Contrastive Loss is All You Need to Recover Analogies as Parallel Lines

Narutatsu Ri
Columbia University
wl2787@columbia.edu

Fei-Tzin Lee
Columbia University
feitzin@cs.columbia.edu

Nakul Verma
Columbia University
verma@cs.columbia.edu

Abstract

While static word embedding models are known to represent linguistic analogies as parallel lines in high-dimensional space, the underlying mechanism as to why they result in such geometric structures remains obscure. We find that an elementary contrastive-style optimization employed over distributional information performs competitively with popular word embedding models on analogy recovery tasks, while achieving dramatic speedups in training time. Further, we demonstrate that a contrastive loss is sufficient to create these parallel structures in word embeddings, and establish a precise relationship between the co-occurrence statistics and the geometric structure of the resulting word embeddings.

1 Introduction

Static word embeddings take inspiration from the distributional hypothesis (Firth, 1957) and assign vector representations to words based on co-occurrence statistics. Such embeddings are known to implicitly encode syntactic and semantic analogies as parallelogram-type structures (Mikolov et al., 2013a,b). This discovery inspired a series of theoretical investigations (Levy and Goldberg, 2014; Gittens et al., 2017; Allen and Hospedales, 2019; Ethayarajh et al., 2019).

Recent studies reconsider whether analogies are indeed represented as parallelograms in the embedding space (Schluter, 2018; Linzen, 2016; Fournier and Dunbar, 2021), and propose a weaker notion of viewing analogies as parallel *lines* (Arora et al., 2016) as a more appropriate model (cf. Figure 1). While this claim is shown to hold empirically for popular word embeddings (Fournier et al., 2020), few analyze the theoretical underpinnings of this phenomenon.

In this paper, we present a remarkable observation that a simple contrastive-style optimization (Chopra et al., 2005) performs just as well as highly-optimized versions of popular word embeddings while achieving 50× speedup in training time. Our work theoretically analyzes the precise conditions under which this optimization procedure can recover analogies as parallel lines. We further investigate the extent to which real-world data satisfies these conditions, and the contrastive loss recovers such parallel structures.

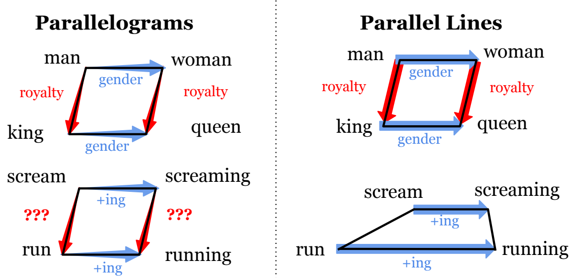


Figure 1: Visualization of analogies as parallelograms and as parallel lines. For the quadruple "`man, woman, king, queen`", two analogy relations coincide ("`man:woman = king:queen`" representing gender and "`man:king = woman:queen`" representing royalty). In contrast, the quadruple "`run, running, scream screaming`" contains only one analogy relation ("`run:running = scream:screaming`" representing present participle). Representing analogies as lines relaxes the geometric requirements on the analogy structure.

In Section 2, we review recent literature on the theory of word embeddings. Sections 3 and 4 present our contrastive learning objective and its analysis. Section 5 showcases the performance of our approach on analogy-based benchmarks.[1]

2 Related Work

Analogies as Parallelograms. Gittens et al. (2017) study the parallelogram phenomenon by analyzing analogies as a relation between paraphrases. Allen and Hospedales (2019) extend this line of work and show that analogies are captured as parallelo-

[1] Code can be found at https://github.com/narutatsuri/cwm.

Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023), pages 164–173
July 13, 2023 ©2023 Association for Computational Linguistics

grams when the vectors are linear projections of the *Pointwise Mutual Informaton* (PMI) matrix. Ethayarajh et al. (2019) further generalize Gittens' result by introducing the *co-occurrence shifted pointwise mutual information* (csPMI)[2] and analyze the conditions on the csPMI for which parallelograms emerge.

Analogies as Parallel Lines. To the best of our knowledge, the only theoretical work that explores analogies more generally as parallel lines is by Arora et al. (2019), who propose that analogies are encoded as such when the inner products between embeddings weakly recover the PMI of word co-occurrence statistics. We take an alternate approach and show that a contrastive-style optimization suffices to encode analogies as parallel lines.

3 The Contrastive Word Model (CWM)

Contrastive learning methods are based on an intuitive yet powerful idea that pulling similar items closer together while pushing dissimilar items away significantly improves model performance.

We can employ the same push-pull dynamics in word embeddings by placing the vector representations of words that co-occur closer together than those of words that do not. We call this the Contrastive Word Model (CWM), detailed below.

3.1 Notation & Formulation

Given a training corpus, we denote the vocabulary as W. We aim to learn a D-dimensional vector representation v_w for each word w in the vocabulary. The collection of all these vectors is denoted by $V = \{v_1, \ldots, v_{|W|}\}$. We refer to the length-normalized version of a vector v as $\hat{v}$.

