self posts are forum submissions which contain only text as opposed to links to external sites. We use self posts in lieu of comments, as the latter tend to be shorter and less orderly due to their nested structure. Further, self posts tend to be more regulated by forum moderators, suggesting that they are more likely to reflect the norms of the community.

`r/The_Donald` and its role in alt-right politics (Massachs et al., 2020; Ribeiro et al., 2020; Shepherd, 2020), as well as conspiracy theorists on the platform (Klein et al., 2019; Phadke et al., 2021; Samory and Mitra, 2018). Research has indicated a sizable and statistically significant overlap in users between the two subreddits (Massachs et al., 2020; Nithyanand et al., 2017). However, these observations have been largely based on subreddit co-subscriber networks, and have paid comparatively less attention to large-scale linguistic patterns over time, a gap to which we hope to contribute.

Data was collected using the Pushshift Reddit dataset publicly available on Google BigQuery (Baumgartner et al., 2020). Because we are interested in change over time, we delineate two discontinuous periods of interest: t_0 , defined as the twelve months following the creation of `r/The_Donald` (July 11, 2015–July 11, 2016), and t_1 , the final twelve months available in the dataset (August 31, 2018–August 31, 2019). These two year-long snapshots, separated by roughly two years, offer a glimpse of the relationship between conspiratorial language and the discourse of self-identified Donald Trump supporters during the Trump presidency.

3.2 Sampling and preprocessing

To reduce computation and run-time, we sample 5,000 posts from each subreddit at t_0 and t_1 for a total of 20,000 posts. Random sampling was restricted to those posts at least 30 words long (to ensure that documents contain adequate lexical information), and having a positive score (2 or greater, as Reddit posts start with a score of 1) to ensure that the lexical content is generally representative of the community. Once sampled, the text is preprocessed using a standard pipeline for NLP applications. The specifics of this pipeline are described in Appendix A. The processed dataset contains 1,509,553 tokens, 596,596 tokens from `r/The_Donald` and 993,957 from `r/conspiracy`.

4 Methods

4.1 Embedding and clustering

After preprocessing, words were converted to vectors using a fine-tuned word2vec model introduced in (Mikolov et al., 2013), originally pre-trained on Google News Vectors containing about 300 billion words. Fine-tuning details and hyperparameters are included in Appendix A. Due to the large number of unique words, and to increase the interpretability

of the final outputs, words were clustered on the basis of their embedding vectors using t-SNE, an algorithm that is commonly used to reduce the dimensionality of word embeddings while still preserving the original structure of the higher-dimensional form (for uses with WMD, see: Huang et al., 2016; Gülle et al., 2020). In our case, we reduce the dimensionality of the original word vectors from 300 to two and use K-means on the reduced dimensions to generate 100 clusters of words according to their semantic similarity. Clustering words allows us to examine not only the changing usage of individual words across subreddits, but also of these higher-level thematic groupings. The number of clusters was chosen heuristically after inspecting both elbow plots and silhouette scores for clusters in the range of 10 to 200. Our method is robust to using raw embeddings as well as other popular reduction techniques before clustering, such as UMAP (McInnes et al., 2020).

4.2 WMDecompose

We now introduce WMDecompose, the core contribution of this paper, which provides the ability to examine the word-level distances required to move between two sets of documents, such as the self posts in r/conspiracy and r/The_Donald. The comparison of these documents happens, on the one hand, through retaining the word-level distances of moving between pairs of documents from each set and, on the other hand, by clustering words and aggregating their added distances by cluster.

More generally: Given two sets of documents, S^a and S^b , the matrix $\mathbf{T}_{i,j}$ of flows between all the individual documents in both sets, and clusters for the input word vectors, WMDecompose returns the following information twice, once in terms of movement from the first set to the second and once in terms of movement from the second set to the first:

- A The aggregate word-level WMD or WMD_w for each word w in a vocabulary V when moving all documents from one set of documents S^a to another set S^b .
- B The aggregate cluster-level or Cluster Mover’s Distance (CMD_c) for each cluster c when moving all documents from S^a to S^b , with keywords for each c determined by the words with the highest WMD_w within the cluster.

The WMD_w gives us a good sense of the most

important words separating the two sets. The CMD_c allows us to organize this information by cluster, with interpretable cluster keywords that are dynamically ranked, depending on their importance for the particular case at hand.

More formally, WMD_w is calculated in the following manner. Let w^a and w^b be two words contained in S^a and S^b , respectively, and let WMD_{w^a} be the total distance accumulated by word w^a when moving from S^a to S^b . Next, if n is the number of documents in S^a that contain the word w^a , m is the number of words in some document d^b that w^a is distributed over, $\mathbf{t}_{i,j}$ is the vector of flows from w^a in document d_i^a to each word w_j^b in d^b , and $c(w^a, w_j^b)$ is the cost in terms of cosine distance to move from each w^a to each w_j^b , then

$$\text{WMD}_{w^a} = \sum_{i=1}^n \sum_{j=1}^m \mathbf{t}_{i,j} c(w^a, w_j^b) \quad (5)$$

WMD_{w^b} is counted in the same manner, only this time using the flow and cost from words in S^b to words in S^a .

