Characterizing the impact of geometric properties of word embeddings on task performance

Brendan Whitaker[1]*; **Denis Newman-Griffis**[1]*; **Aparajita Haldar**[2]*;
Hakan Ferhatosmanoglu[2], **Eric Fosler-Lussier**[1]
[1]The Ohio State University, Columbus, OH, USA
[2]University of Warwick, Coventry, UK
{whitaker.213, newman-griffis.1, fosler-lussier.1}@osu.edu
{aparajita.haldar, h.ferhatosmanoglu}@warwick.ac.uk

Abstract

Analysis of word embedding properties to inform their use in downstream NLP tasks has largely been studied by assessing nearest neighbors. However, geometric properties of the continuous feature space contribute directly to the use of embedding features in downstream models, and are largely unexplored. We consider four properties of word embedding geometry, namely: position relative to the origin, distribution of features in the vector space, global pairwise distances, and local pairwise distances. We define a sequence of transformations to generate new embeddings that expose subsets of these properties to downstream models and evaluate change in task performance to understand the contribution of each property to NLP models. We transform publicly available pretrained embeddings from three popular toolkits (word2vec, GloVe, and FastText) and evaluate on a variety of intrinsic tasks, which model linguistic information in the vector space, and extrinsic tasks, which use vectors as input to machine learning models. We find that intrinsic evaluations are highly sensitive to absolute position, while extrinsic tasks rely primarily on local similarity. Our findings suggest that future embedding models and post-processing techniques should focus primarily on similarity to nearby points in vector space.

1 Introduction

Learned vector representations of words, known as word embeddings, have become ubiquitous throughout natural language processing (NLP) applications. As a result, analysis of embedding spaces to understand their utility as input features has emerged as an important avenue of inquiry, in order to facilitate proper use of embeddings in downstream NLP tasks. Many analyses have focused on nearest neighborhoods, as a viable proxy for semantic information (Rogers et al., 2018; Pierrejean and Tanguy, 2018). However, neighborhood-based analysis is limited by the unreliability of nearest neighborhoods (Wendlandt et al., 2018). Further, it is intended to evaluate the *semantic content* of embedding spaces, as opposed to characteristics of the feature space itself.

Geometric analysis offers another recent angle from which to understand the properties of word embeddings, both in terms of their distribution (Mimno and Thompson, 2017) and correlation with downstream performance (Chandrahas et al., 2018). Through such geometric investigations, neighborhood-based semantic characterizations are augmented with information about the continuous feature space of an embedding. Geometric features offer a more direct connection to the assumptions made by neural models about continuity in input spaces (Szegedy et al., 2014), as well as the use of recent contextualized representation methods using continuous language models (Peters et al., 2018; Devlin et al., 2018).

In this work, we aim to bridge the gap between neighborhood-based semantic analysis and geometric performance analysis. We consider four components of the geometry of word embeddings, and transform pretrained embeddings to expose only subsets of these components to downstream models. We transform three popular sets of embeddings, trained using word2vec (Mikolov et al., 2013),[1] GloVe (Pennington et al., 2014),[2] and FastText (Bojanowski et al., 2017),[3] and use the resulting embeddings in a battery of standard evaluations to measure changes in task performance.

We find that intrinsic evaluations, which model linguistic information directly in the vector space,

*These authors contributed equally to this work.

[1]3M 300-d GoogleNews vectors from https://code.google.com/archive/p/word2vec/

[2]2M 300-d 840B Common Crawl vectors from https://nlp.stanford.edu/projects/glove/

[3]1M 300-d WikiNews vectors with subword information from https://fasttext.cc/docs/en/english-vectors

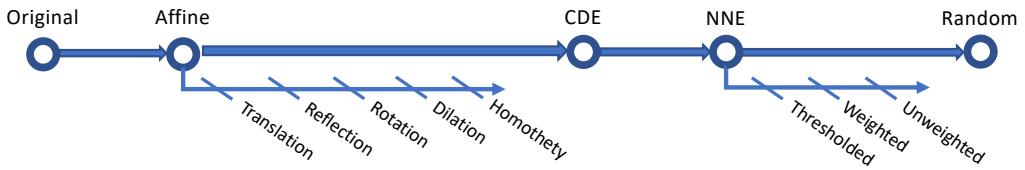


Figure 1: Sequence of transformations applied to word embeddings, including transformation variants. Note that each transformation is applied independently to source word embeddings. Transformations are presented in order of decreasing geometric information retained about the original vectors.

are highly sensitive to absolute position in pretrained embeddings; while extrinsic tasks, in which word embeddings are passed as input features to a trained model, are more robust and rely primarily on information about local similarity between word vectors. Our findings, including evidence that global organization of word vectors is often a major source of noise, suggest that further development of embedding learning and tuning methods should focus explicitly on local similarity, and help to explain the success of several recent methods.