Let $\#(i)$ be the occurrence count of word i and $\#(i, j)$ the co-occurrence count (for a context window of size Δ) of words i and j in the training corpus. We denote *window words* as words that co-occur with a reference *center word* (these are reminiscent of the target and context words in Mikolov et al., 2013b), and *negative window words* as words that do not co-occur with the center word. The center-, window-, and negative window words are denoted as c, w, w' respectively. Let $D_{c,w}$ be the set of negative window words for fixed c, w. We define the CWM objective as:

$$\sum_{c\in W}\sum_{w\in W} \#(c,w) \cdot \sum_{w'\in D_{c,w}} \Big[m - \underbrace{\hat{v}_c \cdot \hat{v}_w}_{\text{pull}} + \underbrace{\hat{v}_c \cdot \hat{v}_{w'}}_{\text{push}}\Big]_+ ,$$

where $[\cdot]_+$ is the hinge function and m is a tunable hyperparameter.

To better understand our proposed loss, consider its effect on a fixed center word c. The difference between the terms $\hat{v}_c \cdot \hat{v}_w$ and $\hat{v}_c \cdot \hat{v}_{w'}$ encourages the *angle* between vectors v_c and v_w to be smaller than that between v_c and $v_{w'}$ by at least a margin of m. The hinge function neutralizes the loss once the vectors satisfy the desired relationship. Such max-margin type losses among triples are well investigated in metric learning literature (Weinberger et al., 2005).

3.2 Relation to Popular Word Embeddings

Interestingly, popular word embedding models such as Skip-gram (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014) can be viewed as implicitly employing a push-pull action similar to CWM. Consider Skip-gram's objective: for a fixed pair of co-occurring words c and w, the model updates the word vector v_c as:

$$\begin{aligned} v_c^{\text{new}} =& v_c^{\text{old}} + \underbrace{\left(1 - \frac{e^{v_w^\intercal u_{c'}}}{\sum_{w'\in W} e^{v_w^\intercal u_{w'}}}\right) v_w}_{\text{pull}} \\ & - \underbrace{\mathbb{E}_{w'\sim W}[v_{w'}]}_{\text{push}} + \text{additional terms.} \end{aligned} \quad (1)$$

Here, the word c' (and its target vector $u_{c'}$) co-occurs with both c and w, encouraging all of them to be mapped together (pull), whereas the negative term pushes away randomly sampled words w' from c. See Appendix A.3 for a derivation.

The GloVe objective, on the other hand, performs a series of updates on c, w, and w' as:

$$\begin{aligned} &\text{pull} \begin{cases} v_c^{\text{new}} &= v_c^{\text{old}} + g(c,c')u_{c'} \\ v_w^{\text{new}} &= v_w^{\text{old}} + g(w,c')u_{c'} \end{cases} \\ &\text{push} \left\{ v_{w'}^{\text{new}} = v_{w'}^{\text{old}} - g(w',c')u_{c'}, \right. \end{aligned} \quad (2)$$

where $g(\cdot,\cdot)$ always returns a positive value. Notice that the *positive* contribution of g in the first two updates encourages v_c and v_w to be closer together (pull), while the *negative* contribution to the $v_{w'}$ update encourages it to be pushed away. See Appendix A.4 for a derivation.

[2] $\text{csPMI}(a,b) = \text{PMI}(a,b) + \log p(a,b)$.

We believe that part of the success of these word embedding models is due to their implicit push-pull dynamics. Hence, a natural question to consider is what happens when one purely optimizes for the push-pull action alone.

4 Analysis

In this section, we provide a theoretical justification for the emergence of analogies as parallel lines when we optimize for the CWM objective.

Consider the expression for word vectors $v_c \in V$ that minimizes the global objective:

$$v_c = \rho_c \left(\sum_{w \in W} \left(\frac{\#(c,w)}{\#(c)} \hat{v}_w \right) - \mathop{\mathbb{E}}_{w' \sim U(W)} [\hat{v}_{w'}] \right), \quad (3)$$

where $\rho_c \in \mathbb{R}$ is a constant dependent on c. In essence, v_c is the difference between the weighted average of the window words and the mean of all word vectors. See Appendix A.1 for derivation.

Under Eq. (3), we consider the conditions that word co-occurrence statistics need to satisfy for a set of words a, b, c, d to form parallel geometric structures.

Theorem 1 *For any quadruple of words $a, b, c, d \in W$, if there exists a constant $\zeta \in \mathbb{R}$ where the co-occurrence statistics satisfy the condition:* $\forall w \in W$

$$\left(\tfrac{\#(a,w)}{\#(a)} - \tfrac{\#(b,w)}{\#(b)} \right) \Big/ \left(\tfrac{\#(c,w)}{\#(c)} - \tfrac{\#(d,w)}{\#(d)} \right) := \zeta, \quad (4)$$

then the corresponding word vectors satisfy the property:

$$\hat{v}_a - \hat{v}_b = \zeta \left(\hat{v}_c - \hat{v}_d \right).$$

Note that Theorem 1 establishes a direct relationship between word co-occurrence statistics—which are solely derived from the training corpus—and the geometric structure of the word embedding.