The CMD_{c^a} , i.e. the CMD_c for movement from S^a to S^b , is then calculated simply by summing over the aggregated distances of all $w^a \in c$, i.e. all words w^a belonging to cluster c . If there are p such words in S^a for some cluster c , then

$$\text{CMD}_{c^a} = \sum_{k=1}^p \text{WMD}_{w^a} \quad \forall w^a \in c^a \quad (6)$$

Again, CMD_{c^b} is counted in the same manner. Furthermore, the total WMD_{S^a} when moving all documents from S^a to some pair in S^b can consequently be described as

$$\text{WMD}_{S^a} = \sum \text{WMD}_{w^a} = \sum \text{CMD}_{c^a} \quad (7)$$

In order to further accentuate the differences between the word-level distances WMD_{w^a} and WMD_{w^b} , we subtract the WMD_{w^a} for each word w^a in the vocabulary of S^a from the corresponding WMD_{w^b} (if it exists) in the other set and vice versa. This way results will not be cluttered by words for which the WMD_w is very similar across sets and instead the differences between the two sets will be highlighted. Hence, the final WMD_{w^a} with “difference” or $\text{WMD}_{w^a}^d$ for word w^a is

$$\text{WMD}_{w^a}^d = \text{WMD}_{w^a} - \text{WMD}_{w^b} \quad (8)$$

if there exists such a w^b that $w^b = w^a$. In other cases, the $\text{WMD}_{w^a}^d$ is just equal to the WMD_{w^a} . For the rest of the paper, we will use WMD_{w^a} as a short-hand when writing about $\text{WMD}_{w^a}^d$ as the model with difference yields far better and more interpretable results and is the only one we will consider in our analysis.

Finally, the WMD_w values can be calculated using WMD with or without relaxed constraints. However, while the LC-RWMD cannot be used, we will next suggest a way in which its speed can be leveraged to yield reproducible and conservative estimates of the WMD_w values.

4.3 Gale-Shapley matching for WMD

To run WMD on all 50 million pairwise combinations of documents in our corpus (5000^2 at both t_0 and t_1) would unfortunately be computationally prohibitive in many instances. With larger document sets, even using R-WMD with quadratic time complexity could be too costly in terms of time and computational resources, while using the LC-RWMD with linear time complexity sacrifices word-level decomposability. One solution would be to take a random sample from both sets of documents. However, this is not always ideal, as we will demonstrate next.

We first generate 50,000 random pairs of documents (each pair containing one post from each subreddit) at both t_0 and t_1 , yielding a total of 100,000 random pairs. The distribution of distances is displayed in Figure 2. Although a t -test confirms high significance ($t = 13.71$), the measured difference in means is very small (0.548 at t_0 , versus 0.545 at t_1), and it is difficult to determine to what meaningful extent this reflects increased similarity between user language in `r/The_Donald` and `r/conspiracy` over time. We will return to this point later, in Section 5.2.

Looking at Figure 2, it is clear that merely drawing a small random sample from both sets could produce unreliable results. Researchers working with very long documents and/or lacking computational resources might be limited to drawing small samples if running the full WMD model. If asking whether there is any evidence of document-level semantic alignment—i.e., a reduction in mean document pair distance over time—present in our corpus in the first place, using a random sample to

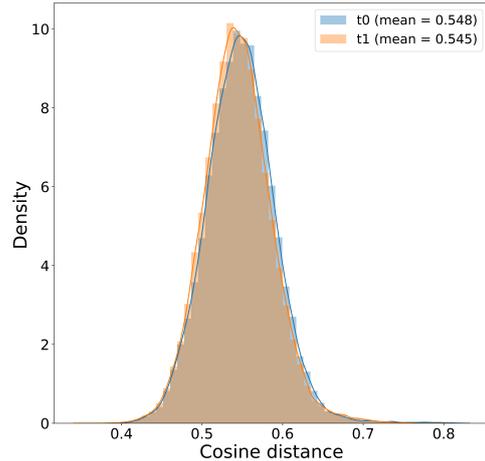


Figure 2: Distribution of WMD distances for random document pairs (one from each subreddit) at t_0 and t_1 .

offset computational costs could therefore produce misleading results. We therefore introduce a second contribution of WMDecompose: the option of effective and consistent pairing of documents using the Gale-Shapley (GS) stable matching algorithm (Gale and Shapley, 1962) to reduce the computational burden incurred by analysis.

Originally introduced through the hypothetical problem of pairing colleges with applicants, the GS algorithm iteratively finds the optimal match between two sets of equal size, given the preferences of all members of the two sets. The optimal match is biased towards the party taking initiative, i.e. the “suitor” (Dubins and Freedman, 1981; Iwama and Miyazaki, 2008). To find the “preferences” of documents, we utilize LC-RWMD to first get an approximation of distances between all pairs from two sets of documents. Each document in a set then “prefers” the closest document in the other set. Given these preferences, we can use GS to find the pairs which minimize the distance between the two sets. We posit that GS ensures that our document pairs represent a conservative estimate of the distance between the two sets, making our method reproducible while also ensuring the robustness of results. By reducing computational costs, GS pairing is additionally aligned with mounting calls for NLP and other ML researchers to intentfully pursue algorithms which minimize the growing environmental toll of their technologies (Bender et al., 2021; Strubell et al., 2019).

For our specific case study, we first calculate the LC-RWMD from `r/conspiracy` to `r/The_Donald` and vice versa. This computa-

tion on 50M total pairs requires less than an hour on a regular laptop. Next, these relaxed distances from each document i in `r/conspiracy` to all N documents in `r/The_Donald` are used as the preferences of document i vis-a-vis documents $1, \dots, N$. We are primarily interested in movement from `r/conspiracy` to `r/The_Donald`, hence the former is given the role of “suitor”.