2 Related Work

Word embedding models and outputs have been analyzed from several angles. In terms of performance, evaluating the "quality" of word embedding models has long been a thorny problem. While intrinsic evaluations such as word similarity and analogy completion are intuitive and easy to compute, they are limited by both confounding geometric factors (Linzen, 2016) and task-specific factors (Faruqui et al., 2016; Rogers et al., 2017). Chiu et al. (2016) show that these tasks, while correlated with some semantic content, do not always predict downstream performance. Thus, it is necessary to use a more comprehensive set of intrinsic and extrinsic evaluations for embeddings.

Nearest neighbors in sets of embeddings are commonly used as a proxy for qualitative semantic information. However, their instability across embedding samples (Wendlandt et al., 2018) is a limiting factor, and they do not necessarily correlate with linguistic analyses (Hellrich and Hahn, 2016). Modeling neighborhoods as a graph structure offers an alternative analysis method (Cuba Gyllensten and Sahlgren, 2015), as does 2-D or 3-D visualization (Heimerl and Gleicher, 2018). However, both of these methods provide qualitative insights only. By systematically analyzing geometric information with a wide variety of evaluations, we provide a quantitative counterpart to these understandings of embedding spaces.

3 Methods

In order to investigate how different geometric properties of word embeddings contribute to model performance on intrinsic and extrinsic evaluations, we consider the following attributes of word embedding geometry:

- position relative to the origin;
- distribution of feature values in $\mathbb{R}^d$;
- global pairwise distances, i.e. distances between any pair of vectors;
- local pairwise distances, i.e. distances between nearby pairs of vectors.

Using each of our sets of pretrained word embeddings, we apply a variety of transformations to induce new embeddings that only expose subsets of these attributes to downstream models. These are: affine transformation, which obfuscates the original position of the origin; cosine distance encoding, which obfuscates the original distribution of feature values in $\mathbb{R}^d$; nearest neighbor encoding, which obfuscates global pairwise distances; and random encoding. This sequence is illustrated in Figure 1, and the individual transformations are discussed in the following subsections.

General notation for defining our transformations is as follows. Let W be our vocabulary of words taken from some source corpus. We associate with each word $w \in W$ a vector $\mathbf{v} \in \mathbb{R}^d$ resulting from training via one of our embedding generation algorithms, where d is an arbitrary dimensionality for the embedding space. We define V to be the set of all pretrained word vectors $\mathbf{v}$ for a given corpus, embedding algorithm, and parameters. The matrix of embeddings M_V associated with this set then has shape $|V| \times d$. For simplicity, we restrict our analysis to transformed embeddings of the same dimensionality d as the original vectors.

3.1 Affine transformations

Affine transformations have been previously utilized for post-processing of word embeddings. For example, Artetxe et al. (2016) learn a matrix transform to align multilingual embedding spaces, and Faruqui et al. (2015) use a linear sparsification to better capture lexical semantics. In addition, the simplicity of affine functions in machine learning contexts (Hofmann et al., 2008) makes them a good starting point for our analysis.

Given a set of embeddings in $\mathbb{R}^d$, referred to as an **embedding space**, affine transformations

$$f_{\text{affine}} : \mathbb{R}^d \to \mathbb{R}^d$$

change positions of points relative to the origin.

While prior work has typically focused on linear transformations, which fix the origin, we consider the broader class of affine transformations, which do not. Thus, affine transformations such as translation cannot in general be represented as a square matrix for finite-dimensional spaces.

We use the following affine transformations:

- translations;
- reflections over a hyperplane;
- rotations about a subspace;
- homotheties.

We give brief definitions of each transformation.

Definition 1. A **translation** is a function $T_{\mathbf{x}} : \mathbb{R}^d \to \mathbb{R}^d$ given by

$$T_{\mathbf{x}}(\mathbf{v}) = \mathbf{v} + \mathbf{x} \tag{3.1}$$

where $\mathbf{x} \in \mathbb{R}^d$.

Definition 2. For every $\mathbf{a} \in \mathbb{R}^d$, we call the map $\text{Refl}_{\mathbf{a}} : \mathbb{R}^d \to \mathbb{R}^d$ given by

$$\text{Refl}_{\mathbf{a}}(\mathbf{v}) = \mathbf{v} - 2\frac{\mathbf{v} \cdot \mathbf{a}}{\mathbf{a} \cdot \mathbf{a}}\mathbf{a} \tag{3.2}$$

the **reflection** over the hyperplane through the origin orthogonal to $\mathbf{a}$.