For a given quadruple $a, b, c, d \in W$ (regardless of whether they form an analogy), the existence of ζ induces parallel structures between $\hat{v}_a, \hat{v}_b, \hat{v}_c, \hat{v}_d$. If such a ζ exists and is equal to 1, then $\hat{v}_b - \hat{v}_a = \hat{v}_d - \hat{v}_c$ and the quadruple forms a parallelogram. When $\zeta \neq 1$, then the difference vectors $\hat{v}_b - \hat{v}_a$ and $\hat{v}_d - \hat{v}_c$ are mainly parallel, inducing a trapezoidal structure among $\hat{v}_a, \hat{v}_b, \hat{v}_c, \hat{v}_d$ (cf. Figure 1).

One would expect that the co-occurrence statistics of real data conform with the existence of such a ζ value for analogy quadruples, whereas ζ does not exist for random quadruples. This is empirically investigated in Section 5.3. The relationship between the value of ζ and the resulting parallelogram structure (parallelogram vs. trapezoid) is empirically verified in Section 5.4.

	Analogies		Training	
Model	**PCS**	**MSM**	**Time (hrs)**	**Speedup**
CWM	**0.677**	**0.469**	**0.59**	**49×**
SGNS	0.675	0.433	29.27	1×
GloVe	0.667	0.423	30.71	0.91×

Table 1: Performances for word embedding models. CWM refers to our contrastive word model. SGNS refers to Skip-gram with negative sampling. Best numbers are bolded.

5 Experiments

We first compare the performance of CWM to that of other popular word embedding methods on analogy recovery (Section 5.2). We then empirically verify the degree to which our assumptions regarding co-occurrences hold on real data (Section 5.3) as well as the relation between ζ and the parallelogram structure (Section 5.4).

5.1 Data and Training Procedure

We use the 03/2023 version of Wikimedia Downloads dump (Foundation, 2023) and train CWM for a single pass over the corpus using $\Delta = 5$ and $m = 0.2$ (chosen via cross validation from the range $0.1 \sim 1$). For comparison, we also train Skip-gram with Negative Sampling (SGNS) and GloVe over the same corpus with the default parameter settings provided by Mikolov et al. (2013b) and Pennington et al. (2014) respectively.

We utilize the BATS analogy dataset (Gladkova et al., 2016) for all analogy related tasks. For all word embeddings, we use dimension $D = 300$ and the vectors are length-normalized to follow practical conventions (Mikolov et al., 2013b). Training was done on 256 instances of AMD EPYC 7763 64-Core Processor machine.

5.2 Analogy Recovery

To assess the degree to which word embeddings encode analogies as lines consistently in the embedding space, we use two intuitive metrics proposed by Fournier et al. (2020): the *Pairing Consistency Score* (PCS) and *Mean Similarity Measure* (MSM). PCS assesses analogy alignment precision (the number of non-analogy offsets incorrectly aligned

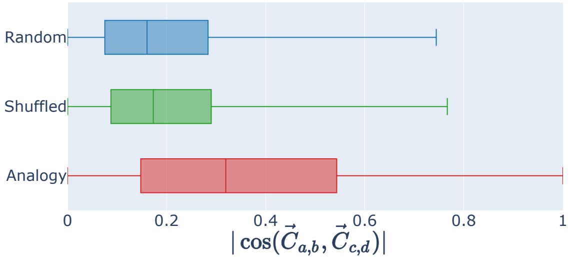


Figure 2: Cosine similarities between co-occurrence vectors $\vec{C}_{a,b}$ and $\vec{C}_{c,d}$ for words a, b, c, d from uniformly sampled word quadruples (Random), shuffled analogy pairs (Shuffled), and true analogy pairs (Analogy).

Analogy Quadruple	Sim.
`fall:rise = under:over`	1.000
`prevent:preventing = follow:following`	0.9901
`lancaster:lancashire = salford:manchester`	0.9812
`refer:referred = agree:agreed`	0.9740
`organized:arranged = dollars:bucks`	0.0006
`staircase:step = shilling:pence`	0.0006
`guitar:string = church:altar`	0.0004
`monkey:infant = fox:cub`	0.0001

Table 2: Samples of analogy quadruples illustrating cosine similarity values between $\vec{C}_{a,b}$ and $\vec{C}_{c,d}$. "Sim." denotes the value of $|\cos(\vec{C}_{a,b}, \vec{C}_{c,d})|$.

with true analogy offsets), while MSM measures absolute alignment.

Table 1 shows the relative performance of popular word embeddings. Notice that our method performs 7% better than popular word embeddings on the MSM metric, indicating that the word vectors learned by CWM exhibit higher alignment among analogy quadruples than Skip-gram and GloVe. CWM's performance on the PCS metric indicates that parallel lines are not erroneously encoded for non-analogy words. For completeness, see Appendix B.3 for parallelogram recovery performances (previous literature questions the validity of the standard evaluation method).

5.3 Existence of ζ and Analogies

Theorem 1 provides insight into the conditions required for CWM to induce parallel lines in the learned word vectors, but these conditions are not specific to analogy word pairs. Thus, the question remains: does ζ exist only when a quadruple forms an analogy?