Then, the Gale-Shapley algorithm is executed to find the optimal match, i.e., the set of document pairs that is least costly to move between `r/conspiracy` and `r/The_Donald`. Effectively, this shows one optimal solution to moving the mass of words from former to the latter, if each document in the first set were to be transferred only to the most similar document in the other set. All these steps are repeated for the documents at t_0 and t_1 . To summarize, by matching pairs of documents which are more similar in semantic space, `WMDecompose` produces conservative measures of semantic distance between corpora. Thus, in addition to providing a non-random, reproducible means of reducing computing costs while retaining full WMD calculations, GS ensures the robustness of findings.

4.4 Exploring lexical trends with `WMDecompose`

Once we have the optimal pairs between `r/conspiracy` and `r/The_Donald`, we can run full WMD for the two sets of pairs, i.e. the set at t_0 and the set at t_1 ⁴. We do this using `wmdecompose`, a Python package written specifically for this paper with EMD executed using the `PyEMD` library under the hood (Pele and Werman, 2008, 2009).⁵

5 Results

5.1 Overall document distance

After generating document pairs with the Gale-Shapley algorithm, it is worth asking whether mean document distances using GS pairs follow a similar pattern than those generated by random pairs (as in Figure 2). As it turns out, they do; both are normally distributed, with pair distances having a mean of 0.478 at t_0 (versus 0.548 with random

pairing), and 0.469 at t_1 (versus 0.545 with random pairing). A t -test again shows significance ($t = 8.05$), but the absolute difference is small enough as to be difficult to interpret at the document level. We can therefore conclude that, when pairing documents on the basis of semantic proximity with GS and when pairing documents randomly, some modest but statistically significant reduction in distance is taking place from t_0 to t_1 . Without further analysis, however, we cannot conclude much else.

5.2 Identifying distinguishing words with `WMDecompose`

As we have stated throughout, document-level distance measures tell one nothing about the nature and source of that distance. A researcher doing exploratory analysis of a new dataset might look at the relatively small, albeit significant, difference in mean document distance at t_0 and t_1 and conclude, absent any other contextual information, that further analysis is not warranted.

However, distances between documents merely represent the distances between their aggregate words. As such, it is quite possible that relative document-level stability is masking a much greater degree of variation between t_0 and t_1 for specific words, and these words might merit further attention. Figure 3 displays the distribution of changes in word cost (i.e., the total distance contributed at t_1 minus the total distance contributed at t_0) for each unique word in the corpus. As we can see, despite the longitudinal stability of the corpora at the document level, a great deal of lexical and semantic change over time is nevertheless taking place. Though the majority of words show little change in this regard, many words which significantly distinguished the corpora at t_0 no longer do at t_1 , and vice versa. Our suggestion is that these words might be qualitatively instructive, motivating a closer inspection of the lexical data produced by `WMDecompose`. We therefore turn to `WMDecompose` to examine the sorted list of words and clusters which most distinguish our corpora at each time period, displayed in Table 1.

These top words conform nicely with expectations, with vocabulary directly associated with party and electoral politics characterizing `r/The_Donald`, and words associated with conspiracy theories characterizing their eponymous subreddit. Further, we see the clear effects of temporality and current events: words associated with

⁴Relaxed WMD can also be used to further speed up the calculations. While using R-WMD gives similar results as the full WMD, we focus here on the full WMD results as this is the baseline we are looking to establish.

⁵<https://github.com/laszukdawid/PyEMD>

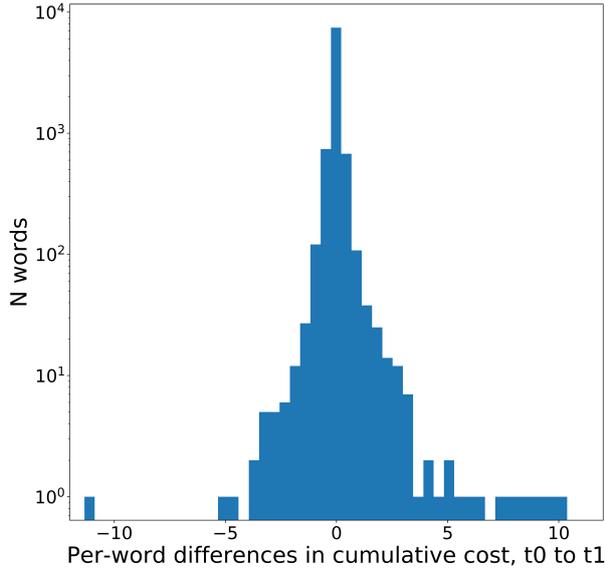


Figure 3: The distribution of word distance changes over time, calculated as the aggregate distance each word contributed to t_1 document pairs minus the aggregate distance it added to t_0 document pairs. Despite the general stability of document-level distances, many individual words show considerable differences in distance contributed from t_0 to t_1 .

the 2016 election cycle (*bernie, hillary, cruz, delegate*) disappear at t_1 , while more general political terms (*democrat, president, liberal*) appear; for *r/conspiracy*, *epstein* and *cia* enter the top ten most distinctive words, while *isis* drops out.

We can also note that, though the top word distances are higher for *r/The_Donald* at each period, the difference greatly shrinks, reflecting both an attenuation of the ritualized, insider language associated with Trump’s online following during the 2016 election cycle, as well as an increase in talk related to Trump on *r/conspiracy*. While these results are perhaps unsurprising, they demonstrate the greater richness of WMDecompose for the comparative analysis of corpora by introducing word-level data. Clustering words similarly allows for the discovery of pervasive thematic differences between corpora. Table 2 displays some example clusters, separated by subreddit and time period.