Definition 3. A **rotation** through the span of vectors $\mathbf{u}, \mathbf{x}$ by angle θ is a map $\text{Rot}_{\mathbf{u},\mathbf{x}} : \mathbb{R}^d \to \mathbb{R}^d$ given by

$$\text{Rot}_{\mathbf{u},\mathbf{x}}(\mathbf{v}) = A\mathbf{v} \tag{3.3}$$

where

$$\begin{aligned} A = I &+ \sin\theta(\mathbf{x}\mathbf{u}^T - \mathbf{u}\mathbf{x}^T) \\ &+ (\cos\theta - 1)(\mathbf{u}\mathbf{u}^T + \mathbf{x}\mathbf{x}^T) \end{aligned} \tag{3.4}$$

and $I \in \text{Mat}_{d,d}(\mathbb{R})$ is the identity matrix.

Definition 4. For every $\mathbf{a} \in \mathbb{R}^d$ and $\lambda \in \mathbb{R} \setminus \{0\}$, we call the map $H_{\mathbf{a},\lambda} : \mathbb{R}^d \to \mathbb{R}^d$ given by

$$H_{\mathbf{a},\lambda}(\mathbf{v}) = \mathbf{a} + \lambda(\mathbf{v} - \mathbf{a}) \tag{3.5}$$

a **homothety** of center $\mathbf{a}$ and ratio λ. A homothety centered at the origin is called a **dilation**.

Parameters used in our analysis for each of these transformations are provided in Appendix A.

3.2 Cosine distance encoding (CDE)

Our cosine distance encoding transformation

$$f_{\text{CDE}} : \mathbb{R}^d \to \mathbb{R}^{|V|}$$

obfuscates the distribution of features in $\mathbb{R}^d$ by representing a set of word vectors as a pairwise distance matrix. Such a transformation might be used to avoid the non-interpretability of embedding features (Fyshe et al., 2015) and compare embeddings based on relative organization alone.

Definition 5. Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$. Then their **cosine distance** $d_{\cos} : \mathbb{R}^d \times \mathbb{R}^d \to [0, 2]$ is given by

$$d_{\cos}(\mathbf{a}, \mathbf{b}) = 1 - \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}||||\mathbf{b}||} \tag{3.6}$$

where the second term is the **cosine similarity**.

As all three sets of embeddings evaluated in this study have vocabulary size on the order of 10^6, use of the full distance matrix is impractical. We use a subset consisting of the distance from each point to the embeddings of the 10K most frequent words from each embedding set, yielding

$$f_{\text{CDE}} : \mathbb{R}^d \to \mathbb{R}^{10^4}$$

This is not dissimilar to the global frequency-based negative sampling approach of word2vec (Mikolov et al., 2013). We then use an autoencoder to map this back to $\mathbb{R}^d$ for comparability.

Definition 6. Let $\mathbf{v} \in \mathbb{R}^{|V|}, \mathbf{W_1}, \mathbf{W_2} \in \mathbb{R}^{|V| \times d}$. Then an **autoencoder** over $\mathbb{R}^{|V|}$ is defined as

$$\mathbf{h} = \varphi(\mathbf{v}\mathbf{W_1}) \tag{3.7}$$

$$\hat{\mathbf{v}} = \varphi(\mathbf{W_2}^T\mathbf{h}) \tag{3.8}$$

Vector $\mathbf{h} \in \mathbb{R}^d$ is then used as the compressed representation of $\mathbf{v}$.

In our experiments, we use ReLU as our activation function φ, and train the autoencoder for 50 epochs to minimize L^2 distance between $\mathbf{v}$ and $\hat{\mathbf{v}}$.

We recognize that low-rank compression using an autoencoder is likely to be noisy, thus potentially inducing additional loss in evaluations. However, precedent for capturing geometric structure with autoencoders (Li et al., 2017b) suggests that this is a viable model for our analysis.

3.3 Nearest neighbor encoding (NNE)

Our nearest neighbor encoding transformation

$$f_{\text{NNE}} : \mathbb{R}^d \rightarrow \mathbb{R}^{|V|}$$

discards the majority of the global pairwise distance information modeled in CDE, and retains only information about nearest neighborhoods. The output of $f_{\text{NNE}}(\mathbf{v})$ is a sparse vector.

This transformation relates to the common use of nearest neighborhoods as a proxy for semantic information (Wendlandt et al., 2018; Pierrejean and Tanguy, 2018). We take the previously proposed approach of combining the output of $f_{\text{NNE}}(\mathbf{v})$ for each $\mathbf{v} \in V$ to form a sparse adjacency matrix, which describes a directed nearest neighbor graph (Cuba Gyllensten and Sahlgren, 2015; Newman-Griffis and Fosler-Lussier, 2017), using three versions of f_{NNE} defined below.