Here, we study the level at which the co-occurrence statistics of analogy and non-analogy pairs satisfy the condition in Theorem 1. To assess the existence of ζ, consider the vectors $\vec{C}_{a,b}, \vec{C}_{c,d} \in \mathbb{R}^{|W|}$ (derived purely from co-occurrence counts):

$$\vec{C}_{a,b} = \left[\left(\frac{\#(a,w_1)}{\#(a)} - \frac{\#(b,w_1)}{\#(b)}\right), \dots, \left(\frac{\#(a,w_{|W|})}{\#(a)} - \frac{\#(b,w_{|W|})}{\#(b)}\right)\right],$$

$$\vec{C}_{c,d} = \left[\left(\frac{\#(c,w_1)}{\#(c)} - \frac{\#(d,w_1)}{\#(d)}\right), \dots, \left(\frac{\#(c,w_{|W|})}{\#(c)} - \frac{\#(d,w_{|W|})}{\#(d)}\right)\right].$$

Existence of a ζ where Eq. (4) holds for a, b, c, d, implies that all entries in $\vec{C}_{a,b}$ are equal to the corresponding entries in $\vec{C}_{c,d}$ scaled by a factor of ζ. This indicates that when ζ exists, $\vec{C}_{a,b}$ and $\vec{C}_{c,d}$ are collinear. Thus, we can approximate assessing the existence of ζ by evaluating whether the cosine similarity between $\vec{C}_{a,b}$ and $\vec{C}_{c,d}$ is sufficiently high.

We consider three settings from which the quadruples are obtained: randomly sampled word quadruples, false shuffled analogies, and true analogies using the BATS dataset. We compute the distribution of cosine similarities for all quadruples from these settings.

Results are shown in Figure 2. Observe that the cosine similarities of random and shuffled quadruples is significantly lower than that for analogy words. This indicates a positive association between ζ and analogy word quadruples in real world corpora.

Furthermore, it is worth noting the presence of "ambiguous" analogies within the BATS dataset. These include analogies with valid alternative replacements (e.g. `sun:orange = sea:blue` can also be `sun:red = sea:blue`), or analogies with unclear relationships (e.g. lexicographic analogies such as `father:dad = lady:madam`). We investigate whether the ambiguity of an analogy correlates with its cosine similarity between $\vec{C}_{a,b}$ and $\vec{C}_{c,d}$ by sampling from analogy quadruples with high and low values of $|\cos(\vec{C}_{a,b}, \vec{C}_{c,d})|$.

Results are shown in Table 2. Observe that analogy quadruples with high cosine similarity between $\vec{C}_{a,b}$ and $\vec{C}_{c,d}$ seems to demonstrate a clear relationships, whereas those with low cosine similarity exhibit weaker/ambiguous relationships.

5.4 ζ and Geometric Structure

We now examine the effect of ζ on the geometry of analogy word pairs. Recall that ζ exists for quadruples where $|\cos(\vec{C}_{a,b}, \vec{C}_{c,d})| = 1$. As this condition is unlikely to hold exactly on real data, we approximate ζ with the ratio $\hat{\zeta} := \|\vec{C}_{a,b}\|/\|\vec{C}_{c,d}\|$ for quadruples with high cosine similarity (which we define as $|\cos(\vec{C}_{a,b}, \vec{C}_{c,d})| \geq 0.9$). We expect the word vectors to form parallelograms when $\hat{\zeta} \approx 1$

	$k = 1$	$k = 5$
$\hat{\zeta} \not\approx 1$	0.800 (619/774)	0.862 (667/774)
$\hat{\zeta} \approx 1$	0.652 (137/210)	0.871 (183/210)

Table 3: Parallelogram/trapezoid recovery performances for different values of $\hat{\zeta}$. Parallelogram recovery for all analogy pairs is 0.27 (see Table 4 in Appendix), indicating dramatic performance increase for the analogy subset where $\hat{\zeta} \approx 1$.

($0.95 \leq \hat{\zeta} \leq 1.05$), and form trapezoids otherwise.

Specifically, for each such quadruple, we compute the word w that minimizes $\|\hat{v}_b - \hat{v}_a + \hat{v}_c - \hat{v}_w\|$ for parallelograms; ideally, w should equal d. For trapezoids, we retrieve the word w that maximizes the quantity $\cos(\hat{v}_b - \hat{v}_a, \hat{v}_w - \hat{v}_c)$. If the word d is among the top k words, we deem the quadruple to satisfy the corresponding geometric structure. For both cases, we consider $k = 1$ and 5.

Results are shown in Table 3. Observe that for $k = 5$, 87% of the quadruples form parallelograms when $\hat{\zeta} \approx 1$ (i.e., $0.95 \leq \hat{\zeta} \leq 1.05$), and 86% of quadruples form trapezoid-type structures when $\hat{\zeta} \not\approx 1$. This validates our expectation that parallelograms and trapezoids indeed form when $\hat{\zeta} \approx 1$ and $\hat{\zeta} \not\approx 1$ respectively.

6 Conclusion and Discussion

We demonstrate that optimizing a contrastive-style objective over word co-occurrences is indeed sufficient to encode analogies as parallel lines. Our analysis (Theorem 1) sheds light on the inner workings of word embeddings: parallel geometry is induced largely from word co-occurrence statistics for any push-pull model. Our work builds upon and generalizes previous literature that illuminates the underlying mechanisms governing the geometry of word embeddings.