5.3 Inductive discovery of thematic assimilation over time

While WMDecompose can be used to identify distinguishing keywords, it can also be used to uncover fine-grained semantic assimilation. If a word w strongly distinguishes corpus S^a from corpus S^b at t_0 , but no longer at t_1 , it could represent assim-

The_Donald (t_0)		conspiracy (t_0)	
Word	Cost	Word	Cost
trump	45.85	conspiracy	8.79
vote	14.96	government	6.67
bernie	10.79	world	3.90
hillary	10.49	israel	3.72
cruz	10.01	find	3.66
win	9.58	video	3.56
donald	9.58	information	3.08
donald_trump	8.81	documentary	3.00
delegate	7.95	theory	2.97
cuck	7.50	isis	2.97
The_Donald (t_1)		conspiracy (t_1)	
Word	Cost	Word	Cost
trump	12.83	conspiracy	11.17
vote	10.70	epstein	8.97
democrat	6.50	video	5.30
president	6.45	control	4.97
dem	5.90	vaccine	4.46
left	5.12	theory	3.94
pedes	4.76	world	3.69
election	4.74	find	3.46
liberal	4.70	government	3.33
win	4.33	cia	2.89

Table 1: Words contributing the most cost between subreddits at t_0 and t_1 .

ilation (S^b starts using w and related words more frequently), but it could equally represent S^a ceasing to use w (as is often the case with words related to current events).

As such, the change in the distance contribution of a word over time must be considered alongside the change in word frequency. We might thus conceive of each word distance change as occupying a position in three-dimensional space, with axes corresponding to 1) change in aggregate word cost, 2) change in frequency in S^a , and 3) change in frequency in S^b . The same conceptualization can be applied to word clusters as well, wherein a cluster’s values are simply the sum of its words’ values.

Given our interest in semantic convergence, cases in which word distance contributions go down because usage ceases are of little interest (though they might interest others). Rather, we are interested in words which contribute less distance, while exhibiting comparable or increased use in both subreddits. Examining words meeting these criteria, and excluding low-frequency outliers, we indeed see many conspiratorial words whose cu-

Subcorpus	Cluster Theme	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8
The_Donald (t_0)	2016 primary election	cruz	kasich	campaign	ted_cruz	republican	rubio	party	vp
The_Donald (t_0)	pro-Trump slang	trump	donald	donald_trump	maga	high_energy	centipede	america_great	meme
The_Donald (t_1)	immigration	wall	illegal	border	immigrant	build_wall	census	illegal_alien	illegal_immigration
The_Donald (t_1)	Mueller report	mueller	kavanaugh	trump_supporter	office	hillary	comey	barr	collusion
conspiracy (t_0)	surveillance	cia	nsa	secret	operation	agency	agent	intelligence	surveillance
conspiracy (t_0)	mass shootings	shooting	police	shoot	shooter	suspect	sandy_hook	kill	paris_attack
conspiracy (t_1)	Epstein	epstein	jeffrey_epstein	pedophile	ring	claim	pedophilia	pedo	dr
conspiracy (t_1)	aliens	alien	moon	earth	nasa	ufo	space	planet	sun

Table 2: A selection of clusters produced by WMDecompose. Clustering facilitates inductive thematic discovery by grouping semantically similar words.

Word	CC (%)	SWiC
hoax	-99.2	discredit, truth, bogus
fraudulent	-98.5	theft, conspire, scam
brainwash	-95.3	teacher, public_school
threat	-67.0	mitigate, deter
jews	-60.4	jewish, zionist
surveillance	-49.6	nsa, agency, agent
alex_jones	-39.0	infowars, interview
propaganda	-33.8	false_flag, agenda
reality	-23.9	veil, perceive

Table 3: CC (%) = Cost change from t_0 to t_1 , represented as a percentage, SWiC = Similar words in cluster (i.e., words from the same cluster whose cost also decreased from t_0 to t_1 , while remaining in use in both subreddits).

mulative distance contribution shrank from t_0 to t_1 . A selection of such words is presented in Table 3. These include terms such as *hoax*, *fraudulent*, *brainwash*, *threat*, *jews*, *surveillance*, *alex_jones*, *propaganda*, and *reality*, to name a few. We can then examine other words in these keywords’ clusters to get a sense of related words following similar patterns of cost reduction and continued usage, as are displayed in the table.

Of course, such observations are of exploratory nature, intended to demonstrate an inductive starting point for a more rigorous analysis, either within or outside the WMDecompose framework. We do not posit this as statistical proof of any process of assimilation. However, this general framework of comparative and longitudinal analysis allows for a much richer engagement with discursive change due to the robust semantic relationships captured by WMDecompose while still providing simple outputs, e.g., for significance testing. Despite its cursory nature, we hope this short vignette has demonstrated the interpretable potential of WMDecompose, and how it might aid social science and humanities researchers with qualitative

thematic discovery in large corpora, while also providing a computable metric for quantitative models.

6 Conclusion

We have introduced a novel iteration of WMD, one which departs from the WMD variations which, despite their high performance on a variety of tasks, do away with one of the key contributions of the original model: the inherent and decomposable relationship between document-level semantic distances and the lexical semantics that underlie them. In so doing, we heed a recent, but hopefully long-lived, call for ML and particularly NLP researchers to prioritize models which are interpretable, providing explanatory value at the level of social meaning. We hope that our Python package might aid future researchers similarly interested in interpretable computational approaches. Furthermore, through the Gale-Shapley algorithm, we propose an approach for combining interpretability with environmental sustainability. Down the line, this framework could also be expanded to support dynamic embeddings from models such as BERT, which due to their technical opacity have yet to be widely adapted in sociocultural analysis.