Thresholded The set of non-zero indices in $f_{\text{NNE}}(\mathbf{v})$ correspond to word vectors $\tilde{\mathbf{v}}$ such that the cosine similarity of $\mathbf{v}$ and $\tilde{\mathbf{v}}$ is greater than or equal to an arbitrary threshold t. In order to ensure that every word has non-zero out degree in the graph, we also include the k nearest neighbors by cosine similarity for every word vector. Non-zero values in $f_{\text{NNE}}(\mathbf{v})$ are set to the cosine similarity of $\mathbf{v}$ and the relevant neighbor vector.

Weighted The set of non-zero indices in $f_{\text{NNE}}(\mathbf{v})$ corresponds to only the set of k nearest neighbors to $\mathbf{v}$ by cosine similarity. Cosine similarity values are used for edge weights.

Unweighted As in the previous case, only k nearest neighbors are included in the adjacency matrix. All edges are weighted equally, regardless of cosine similarity.

We report results using $k = 5$ and $t = 0.05$; other settings are discussed in Appendix B.

Finally, much like the CDE method, we use a second mapping function

$$\psi : \mathbb{R}^{|V|} \rightarrow \mathbb{R}^d$$

to transform the nearest neighbor graph back to d-dimensional vectors for evaluation. Following Newman-Griffis and Fosler-Lussier (2017), we use node2vec (Grover and Leskovec, 2016) with default parameters to learn this mapping. Like the autoencoder, this is a noisy map, but the intent of node2vec to capture patterns in local graph structure makes it a good fit for our analysis.

3.4 Random encoding

Finally, as a baseline, we use a random encoding

$$f_{\text{Rand}} : \mathbb{R}^d \rightarrow \mathbb{R}^d$$

that discards original vectors entirely.

While intrinsic evaluations rely only on input embeddings, and thus lose all source information in this case, extrinsic tasks learn a model to transform input features, making even randomly-initialized vectors a common baseline (Lample et al., 2016; Kim, 2014). For fair comparison, we generate one set of random baselines for each embedding set and re-use these across all tasks.

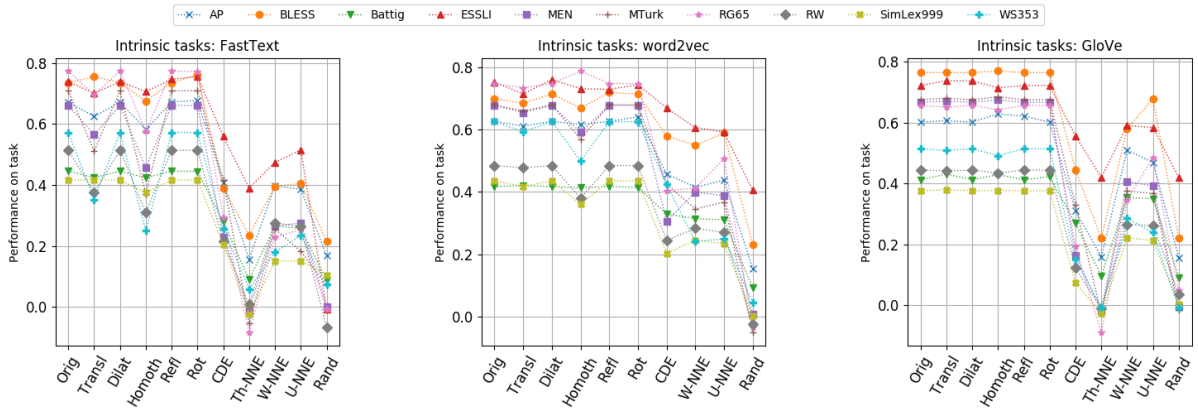
3.5 Other transformations

Many other transformations of a word embedding space could be included in our analysis, such as arbitrary vector-valued polynomial functions, rational vector-valued functions, or common decomposition methods such as principal components analysis (PCA) or singular value decomposition (SVD). Additionally, though they cannot be effectively applied to the unordered set of word vectors in a raw embedding space, transformations for sequential data such as discrete Fourier transforms or discrete wavelet transforms could be used for word sequences in specific text corpora.