Note that while our results demonstrate the sufficiency of the push-pull mechanism for recovering analogies as parallel lines, it remains unclear whether push-pull is a necessary condition for this phenomenon. Investigating alternative mechanisms and their ability to achieve similar results would provide further insight into the relationship between word co-occurrence statistics and the recovery of analogies.

References

Carl Allen and Timothy Hospedales. 2019. Analogies explained: Towards understanding word embeddings.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2019. A latent variable model approach to pmi-based word embeddings.

Elia Bruni, Nam Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Intell. Res.*, 49:1–47.

S. Chopra, R. Hadsell, and Y. LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1.

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. 2019. Towards understanding linear word analogies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3253–3262, Florence, Italy. Association for Computational Linguistics.

Lev Finkelstein, Evgeniy Gabrilovich, Y. Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2002. Placing search in context: the concept revisited. *ACM Trans. Inf. Syst.*, 20:116–131.

J. Firth. 1957. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*. Philological Society, Oxford. Reprinted in Palmer, F. (ed. 1968) Selected Papers of J. R. Firth, Longman, Harlow.

Wikimedia Foundation. 2023. Wikimedia downloads.

Louis Fournier and Ewan Dunbar. 2021. Paraphrases do not explain word analogies.

Louis Fournier, Emmanuel Dupoux, and Ewan Dunbar. 2020. Analogies minus analogy test: measuring regularities in word embeddings. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 365–375, Online. Association for Computational Linguistics.

Alex Gittens, Dimitris Achlioptas, and Michael W. Mahoney. 2017. Skip-gram - Zipf + uniform = vector additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 69–76, Vancouver, Canada. Association for Computational Linguistics.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15, San Diego, California. 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.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

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, Berlin, Germany. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

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.

Natalie Schluter. 2018. The word analogy testing caveat. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 242–246, New Orleans, Louisiana. Association for Computational Linguistics.

Kilian Q Weinberger, John Blitzer, and Lawrence Saul. 2005. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems*, volume 18. MIT Press.

A Proofs

A.1 Derivation of Eq. (3)

Recall that the global objective of CWM for the vocabulary W and set of word vectors V can be written as:

$$\mathcal{L}(V) = \sum_{c\in W}\sum_{w\in W}\#(c,w)$$
$$\sum_{w'\in D_{c,w}}\left[m - \hat{v}_c\cdot\hat{v}_w + \hat{v}_c\cdot\hat{v}_{w'}\right]_+,$$

where $D_{c,w} = \{w'|w' \sim U(W)\}, |D_{c,w}| = k$ denotes the set of k negative window words sampled uniformly from the vocabulary for each c, w word pair and $U(W)$ denotes the uniform distribution over the vocabulary.

For fixed c, w, w', consider the two cases where $m-\hat{v}_c\cdot\hat{v}_w+\hat{v}_c\cdot\hat{v}_{w'} > 0$ and $m-\hat{v}_c\cdot\hat{v}_w+\hat{v}_c\cdot\hat{v}_{w'} \le 0$. As the word vectors are not updated for the latter case, we examine the former by taking the partial derivative of $\mathcal{L}(V)$ with respect to v_c and setting it to 0:

$$0 = -\sum_{w\in W}\#(c,w)$$
$$\sum_{w'\in D_{c,w}}\left(\frac{v_w}{\|v_c\|\|v_w\|} - \frac{v_{w'}}{\|v_c\|\|v_{w'}\|}\right.$$
$$\left.+\left(\frac{\hat{v}_c\cdot\hat{v}_{w'}}{\|v_c\|^2} - \frac{(\hat{v}_c\cdot\hat{v}_w)}{\|v_c\|^2}\right)v_c\right)$$
$$\Leftrightarrow \sum_{w\in W}\#(c,w)\sum_{w'\in D_{c,w}}\frac{v_w}{\|v_c\|\|v_w\|}$$
$$-\sum_{w\in W}\#(c,w)\sum_{w'\in D_{c,w}}\frac{v_{w'}}{\|v_c\|\|v_{w'}\|}$$
$$=\sum_{w\in W}\#(c,w)\sum_{w'\in D_{c,w}}\left(\frac{v_c v_w}{\|v_c\|^2\|v_w\|}\right.$$
$$\left.-\frac{v_c v_{w'}}{\|v_c\|^2\|v_{w'}\|}\right)\frac{v_c}{\|v_c\|}.$$

As $\sum_{w\in W}\#(c,w)\sum_{w'\in D_{c,w}}\frac{v_{w'}}{\|v_{w'}\|}$ represents $\sum_{w\in W}\#(c,w)\cdot k = k\cdot\#(c)$ uniform i.i.d. draws from the vocabulary, the following holds for sufficiently large values of $k\cdot\#(c)$:

$$\sum_{w\in W}\#(c,w)\sum_{w'\in D_{c,w}}\frac{v_{w'}}{\|v_{w'}\|} =$$
$$k\#(c)\mathbb{E}_{w'\sim U(W)}\left[\frac{v_{w'}}{\|v_{w'}\|}\right].$$