Our vignette presented how one might use WMDecompose in a comparative and exploratory context. That is, we imagine this as somewhat of an inductive starting point—in this case, for an analysis of the relationship between Trumpist and conspiracy theorist online communities—revealing trends related to one’s particular research question that can be pursued with more focused and rigorous analysis, be it qualitative, discourse analytic, or statistical. However, because of the rich, word-level quantitative measures WMDecompose provides, future research might attempt to employ it in, for example, causal models seeking to estimate the effect of a treatment condition on particular lexical features, time series analyses, and so on.

References

- Kubilay Atasu, Thomas Parnell, Celestine Dünner, Manolis Sifalakis, Haralampos Pozidis, Vasileios Vasileiadis, Michail Vlachos, Cesar Berrospi, and Abdel Labbi. 2017. [Linear-complexity relaxed word Mover’s distance with GPU acceleration](#). In *2017 IEEE International Conference on Big Data (Big Data)*, pages 889–896.
- Michael Barkun. 2017. [President Trump and the “Fringe”](#). *Terrorism and Political Violence*, 29(3):437–443. Publisher: Routledge. eprint: <https://doi.org/10.1080/09546553.2017.1313649>.
- Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. [The Pushshift Reddit Dataset](#). *arXiv:2001.08435 [cs]*. ArXiv: 2001.08435.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. [On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?](#) In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT ’21*, pages 610–623, New York, NY, USA. Association for Computing Machinery.
- Yochai Benkler, Robert Faris, Hal Roberts, and Ethan Zuckerman. 2017. [Study: Breitbart-led right-wing media ecosystem altered broader media agenda](#). *Columbia Journalism Review*, 3(2).
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. [Enriching Word Vectors with Subword Information](#). *arXiv:1607.04606 [cs]*. ArXiv: 1607.04606.
- Lorna Bracewell. 2021. [Gender, Populism, and the QAnon Conspiracy Movement](#). *Frontiers in Sociology*, 5:1–14.
- Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. [A Survey of the State of Explainable AI for Natural Language Processing](#). In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459, Suzhou, China. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- L. E. Dubins and D. A. Freedman. 1981. [Machiavelli and the Gale-Shapley Algorithm](#). *The American Mathematical Monthly*, 88(7):485–494. Publisher: Mathematical Association of America.
- Charles Elkan. 2003. [Using the triangle inequality to accelerate k-means](#). In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning, ICML’03*, pages 147–153, Washington, DC, USA. AAAI Press.
- Christopher M. Federico, Allison L. Williams, and Joseph A. Vitriol. 2018. [The role of system identity threat in conspiracy theory endorsement](#). *European Journal of Social Psychology*, 48(7):927–938.
- D. Gale and L. S. Shapley. 1962. [College Admissions and the Stability of Marriage](#). *The American Mathematical Monthly*, 69(1):9–15.
- K. J. Gülle, N. Ford, P. Ebel, F. Brokhausen, and A. Vögelsang. 2020. [Topic Modeling on User Stories using Word Mover’s Distance](#). In *2020 IEEE Seventh International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, pages 52–60.
- Daniel C. Hellinger. 2018. *Conspiracies and Conspiracy Theories in the Age of Trump*. Springer. Google-Books-ID: U3hvDwAAQBAJ.
- Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.
- Matthew J. Hornsey, Matthew Finlayson, Gabrielle Chatwood, and Christopher T. Begey. 2020. [Donald Trump and vaccination: The effect of political identity, conspiracist ideation and presidential tweets on vaccine hesitancy](#). *Journal of Experimental Social Psychology*, 88:103947.
- Gao Huang, Chuan Guo, Matt J. Kusner, Yu Sun, Fei Sha, and Kilian Q. Weinberger. 2016. [Supervised Word Mover’s Distance](#). *Advances in Neural Information Processing Systems*, 29:4862–4870.
- Kazuo Iwama and Shuichi Miyazaki. 2008. [A Survey of the Stable Marriage Problem and Its Variants](#). In *International Conference on Informatics Education and Research for Knowledge-Circulating Society (icks 2008)*, pages 131–136.
- Colin Klein, Peter Clutton, and Adam G. Dunn. 2019. [Pathways to conspiracy: The social and linguistic precursors of involvement in Reddit’s conspiracy theory forum](#). *PLOS ONE*, 14(11):e0225098. Publisher: Public Library of Science.
- Matt J Kusner, Yu Sun, Nicholas I Kolkin, and Kilian Q Weinberger. 2015. [From Word Embeddings To Document Distances](#). *Proceedings of the 32nd International Conference on Machine Learning*, 37:957–966.
- S. Lloyd. 1982. [Least squares quantization in PCM](#). *IEEE Transactions on Information Theory*, 28(2):129–137. Conference Name: IEEE Transactions on Information Theory.