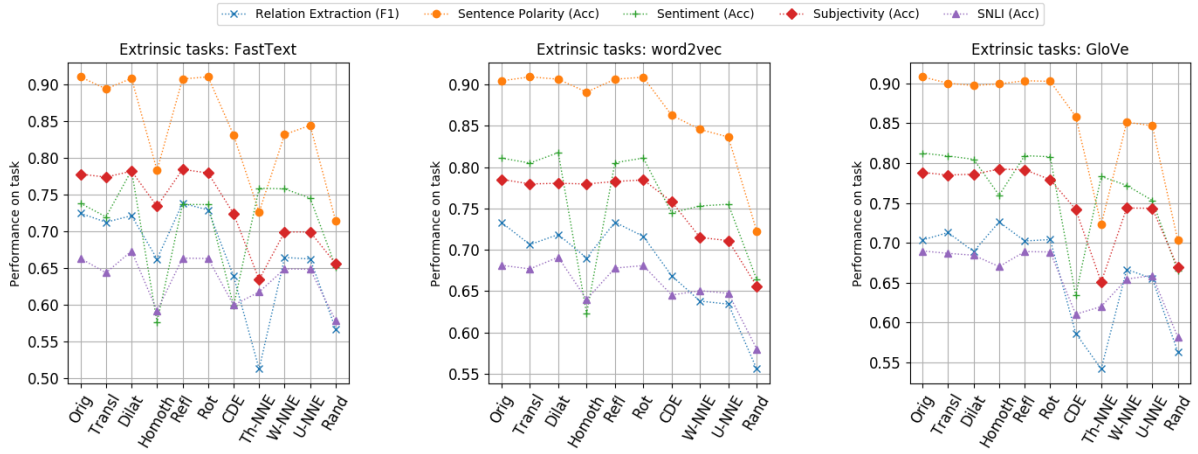
For this study, we limit our scope to the transformations listed above. These transformations align with prior work on analyzing and post-processing embeddings for specific tasks, and are highly interpretable with respect to the original embedding space. However, other complex transformations represent an intriguing area of future work.

4 Evaluation

In order to measure the contributions of each geometric aspect described in Section 3 to the utility of word embeddings as input features, we evaluate embeddings transformed using our sequence of operations on a battery of standard intrinsic evaluations, which model linguistic information directly in the vector space; and extrinsic evaluations, which use the embeddings as input to learned models for downstream applications Our intrinsic evaluations include:



(a) Results of intrinsic evaluations



(b) Results of extrinsic evaluations

Figure 2: Performance metrics on intrinsic and extrinsic tasks, comparing across different transformations applied to each set of word embeddings. Dotted lines are for visual aid in tracking performance on individual tasks, and do not indicate continuous transformations. Transformations are presented in order of decreasing geometric information about the original vectors, and are applied independent of one another to the original source embedding.

- Word similarity and relatedness, using cosine similarity: WordSim-353 (Finkelstein et al., 2001), SimLex-999 (Hill et al., 2015), RareWords (Luong et al., 2013), RG65 (Rubenstein and Goodenough, 1965), MEN (Bruni et al., 2014), and MTURK (Radinsky et al., 2011).[4]
- Word categorization, using an oracle combination of agglomerative and k-means clustering: AP (Almuhareb and Poesio, 2005), BLESS (Baroni and Lenci, 2011), Battig (Battig and Montague, 1969), and the ESSLLI 2008 shared task (Baroni et al. (2008), performance averaged across nouns, verbs, and concrete nouns).[5]

Given the well-documented issues with using vector arithmetic-based analogy completion as an intrinsic evaluation (Linzen, 2016; Rogers et al., 2017; Newman-Griffis et al., 2017), we do not include it in our analysis.

We follow Rogers et al. (2018) in evaluating on a set of five extrinsic tasks:[5]

- Relation classification: SemEval-2010 Task 8 (Hendrickx et al., 2010), using a CNN with word and distance embeddings (Zeng et al., 2014).
- Sentence-level sentiment polarity classification: MR movie reviews (Pang and Lee, 2005), with a simplified CNN model from (Kim, 2014).

[4] `https://github.com/kudkudak/word-embeddings-benchmarks` using single-word datasets only. For brevity, we omit the Sim/Rel splits of WordSim-353 (Agirre et al., 2009), which showed the same trends as the full dataset.

[5] `https://github.com/drgriffis/Extrinsic-Evaluation-tasks`

- Sentiment classification: IMDB movie reviews (Maas et al., 2011), with a single 100-d LSTM.
- Subjectivity/objectivity classification: Rotten Tomato snippets (Pang and Lee, 2004), using a logistic regression over summed word embeddings (Li et al., 2017a).
- Natural language inference: SNLI (Bowman et al., 2015), using separate LSTMs for premise and hypothesis, combined with a feed-forward classifier.