Setting $\mathbb{E}_{w'\sim U(W)}\left[\frac{v_{w'}}{\|v_{w'}\|}\right] = v_p$ and dividing both sizes by $\frac{k\#(c)}{\|v\|}$,

$$\sum_{w\in W}\frac{\#(c,w)}{\#(c)}\frac{v_w}{\|v_w\|} - v_p$$
$$=\left[\frac{v_c}{\|v_c\|}\left(\sum_{w\in W}\frac{\#(c,w)}{\#(c)}\frac{v_w}{\|v_w\|} - v_p\right)\right]\odot\frac{v_c}{\|v_c\|}.$$

Setting $\sum_{w\in W}\frac{\#(c,w)}{\#(c)}\frac{v_w}{\|v_w\|} = v_{p'}$ and $\gamma_c = \left\|\frac{v_c}{\|v_c\|}\left(\sum_{w\in W}\frac{\#(c,w)}{\#(c)}\frac{v_w}{\|v_w\|} - v_p\right)\right\|$, the above equation can be rewritten as:

$$\frac{v_c}{\|v_c\|} = \frac{v_{p'}}{\gamma_c}\cdot\frac{1}{\left\|\frac{v_c}{\|v_c\|}\right\|}\cdot\frac{1}{\cos\theta}.$$

where θ indicates the angle between $v_{p'}$ and $\frac{v_c}{\|v_c\|}$.

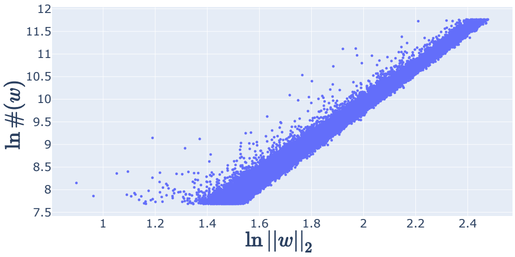
As $\left\|\frac{v_c}{\|v_c\|}\right\| = 1$,

$$v_c = \|v_c\|\cdot\frac{v_{p'}}{\gamma_c}\cdot\frac{1}{\cos\theta}$$

Notice that $v_c \parallel v_{p'}$ by the above construction , so $\cos\theta = 1$. Thus,

$$v_c = \|v_c\|\cdot\frac{v_{p'}}{\gamma_c} = \frac{\alpha\#(c)^{\frac{1}{\beta}}}{\gamma_c}\cdot v_{p'}. \quad \blacksquare$$

The second equality is derived from the empirically observed property $\|v_c\| \propto \#(c)^{\frac{1}{\beta}}$ for some constant $\beta\in\mathbb{R}$, which is verified below.



Interestingly, a similar linear relationship is also observed in existing word embedding models (Arora et al., 2016).

A.2 Proof for Theorem 1

Under the assumption that Eq. (3) holds, we can write the expressions for $\hat{v}_a - \hat{v}_b, \hat{v}_c - \hat{v}_d$ as follows.

$$\hat{v}_a - \hat{v}_b = \frac{1}{\gamma_a}\left(\sum_{w\in W}\left(\frac{\#(a,w)}{\#(a)}\frac{v_w}{\|v_w\|}\right) - v_p\right)$$
$$-\frac{1}{\gamma_b}\left(\sum_{w\in W}\left(\frac{\#(b,w)}{\#(b)}\frac{v_w}{\|v_w\|}\right) - v_p\right).$$

Under the assumption that $\forall c \in W : \gamma_c = \gamma$ for some $\gamma \in \mathbb{R}$,

$$\hat{v}_a - \hat{v}_b = \frac{1}{\gamma} \sum_{w \in W} \left(\frac{\#(a,w)}{\#(a)} - \frac{\#(b,w)}{\#(b)} \right) \frac{v_w}{\|v_w\|}$$

Using Eq. (4),

$$\hat{v}_a - \hat{v}_b = \frac{\zeta}{\gamma} \sum_{w \in W} \left(\frac{\#(c,w)}{\#(c)} - \frac{\#(d,w)}{\#(d)} \right) \frac{v_w}{\|v_w\|}$$
$$= \zeta(\hat{v}_c - \hat{v}_d). \quad \blacksquare$$

The invariance of the value γ_c can be verified through randomly sampling 5000 words c and computing the respectivve γ_c. The resulting mean and variance are respectively $\bar{\gamma_c} = 5.626, \text{Var}(\gamma_c) = 0.033$, indicating a tight concentration around the mean.

A.3 Derivation of Eq. (1)

Here, we show that vanilla Skip-gram with the cross-entropy loss where the target distribution is represented as a one-hot vector induces an implicit pulling action on co-occuring words and pushes away other words.