- Laurens van der Maaten and Geoffrey Hinton. 2008. [Visualizing Data using t-SNE](#). *Journal of Machine Learning Research*, 9(86):2579–2605.
- Alice Marwick and Rebecca Lewis. 2017. *Media Manipulation and Disinformation Online*. Data & Society Research Institute, New York.
- Joan Massachs, Corrado Monti, Gianmarco De Francisci Morales, and Francesco Bonchi. 2020. [Roots of Trumpism: Homophily and Social Feedback in Donald Trump Support on Reddit](#). In *12th ACM Conference on Web Science, WebSci '20*, pages 49–58, New York, NY, USA. Association for Computing Machinery.
- Leland McInnes, John Healy, and James Melville. 2020. [UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction](#). *arXiv:1802.03426 [cs, stat]*. ArXiv: 1802.03426.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Efficient Estimation of Word Representations in Vector Space](#). *arXiv:1301.3781 [cs]*. ArXiv: 1301.3781.
- Joanne M. Miller, Kyle L. Saunders, and Christina E. Farhart. 2016. [Conspiracy Endorsement as Motivated Reasoning: The Moderating Roles of Political Knowledge and Trust](#). *American Journal of Political Science*, 60(4):824–844.
- Md Nasir, Sandeep Nallan Chakravarthula, Brian Baucum, David C. Atkins, Panayiotis Georgiou, and Shrikanth Narayanan. 2019. [Modeling Interpersonal Linguistic Coordination in Conversations using Word Mover’s Distance](#). *arXiv:1904.06002 [cs]*. ArXiv: 1904.06002.
- Ryan Neville-Shepard. 2019. [Post-presumption argumentation and the post-truth world: on the conspiracy rhetoric of Donald Trump](#). *Argumentation and Advocacy*, 55(3):175–193. Publisher: Routledge. eprint: <https://doi.org/10.1080/10511431.2019.1603027>.
- Rishab Nithyanand, Brian Schaffner, and Phillipa Gill. 2017. [Online Political Discourse in the Trump Era](#). *arXiv:1711.05303 [cs]*. ArXiv: 1711.05303.
- Ofir Pele and Michael Werman. 2008. [A Linear Time Histogram Metric for Improved SIFT Matching](#). In *Computer Vision – ECCV 2008*, Lecture Notes in Computer Science, pages 495–508, Berlin, Heidelberg. Springer.
- Ofir Pele and Michael Werman. 2009. [Fast and robust Earth Mover’s Distances](#). In *2009 IEEE 12th International Conference on Computer Vision*, pages 460–467. ISSN: 2380-7504.
- Jeffrey Pennington, 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)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Shruti Phadke, Mattia Samory, and Tanushree Mitra. 2021. [What Makes People Join Conspiracy Communities? Role of Social Factors in Conspiracy Engagement](#). *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW3):223:1–223:30.
- Francesca Polletta and Jessica Callahan. 2019. [Deep Stories, Nostalgia Narratives, and Fake News: Storytelling in the Trump Era](#). In Jason L. Mast and Jeffrey C. Alexander, editors, *Politics of Meaning/Meaning of Politics: Cultural Sociology of the 2016 U.S. Presidential Election*, Cultural Sociology, pages 55–73. Springer International Publishing, Cham.
- Marcus Pöckelmann, Janis Dähne, Jörg Ritter, and Paul Molitor. 2020. [Fast paraphrase extraction in Ancient Greek literature](#). *it - Information Technology*, 62(2):75–89. Publisher: De Gruyter Oldenbourg Section: it - Information Technology.
- Fuji Ren and Ning Liu. 2018. [Emotion computing using Word Mover’s Distance features based on Ren_cecps](#). *PLOS ONE*, 13(4):e0194136. Publisher: Public Library of Science.
- Manoel Horta Ribeiro, Shagun Jhaver, Savvas Zannettou, Jeremy Blackburn, Emiliano De Cristofaro, Gianluca Stringhini, and Robert West. 2020. [Does Platform Migration Compromise Content Moderation? Evidence from r/The_donald and r/Incels](#). *arXiv:2010.10397 [cs]*. ArXiv: 2010.10397.
- Y. Rubner, C. Tomasi, and L.J. Guibas. 1998. [A metric for distributions with applications to image databases](#). In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 59–66.
- Mattia Samory and Tanushree Mitra. 2018. [Conspiracies Online: User Discussions in a Conspiracy Community Following Dramatic Events](#). *Proceedings of the International AAAI Conference on Web and Social Media*, 12(1). Number: 1.
- Ryan P. Shepherd. 2020. [Gaming Reddit’s Algorithm: r/the_donald, Amplification, and the Rhetoric of Sorting](#). *Computers and Composition*, 56:1–14.
- Dominik A. Stecula and Mark Pickup. 2021. [How populism and conservative media fuel conspiracy beliefs about COVID-19 and what it means for COVID-19 behaviors](#). *Research & Politics*, 8(1):2053168021993979. Publisher: SAGE Publications Ltd.
- Dustin S. Stoltz and Marshall A. Taylor. 2019. [Concept Mover’s Distance: measuring concept engagement via word embeddings in texts](#). *Journal of Computational Social Science*, 2(2):293–313.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. [Energy and Policy Considerations for Deep Learning in NLP](#). *arXiv:1906.02243 [cs]*. ArXiv: 1906.02243.

Marshall A. Taylor and Dustin S. Stoltz. 2020. [Concept Class Analysis: A Method for Identifying Cultural Schemas in Texts](#). *Sociological Science*, 7:544–569.

Jesmin Jahan Tithi and Fabrizio Petrini. 2021. [An Efficient Shared-memory Parallel Sinkhorn-Knopp Algorithm to Compute the Word Mover’s Distance](#). *arXiv:2005.06727 [cs, stat]*. ArXiv: 2005.06727.

Joseph E. Uscinski, Adam M. Enders, Casey Klofstad, Michelle Seelig, John Funchion, Caleb Everett, Stefan Wuchty, Kamal Premaratne, and Manohar Murthi. 2020. [Why do people believe COVID-19 conspiracy theories?](#) *Harvard Kennedy School Misinformation Review*, 1(3).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is All you Need](#). *Advances in Neural Information Processing Systems*, 30.