5 Analysis and Discussion

Figure 2 presents the results of each intrinsic and extrinsic evaluation on the transformed versions of our three sets of word embeddings.[6] The largest drops in performance across all three sets for intrinsic tasks occur when explicit embedding features are removed with the CDE transformation. While some cases of NNE-transformed embeddings recover a measure of this performance, they remain far under affine-transformed embeddings. Extrinsic tasks are similarly affected by the CDE transformation; however, NNE-transformed embeddings recover the majority of performance.

Comparing within the set of affine transformations, the innocuous effect of rotations, dilations, and reflections on both intrinsic and extrinsic tasks suggests that the models used are robust to simple linear transformations. Extrinsic evaluations are also relatively insensitive to translations, which can be modeled with bias terms, though the lack of learned models and reliance on cosine similarity for the intrinsic tasks makes them more sensitive to shifts relative to the origin. Interestingly, homothety, which effectively combines a translation and a dilation, leads to a noticeable drop in performance across all tasks. Intuitively, this result makes sense: by both shifting points relative to the origin and changing their distribution in the space, angular similarity values used for intrinsic tasks can be changed significantly, and the zero mean feature distribution preferred by neural models (Clevert et al., 2016) becomes harder to achieve. This suggests that methods for tuning embeddings should attempt to preserve the origin whenever possible.

The large drops in performance observed when using the CDE transformation is likely to relate to the instability of nearest neighborhoods and the importance of locality in embedding learning (Wendlandt et al., 2018), although the effects of the autoencoder component also bear further investigation. By effectively increasing the size of the neighborhood considered, CDE adds additional sources of semantic noise. The similar drops from thresholded-NNE transformations, by the same token, is likely related to observations of the relationship between the frequency ranks of a word and its nearest neighbors (Faruqui et al., 2016). With thresholded-NNE, we find that the words with highest out degree in the nearest neighbor graph are rare words (e.g., "Chanterelle" and "Courtier" in FastText, "Tiegel" and "demangler" in GloVe), which link to other rare words. Thus, node2vec's random walk method is more likely to traverse these dense subgraphs of rare words, adding noise to the output embeddings.

Finally, we note that Melamud et al. (2016) showed significant variability in downstream task performance when using different embedding dimensionalities. While we fixed vector dimensionality for the purposes of this study, varying d in future work represents a valuable follow-up.

Our findings suggest that methods for training and tuning embeddings, especially for downstream tasks, should explicitly focus on local geometric structure in the vector space. One concrete example of this comes from Chen et al. (2018), who demonstrate empirical gains when changing the negative sampling approach of word2vec to choose negative samples that are currently near to the target word in vector space, instead of the original frequency-based sampling (which ignores geometric structure). Similarly, successful methods for tuning word embeddings for specific tasks have often focused on enforcing a specific neighborhood structure (Faruqui et al., 2015). We demonstrate that by doing so, they align qualitative semantic judgments with the primary geometric information that downstream models learn from.

6 Conclusion

Analysis of word embeddings has largely focused on qualitative characteristics such as nearest neighborhoods or relative distribution. In this work, we take a quantitative approach analyzing geometric attributes of embeddings in $\mathbb{R}^d$, in order to understand the impact of geometric properties on downstream task performance. We character-

[6] Due to their large vocabulary size, we were unable to run Thresholded-NNE experiments with word2vec embeddings.

ized word embedding geometry in terms of absolute position, vector features, global pairwise distances, and local pairwise distances, and generated new embedding matrices by removing these attributes from pretrained embeddings. By evaluating the performance of these transformed embeddings on a variety of intrinsic and extrinsic tasks, we find that while intrinsic evaluations are sensitive to absolute position, downstream models rely primarily on information about local similarity.

As embeddings are used for increasingly specialized applications, and as recent contextualized embedding methods such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) allow for dynamic generation of embeddings from specific contexts, our findings suggest that work on tuning and improving these embeddings should focus explicitly on local geometric structure in sampling and evaluation methods. The source code for our transformations and complete tables of our results are available online at `https://github.com/OSU-slatelab/geometric-embedding-properties`.

Acknowledgments

We gratefully acknowledge the use of Ohio Supercomputer Center (Ohio Supercomputer Center, 1987) resources for this work, and thank our anonymous reviewers for their insightful comments. Denis is supported via a Pre-Doctoral Fellowship from the National Institutes of Health, Clinical Center. Aparajita is supported via a Feuer International Scholarship in Artificial Intelligence.

References

Eneko Agirre, Oier Lopez De Lacalle, and Aitor Soroa. 2009. Knowledge-based wsd on specific domains: Performing better than generic supervised WSD. *IJCAI International Joint Conference on Artificial Intelligence*, pages 1501–1506.

A Almuhareb and M Poesio. 2005. Concept Learning and Categorization from the Web. *Proceedings of the Annual Meeting of the Cognitive Science Society*, 27.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2016. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas. Association for Computational Linguistics.