For a given context word c, the cross-entropy loss is:

$$H(p(\cdot|c), \hat{p}(\cdot|c)) = -\sum_{w \in W} \hat{p}(w|c) \log p(w|c),$$

where $p(w|c) = \frac{e^{v_c^\intercal u_w}}{\sum_{w' \in W} e^{v_c^\intercal u_{w'}}}$ denotes the predicted distribution by Skip-gram. $\hat{p}(\cdot|c)$ denotes the target distribution where:

$$\forall w \in W : \hat{p}(w|c) = \begin{cases} 1 & \text{if } w \text{ is the target word} \\ 0 & \text{otherwise} \end{cases}$$

By construction of $\hat{p}(w|c)$, each term in the sum of the cross-entropy loss reduces to:

$$\hat{p}(w|c) \log p(w|c) =$$
$$\begin{cases} -\log \frac{e^{v_c^\intercal u_w}}{\sum_{w' \in W} e^{v_c^\intercal u_{w'}}} & \text{if } w \text{ is the target word} \\ 0 & \text{otherwise} \end{cases}$$

Thus, for a fixed context word c and target word w, the loss of Skip-gram reduces to:

$$\mathcal{L}_{\text{SGNS}}(c, w) = -\log \frac{e^{v_c^\intercal u_w}}{\sum_{w' \in W} e^{v_c^\intercal u_{w'}}}.$$

Now, consider two words c, w that co-occur. Without loss of generality, if we assume w appears prior to c in the training corpus, Skip-gram first updates the context and target vectors of w and c respectively. Taking the gradient of $\mathcal{L}_{\text{SGNS}}$ with respect to v_a and u_b for two co-occurring words $a, b \in W$,

$$\frac{\partial \mathcal{L}_{\text{SGNS}}}{\partial v_a} = \sum_{w \in W} \left(\frac{e^{v_a^\intercal u_b}}{\sum_{w' \in W} e^{v_a^\intercal u_{w'}}} u_w \right) - u_b, \tag{5}$$

$$\frac{\partial \mathcal{L}_{\text{SGNS}}}{\partial u_b} = \left(\frac{e^{v_a^\intercal u_b}}{\sum_{w' \in W} e^{v_a^\intercal u_{w'}}} - 1 \right) v_a. \tag{6}$$

Observe that the gradients induce a pulling action between the vectors v_a and u_b.

Define the set of words that lie between c and w in the training corpus as C. Notice that c and w will co-occur with $\Delta - 1$ words. Hence, for each word $c' \in C = \{c_1, ..., c_{w-1}\}$, the gradient update in Eq. (5) and (6) is applied to all word pairs $(w, c_1), ..., (w, c_{w-1})$ and $(c, c_1), ..., (c, c_{w-1})$.

Consider the pulling action induced by the word pairs (w, c_i) and (b, c_i) for some $i \in [\Delta - 1]$. As we first update the context and target vectors for w and c_i, notice that

$$v_w^{\text{new}} = v_w + u_{c'} - \sum_{x \in W} \left(\frac{e^{v_w^\intercal u_{c'}}}{\sum_{w' \in W} e^{v_w^\intercal u_{w'}}} u_x \right),$$
$$u_{c'}^{\text{new}} = u_{c'} + \left(1 - \frac{e^{v_w^\intercal u_{c'}}}{\sum_{w' \in W} e^{v_w^\intercal u_{w'}}} \right) v_w. \tag{7}$$

Similarly, if we now update the context and target vectors for c and c_i,

$$v_c^{\text{new}} = v_c + u_{c'}^{\text{new}} - \sum_{x \in W} \left(\frac{e^{v_c^\intercal u_{c'}}}{\sum_{w' \in W} e^{v_c^\intercal u_{w'}}} u_x \right).$$

Plugging the expression for u_c^{new} in Eq. (7), we get

$$v_c^{\text{new}} = v_c + \left(1 - \frac{e^{v_w^\intercal u_{c'}}}{\sum_{w' \in W} e^{v_w^\intercal u_{w'}}} \right) v_w + u_{c'}$$
$$- \sum_{x \in W} \left(\frac{e^{v_w^\intercal u_c}}{\sum_{w' \in W} e^{v_w^\intercal u_{w'}}} u_x \right).$$

The expression above indicates that v_c is pulled towards v_w implicitly and shifted closer to $u_{c'}$ explicitly in the update process while pushing away the weighted average of all word vectors. This update resembles the push-pull action in CWM.

A.4 Derivation of Eq. (2)

For a fixed word pair i, j, GloVe's local objective is:

$$\mathcal{L}_{\text{GloVe}}(i, j) = f(X_{ij})(v_i^\intercal u_j + b_i + \tilde{b}_j - \log X_{ij}),$$

where X_{ij} is the co-occurrence count of words i and j, $f(X_{ij})$ is a weighting term, $b_i, \tilde{b}_j$ are bias terms, and v_i, u_j denote the word vector and context word vectors respectively (cf. Pennington et al., 2014). Typically, $f(X_{ij})$ is set to $\min\{(X_i/X_{\text{max}})^\alpha, 1\}$ where X_i denotes the occurrence count of word i and $X_{\text{max}} = 100$. For the sake of demonstrating the pushing action in the gradient update, we consider a weighting function $f(X_{ij}) = \min\{(X_i/X_{\text{max}})^\alpha + \epsilon, 1\}$ for a arbitrarily small $\epsilon > 0$.

The derivative of the local objective with respect to v_i and u_j are:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{GloVe}}}{\partial v_i} &= 2f(X_{ij})(v_i^\intercal u_j + b_i + \tilde{b}_j - \log X_{ij})u_j,\\ \frac{\partial \mathcal{L}_{\text{GloVe}}}{\partial u_j} &= 2f(X_{ij})(v_i^\intercal u_j + b_i + \tilde{b}_j - \log X_{ij})v_i.\end{aligned} \tag{8}$$

Consider two co-occurring words c, w and a word w' that co-occurs with neither. Then, there exists a word c' that co-occurrs with c and w but does not co-occur with w'. Define $X_{c'w'} = 0, X_{cc'} = \omega_c, X_{wc'} = \omega_w$ where $\omega_c, \omega_w \in \mathbb{N}$.