Matheus Werner and Eduardo Laber. 2020. [Speeding up Word Mover’s Distance and Its Variants via Properties of Distances Between Embeddings](#). *ECAI 2020*, pages 2204–2211. Publisher: IOS Press.

X. Wu and H. Li. 2017. [Topic mover’s distance based document classification](#). In *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, pages 1998–2002. ISSN: 2576-7828.

Sho Yokoi, Ryo Takahashi, Reina Akama, Jun Suzuki, and Kentaro Inui. 2020. [Word Rotator’s Distance](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2944–2960, Online. Association for Computational Linguistics.

A Text preprocessing and word2vec fine-tuning details

The raw text data from the Pushshift dataset was processed as follows, all conducted in Python. Posts were first lowercased, and URLs were removed via regular expressions; stopwords were removed and remaining words were lemmatized, both using spaCy ([Honnibal et al., 2020](#)); mark-down text and other special characters were removed using custom regex functions; finally, words occurring fewer than 20 times in the corpus (i.e., less than once every 1,000 documents) were removed.

Once processed, the text was embedded with a word2vec model that was fine-tuned on a corpus of 533K self posts from these subreddits as well as other conspiracy and conservative subreddits, with preprocessing following the same steps as above. Fine-tuning was done using the skipgram implementation of word2vec, with negative sampling,

Word	Positive	Negative	
	Cost	Word	Cost
great	41.79	told	19.64
best	33.15	order	19.58
amazing	31.35	rude	18.10
friendly	27.02	never	17.94
delicious	26.18	worst	17.44
highly_recommend	24.93	minute	15.14
staff	19.96	hour	14.97
definitely	19.52	said	14.91
always	18.68	money	13.89
favorite	18.42	terrible	13.57
massage	18.29	called	13.14
portland	16.15	asked	13.05

Table 4: Decomposed word-level WMD_w distances for moving from the full set of positive reviews (left) the the full set of negative reviews and vice versa (right). Only the 12 words w that contributed the most are shown.

a context window of 10 tokens, over four epochs and with a learning rate of 0.01. The model was phrased in a similar manner to the recommendations in [Mikolov et al. \(2013\)](#), such that frequently co-occurring bigrams and trigrams were encoded as single lexical entities (e.g., *president_trump*). Further, each document was represented using Tf-Idf and L1-normalization, in order to prevent frequent words from crowding out information from less common but more salient words.

B Sanity Check with Yelp Reviews

To further demonstrate the qualities of `WMDDecompose`, we offer an additional case study using a set of reviews from Yelp. This dataset is intuitive and well-known, and should hence complement our main analysis, as it requires less domain-knowledge for illustrating the basic qualities of `WMDDecompose`. To facilitate comparison with earlier work on the WMD, we here rely on Euclidean distances, the metric which was used in the original WMD paper ([Kusner et al., 2015](#)), even though cosine distances are arguably preferable when working with semantic similarity tasks and therefore used in the main analysis of the paper. For the analysis in this appendix, we look at trends that are highly predicable, in order to provide a “sanity check” to ensure that our model was behaving as expected. Consequently, we looked at the WMD_w and CMD_c from positive

		Positive to Negative Clusters									
		56	51	92	76	26	30	33	78	47	88
Word (distance)	great (30.91)	made (3.18)	good (7.14)	delicious(21.13)	spot (8.24)	friendly(20.8)	professional (9.19)	year (4.98)	massage (16.45)	quality (2.12)	
	best (26.33)	felt (2.85)	really (5.9)	favorite (15.06)	place (6.2)	staff (14.39)	team (6.35)	new (2.89)	relaxing (3.83)	price (1.97)	
	amazing (25.21)	soon (2.54)	nice (5.87)	loved (7.27)	town (5.07)	everyone (6.24)	thorough (3.54)	moved (2.77)	facial (3.72)	worth (1.95)	
	love (21.88)	quickly (2.07)	well (4.3)	fresh (7.24)	found (3.77)	glad (3.36)	feel_comfortable (3.48)	weekly (1.58)	spa (3.48)	option (1.94)	
	highly_recommend (20.17)	hand (1.87)	kind (4.27)	tasty (4.78)	glad (3.36)	welcoming (4.06)	job (2.84)	st (1.03)	treatment (3.15)	beat (1.74)	
	definitely (15.54)	house (1.72)	lot (3.38)	try (4.7)	looking (3.0)	accommodating (3.77)	work (2.74)	year_ago (0.98)	brow (2.81)	every_penny (11.24)	
	always (15.27)	next (1.36)	little (3.34)	enjoyed (4.43)	visit (2.95)	pleasant (3.28)	care (2.61)	first (0.86)	gentle (2.47)	size (0.95)	
	excellent (12.16)	around (1.22)	also (2.79)	must (4.39)	friend (2.62)	personable (2.76)	efficient (2.27)	monthly (0.73)	calming (2.32)	deal (0.95)	
	thank (10.63)	along (1.04)	enough (2.29)	die (2.95)	gem (2.52)	smile (2.38)	felt_comfortable (2.19)	yr (0.71)	hot_stone (2.2)	plus (0.9)	
	super (10.52)	totally (1.04)	much (2.0)	special (2.53)	coming (2.21)	met (1.46)	grateful (2.08)	last_week (0.59)	lash (1.57)	penny (0.88)	
		Negative to Positive Clusters									
		51	92	44	95	35	52	56	26	78	0
Word (distance)	never (15.84)	like (4.84)	order (15.17)	worst (14.89)	told (16.11)	use (3.06)	ever (2.14)	review (5.09)	went (5.2)	terrible (11.24)	
	hour (11.77)	ok (3.46)	minute (12.0)	money (11.35)	said (11.41)	issue (2.41)	treat (1.23)	disappointed (3.2)	closed (4.6)	bad (7.88)	
	would (10.22)	hard (3.31)	waiting (7.83)	horrible (9.74)	asked (10.94)	follow (2.17)	unbelievable (0.98)	close (2.49)	month_later (1.89)	awful (7.0)	
	u (7.87)	maybe (3.27)	table (7.56)	customer_service (6.39)	manager (9.14)	problem (1.9)	cannot (0.63)	decided (1.6)	changed (1.87)	poor (6.84)	
	could (6.59)	however (2.43)	waited (6.24)	waste_time (5.66)	refused (4.05)	frustrating (1.66)	allaround (0.28)	find (1.3)	update (1.85)	star (4.98)	
	another (6.41)	used (2.41)	min (5.72)	please (4.4)	ask (2.86)	need (1.51)	every_aspect (0.24)	read_review (1.26)	day (1.8)	slow (4.86)	
	even (6.29)	seemed (2.33)	sitting (3.8)	unless (3.82)	asking (2.41)	properly (1.48)	everyday (0.2)	heard (1.24)	month_ago (1.68)	worse (3.8)	
	nothing (6.06)	fine (2.31)	waiter (3.71)	somewhere_else (3.63)	mistake (2.26)	multiple (1.27)	comment (0.2)	opened (0.82)	last (1.44)	disappointing (3.16)	
	give (5.36)	least (3.56)	hostess (3.59)	avoid (2.05)	stated (2.05)	due (1.17)	speedy (0.16)	somewhere (0.63)	twice (1.43)	sad (2.96)	
	left (5.35)	something (1.71)	line (3.32)	suck (2.5)	informed (1.87)	failed (1.12)	always (0.13)	first_impression (0.62)	month (1.23)	unfortunately (2.63)	