Marco Baroni, Stefan Evert, and Alessandro Lenci, editors. 2008. *Proceedings of the ESSLLI Workshop on Distributional Lexical Semantics: Bridging the gap between semantic theory and computational simulations*. Hamburg, Germany.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. *GEMS '11 Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10.

William F Battig and William E Montague. 1969. Category norms of verbal items in 56 categories A replication and extension of the Connecticut category norms. *Journal of Experimental Psychology*, 80(3, Pt.2):1–46.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the ACL*, 5:135–146.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal Distributional Semantics. *Journal of Artificial Intelligence Research*, 49(1):1–47.

Chandrahas, Aditya Sharma, and Partha Talukdar. 2018. Towards Understanding the Geometry of Knowledge Graph Embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 122–131. Association for Computational Linguistics.

Long Chen, Fajie Yuan, Joemon M. Jose, and Weinan Zhang. 2018. Improving negative sampling for word representation using self-embedded features. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 99–107, New York, NY, USA. ACM.

Billy Chiu, Anna Korhonen, and Sampo Pyysalo. 2016. Intrinsic Evaluation of Word Vectors Fails to Predict Extrinsic Performance. *Proceedings of the 1st Workshop on Evaluating Vector Space Representations for NLP*, pages 1–6.

Djork-Arne Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2016. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Amaru Cuba Gyllensten and Magnus Sahlgren. 2015. Navigating the Semantic Horizon using Relative Neighborhood Graphs. In *Proceedings of the 2015*

Conference on Empirical Methods in Natural Language Processing, pages 2451–2460. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *arXiv preprint arXiv:1810.04805v1*.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems With Evaluation of Word Embeddings Using Word Similarity Tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35, Berlin, Germany. Association for Computational Linguistics.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A Smith. 2015. Sparse Overcomplete Word Vector Representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1491–1500, Beijing, China. Association for Computational Linguistics.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing Search in Context: The Concept Revisited. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 406–414, Hong Kong. ACM.

Alona Fyshe, Leila Wehbe, Partha P Talukdar, Brian Murphy, and Tom M Mitchell. 2015. A Compositional and Interpretable Semantic Space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 32–41, Denver, Colorado. Association for Computational Linguistics.

Aditya Grover and Jure Leskovec. 2016. Node2Vec: Scalable Feature Learning for Networks. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 855–864, New York, NY, USA. ACM.

F Heimerl and M Gleicher. 2018. Interactive Analysis of Word Vector Embeddings. *Computer Graphics Forum*, 37(3):253–265.

Johannes Hellrich and Udo Hahn. 2016. Bad Company—Neighborhoods in Neural Embedding Spaces Considered Harmful. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2785–2796. The COLING 2016 Organizing Committee.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid 'O S 'eaghdha, Sebastian Pad 'o, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38. Association for Computational Linguistics.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating Semantic Models with (Genuine) Similarity Estimation. *Computational Linguistics*, 41(4):665–695.

Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. 2008. Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171–1220.

Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Bofang Li, Tao Liu, Zhe Zhao, Buzhou Tang, Aleksandr Drozd, Anna Rogers, and Xiaoyong Du. 2017a. Investigating Different Syntactic Context Types and Context Representations for Learning Word Embeddings. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2421–2431. Association for Computational Linguistics.

Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. 2017b. Grass: Generative recursive autoencoders for shape structures. *ACM Trans. Graph.*, 36(4):52:1–52:14.

Tal Linzen. 2016. Issues in evaluating semantic spaces using word analogies. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 13–18.

Thang Luong, Richard Socher, and Christopher Manning. 2013. Better Word Representations with Recursive Neural Networks for Morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, Sofia, Bulgaria. Association for Computational Linguistics.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150. Association for Computational Linguistics.

Oren Melamud, David McClosky, Siddharth Patwardhan, and Mohit Bansal. 2016. The Role of Context Types and Dimensionality in Learning Word Embeddings. In *Proceedings of the 2016 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT '16, pages 1030–1040, San Diego, CA. Association for Computational Linguistics.

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

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

Denis Newman-Griffis and Eric Fosler-Lussier. 2017. Second-order word embeddings from nearest neighbor topological features. *arXiv preprint arXiv:1705.08488*.

Denis Newman-Griffis, Albert M Lai, and Eric Fosler-Lussier. 2017. Insights into Analogy Completion from the Biomedical Domain. In *BioNLP 2017*, pages 19–28, Vancouver, Canada. Association for Computational Linguistics.

Ohio Supercomputer Center. 1987. Ohio supercomputer center. `http://osc.edu/ark:/19495/f5s1ph73`.

Bo Pang and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*.