With Eq. (8), the updated vectors for c, w, w' can be written as:

$$\begin{aligned}v_c^{\text{new}} &= v_c^{\text{old}} + 2f(\omega_c)(v_c^\intercal u_{c'} + b_c + \tilde{b}_{c'} - \log \omega_c)u_{c'},\\ v_w^{\text{new}} &= v_w^{\text{old}} + 2f(\omega_w)(v_w^\intercal u_{c'} + b_w + \tilde{b}_{c'} - \log \omega_w)u_{c'},\\ v_{w'}^{\text{new}} &= v_{w'}^{\text{old}} + 2f(\epsilon)(v_{w'}^\intercal u_{c'} + b_{w'} + \tilde{b}_{c'} - \log \epsilon)u_{c'}.\end{aligned}$$

As $\forall i, j : f_{X_{ij}} > 0$, notice that

$$\begin{aligned}(v_c^\intercal u_{c'} + b_c + \tilde{b}_{c'} - \log \omega_c) &< 0,\\ (v_w^\intercal u_{c'} + b_w + \tilde{b}_{c'} - \log \omega_w) &< 0,\\ (v_{w'}^\intercal u_{c'} + b_{w'} + \tilde{b}_{c'} - \log \epsilon) &> 0,\end{aligned}$$

for sufficiently large ω_c and ω_w and for sufficiently small ϵ. Setting $2 \cdot |f(X_{ij})(v_i^\intercal u_j + b_i + \tilde{b}_j - \log X_{ij})| = g(i, j)$, we see that

$$\begin{aligned}v_c^{\text{new}} &= v_c^{\text{old}} + g(c, c')v_{c'},\\ v_w^{\text{new}} &= v_w^{\text{old}} + g(w, c')v_{c'},\\ v_{w'}^{\text{new}} &= v_{w'}^{\text{old}} - g(w', c')v_{c'}.\end{aligned}$$

This indicates that v_c and v_w will be pulled towards the context word vectors of words that c and w both co-occur with, while words that do not co-occur with c and w will be pushed away from v_c and v_w.

Model	▱	WordSim	MEN	SimLex
CWM	0.27	0.66	0.73	0.34
SGNS	0.29	0.72	0.74	0.36
GloVe	0.29	0.61	0.75	0.37

Table 4: Performances for embedding models on parallelogram analogy recovery and word similarity tasks. ▱ refers to parallelogram recovery task. For word similarity, reported values are Spearman's rank correlation between word similarity rankings of human annotators and cosine similarites computed from word vectors.

B Supplementary Experiments

B.1 Metric Details

Given a set of word pairs in an analogy $A = \{(a_1, b_1), (a_2, b_2), \dots\}$, PCS measures relative directional alignment by computing the separability of cosine similarities between true vector offsets $v_{a_i} - v_{b_i}$ and false offsets $v_{a_i} - v_{b_j}, i \neq j$. Concretely, denoting the set of true and false offsets as P and N respectively, PCS computes the expectation of the ROC-AUC score between P and a subset of the false vector offsets $N' \subset N$ where $|P| = |N'|$:

$$\text{PCS}(A) = \mathbb{E}_{N' \sim U(N)}\left[\text{AUC}(P, N')\right],$$

where $U(N)$ denotes the uniform distribution over all false vector offsets. Typically, the expectation is approximated by sampling $s = 50$ subsets.

In contrast, MSM represents the absolute alignment within analogies by computing the cosine similarities between all true vector offsets and the mean of the true offsets.

$$\text{MSM}(A) = \frac{1}{|P|}\sum_{v_p \in P} \cos\left(v_p, \frac{1}{|P|}\sum_{v_p \in P} v_p\right)$$

A high value of MSM indicates alignment between true vector offsets. However, note that MSM is susceptible to scoring undesirable vector structures with high values (e.g. when all vectors are collapsed onto one point in the embedding space, MSM = 1).

B.2 Word Similarity

While the analogy task is our primary focus, we evaluate CWM on other commonly used benchmarking tasks for completeness. To this end, we benchmark our model on WordSim353 (Finkelstein et al., 2002), the MEN Test Collection (Bruni et al., 2014), and SimLex999 (Hill et al., 2015).

On all tasks, CWM performs comparably with existing models (Table 4). We highlight that minor performance differences on word similarity tasks are negligible, as such benchmarks are built using human annotations and are subject to noise. Nevertheless, we believe further refinement of the CWM model is required to boost performance on various downstream tasks.

B.3 Analogies as Parallelograms

We also benchmark all models on the traditional parallelogram analogy recovery task using the BATS dataset.

Concretely, given an analogy pair $a : b = c : d$, we utilize the most common metric where we compute the x that satisfies:

$$x = \min_{x \in W \setminus \{a,b,c\}} \|v_b - v_a + v_c - v_x\|,$$

and compare whether $x = d$.

Results from Table 4 indicate that CWM recovers analogies as parallelograms comparably to existing models.