Table 5: The CMD_c for the top 10 c when moving from the positive review set to the negative (top) and vice versa (bottom). The keywords and their order for each cluster are determined by the WMD_w of each word.

and negative reviews to see whether the distance would, as expected, be composed mainly of different polarized sentiment words, such as *good* and *bad*.

For this analysis, we use the latest version of the Yelp review dataset, accessed in late July 2021⁶. The data was filtered to only include reviews written in the cities of Atlanta and Portland, two geographically and demographically distinct cities, after December 2017. We further filtered the data to only include reviews for establishments that were labeled with the categories *Restaurant* or *Health Medical*. This way we wanted to show that category-specific trends would also emerge among positive and negative keywords, as well as explore whether any trends specific for the two cities would appear in the results. Reviews were then further filtered so that only highly positive (5 stars) or highly negative (1 star) were included. These were labeled positive and negative, respectively. Reviews were sampled from this pool, so that 2000 reviews were selected from both Atlanta and Portland, 500 negative and 500 positive for both restaurants and health services, for a total of 4000 reviews.

B.1 Preprocessing and word vectors

For the purposes of this analysis, we fine-tuned the same pretrained word2vec model that was used for the main analysis, with the same hyperparameters, but using the filtered Yelp dataset. A series of preprocessing steps were also conducted, including removal of URLs and stopwords and phrasing as proposed by Mikolov et al. (2013).

⁶<https://www.yelp.com/dataset/>

B.2 Clustering and Gale-Shapley

After preprocessing, clusters were detected in the word vectors using K-means. Through inspecting Silhouette scores and elbow plots, the number of clusters was chosen to be 100. Further, instead of performing the analysis on the full set of pairs, we first ran LC-RWMD to find the R-WMD between all pairs and then the Gale-Shapley (GS) algorithm to find the set of optimal pairs.

B.3 Results from WMDDecompose

Next, the pairs from the GS matching were analysed using WMDDecompose. The words that added the most distance (with difference) when moving from the positive set to the negative and vice versa can be seen in Table 4. As expected, the top set of WMD_w word-level distances for the positive documents is dominated by positive sentiment words, with *great*, *best* and *amazing* on top. However, other trends also appear, such as words specific to restaurants (*delicious*) or health services (*massage*). Interestingly enough, the word *portland* appears as a high expense when moving from the positive to the negative set, indicating that positive reviewers located in that city might be more likely to mention it by name.

On the side of WMD_w word-level distances, the top words include negative sentiment words such as *rude*, *worst* and *terrible* when moving from the negative to the positive reviews, but other patterns are also visible. Words about time, such as *hour*, *minute* and the word *money*, contribute heavily to the semantic distance from negative to positive reviews, as do verbs often associated with verbal commands, such as *told* and *said*. Here, words related to the cities or categories of the reviews do

not make the top 12 cut, although category specific words are present when looking at the top 50 (including *insurance*, *manager* and *doctor*).

Switching focus to the CMD_c clusters displayed in Table 5, we see some of the top words from Table 4, now organized by cluster. While there is some overlap in clusters when moving from positive to negative and vice versa (clusters 51, 92, 56, 26, 78), the keywords that define the clusters are different, as they are determined using the top ten WMD_w values of each word in the cluster. In Table 5, we also see how category specific words, such as those in cluster 47 related to spa and massage services, add a significant distance when moving from positive to negative reviews.