Bo Pang and Lillian Lee. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Benedicte Pierrejean and Ludovic Tanguy. 2018. Towards Qualitative Word Embeddings Evaluation: Measuring Neighbors Variation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 32–39. Association for Computational Linguistics.

Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. 2011. A Word at a Time: Computing Word Relatedness Using Temporal Semantic Analysis. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 337–346, New York, NY, USA. ACM.

Anna Rogers, Aleksandr Drozd, and Bofang Li. 2017. The (too Many) Problems of Analogical Reasoning with Word Vectors. *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 135–148.

Anna Rogers, Shashwath Hosur Ananthakrishna, and Anna Rumshisky. 2018. What's in Your Embedding, And How It Predicts Task Performance. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2690–2703, Santa Fe, NM, USA. Association for Computational Linguistics.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *International Conference on Learning Representations*.

Laura Wendlandt, Jonathan K Kummerfeld, and Rada Mihalcea. 2018. Factors Influencing the Surprising Instability of Word Embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2092–2102. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation Classification via Convolutional Deep Neural Network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344. Dublin City University and Association for Computational Linguistics.

Appendix A Parameters

We give the following library of vectors in $\mathbb{R}^d$ used as parameter values:

$$\mathbf{v}_{\text{diag}} = \begin{bmatrix} \frac{1}{\sqrt{d}} \\ \vdots \\ \frac{1}{\sqrt{d}} \end{bmatrix}; \quad \mathbf{v}_{\text{diagNeg}} = \begin{bmatrix} -\frac{1}{\sqrt{d}} \\ \frac{1}{\sqrt{d}} \\ \vdots \\ \frac{1}{\sqrt{d}} \end{bmatrix}. \tag{A.1}$$

Transform	**Parameter**	**Value**
Translation	Direction:	$\mathbf{0}$
	Magnitude:	1
Dilation	Magnitude:	2
Homothety	Center:	$\mathbf{v}_{\text{diag}}$
	Magnitude:	0.25
Reflection	Hyperplane Vector:	$\mathbf{v}_{\text{diag}}$
2-D Rotation	Basis Vector 1:	$\mathbf{v}_{\text{diag}}$
	Basis Vector 2:	$\mathbf{v}_{\text{diagNeg}}$
	Angle:	$\pi/4$

Table 1: Transform parameters.

Appendix B NNE settings

We experimented with $k \in \{5, 10, 15\}$ for our weighted and unweighted NNE transformations. For thresholded NNE, in order to best evaluate the impact of thresholding over uniform k, we used the minimum $k = 5$ and experimented with $t \in \{0.01, 0.05, 0.075\}$; higher values of t increased graph size sufficiently to be impractical. We report using $k = 5$ for weighted and unweighted settings in our main results for fairer comparison with the thresholded setting.

The effect of thresholding on nearest neighbor graphs was a strongly right-tailed increase in out degree for a small portion of nodes. Our reported value of $t = 0.05$ increased the out degree of 20,229 nodes for FastText (out of 1M total nodes), with the maximum increase being 819 ("Chanterelle"), and 1,354 nodes increasing out degree by only 1. For GloVe, 7,533 nodes increased in out degree (out of 2M total), with maximum increase 240 ("Tiegel"), and 372 nodes increasing out degree by only 1.

Table 2 compares averaged performance values across all intrinsic tasks for these settings, and Table 3 compares average extrinsic task performance.

NNE params	FastText	word2vec	GloVe
Thresholded			
$k = 5, t = 0.01$	0.160	–	0.106
$k = 5, t = 0.05$	0.129	–	0.130
$k = 5, t = 0.075$	0.150	–	0.132
Weighted			
$k = 5$	0.320	0.419	0.426
$k = 10$	0.342	0.363	0.460
$k = 15$	0.346	0.376	0.448
Unweighted			
$k = 5$	0.330	0.428	0.435
$k = 10$	0.351	0.396	0.463
$k = 15$	0.341	0.365	0.432

Table 2: Mean performance on intrinsic tasks under different NNE settings.

NNE params	FastText	word2vec	GloVe
Thresholded			
$k = 5, t = 0.01$	0.642	–	0.666
$k = 5, t = 0.05$	0.650	–	0.664
$k = 5, t = 0.075$	0.649	–	0.663
Weighted			
$k = 5$	0.721	0.720	0.738
$k = 10$	0.728	0.713	0.740
$k = 15$	0.725	0.713	0.739
Unweighted			
$k = 5$	0.720	0.717	0.732
$k = 10$	0.724	0.712	0.738
$k = 15$	0.729	0.708	0.725

Table 3: Mean performance on extrinsic tasks under different NNE settings.