White Rose Research Online

eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk
Universities of Leeds, Sheffield and York

Deposited via The University of Sheffield.

White Rose Research Online URL for this paper:
https://eprints.whiterose.ac.uk/id/eprint/184560/

Version: Accepted Version

Article:
Salle, A. and Villavicencio, A. (2022) Understanding the effects of negative (and positive) pointwise mutual information on word vectors. Journal of Experimental and Theoretical Artificial Intelligence, 35 (8). pp. 1161-1199. ISSN: 0952-813X

https://doi.org/10.1080/0952813X.2022.2072004

This is an Accepted Manuscript of an article published by Taylor & Francis in Journal of Experimental and Theoretical Artificial Intelligence on 15 Jun 2022, available online: http://www.tandfonline.com/10.1080/0952813X.2022.2072004

Reuse
Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.

UNIVERSITY OF LEEDS
University of Sheffield
UNIVERSITY of York

ARTICLE TEMPLATE

Understanding the Effects of Negative (and Positive) Pointwise Mutual Information on Word Vectors

Alexandre Salle[a] and Aline Villavicencio[a,b]

[a]Institute of Informatics, Federal University of Rio Grande do Sul, BR
[b]Department of Computer Science, University of Sheffield, UK

ARTICLE HISTORY
Compiled April 1, 2022

ABSTRACT
Despite the recent popularity of contextual word embeddings, static word embeddings still dominate lexical semantic tasks, making their study of continued relevance. A widely adopted family of such static word embeddings is derived by explicitly factorizing the Pointwise Mutual Information (PMI) weighting of the cooccurrence matrix. As unobserved cooccurrences lead PMI to negative infinity, a common workaround is to clip negative PMI at 0. However, it is unclear what information is lost by collapsing negative PMI values to 0. To answer this question, we isolate and study the effects of negative (and positive) PMI on the semantics and geometry of models adopting factorization of different PMI matrices. Word and sentence-level evaluations show that only accounting for positive PMI in the factorization strongly captures both semantics and syntax, whereas using only negative PMI captures little of semantics but a surprising amount of syntactic information. Results also reveal that incorporating negative PMI induces stronger rank invariance of vector norms and direction, as well as improved rare word representations.

KEYWORDS
word embedding; lexical semantics; pointwise mutual information

1. Introduction

Contextualized word embeddings (Devlin, Chang, Lee, & Toutanova, 2019; Peters et al., 2018) – where the vector representation of a word is dependent on context – are the mainstay of natural language processing (NLP) in sequence-related tasks such as semantic textual similarity and text classification, largely replacing static (non-contextual) word embeddings such as word2vec (Mikolov, Chen, Corrado, & Dean, 2013) in these applications. However, static word embeddings still dominate most *lexical semantic tasks* (such as word synonymy, similarity, relatedness, categorization, and analogy completion) (Lenci, Sahlgren, Jeuniaux, Gyllensten, & Miliani, 2021) where words are given *out-of-context*, making static word embeddings the appropriate choice. Given the continued importance of static word embeddings (from hereon simply referred to as word embeddings or vectors), this paper follows a line of work that aims to understand what contributes to the strong performance of these models in lexical semantic tasks.

CONTACT Alexandre Salle. Email: alex@alexsalle.com

Word vectors can be learned by exploiting the distributional hypothesis (Harris, 1954), paraphrased by Firth (1957) as *"a word is characterized by the company that it keeps"*. One approach is to use as vectors the rows of a word-context cooccurrence matrix and re-weight it using Pointwise Mutual Information (PMI) (Church & Hanks, 1990). The cooccurrence matrix M is constructed by sliding a symmetric window over the training corpus and for each center word $w \in V$ (V is the word vocabulary) and context word $c \in V$ within the window, incrementing $M_{w,c}$. The PMI matrix is then equal to:

$$PMI_{w,c} = \log \frac{P(w,c)}{P(w)P(c)} = \log \frac{\frac{M_{w,c}}{M_{*,*}}}{\frac{M_{w,*}}{M_{*,*}} \frac{M_{*,c}}{M_{*,*}}} = \log \frac{M_{w,c} M_{*,*}}{M_{w,*} M_{*,c}} \tag{1}$$

where * denotes summation over the corresponding index. We refer to the set $\{(w,c) \mid PMI_{w,c} \leq 0\}$ as negative pointwise mutual information (nPMI), the set $\{(w,c) \mid PMI_{w,c} > 0\}$ as positive pointwise mutual information (pPMI), and the set $\{(w,c) \mid M_{w,c} = 0\}$ as maximally negative pointwise mutual information (mnPMI).

Although the rows of the PMI matrix can be used directly as word vectors (Bullinaria & Levy, 2007; Levy & Goldberg, 2014a; Schütze, 1993), performing the low-rank factorization $PMI = WC^T$ (where word vectors W and context vectors C are $|V| \times d$) yields word vectors which are more computationally friendly – since $d \ll |V|$, they use significantly less memory and require smaller input matrices when used as inputs to neural networks. They also arguably lead to better generalization by compressing representations (eliminating noise) through a small set of latent variables (Deerwester, Dumais, Landauer, Furnas, & Harshman, 1990). The most popular method for generating this type of factorization is the word2vec Skip-gram model (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, Chen, Corrado, & Dean, 2013), which Goldberg and Levy (2014) theoretically prove *implicitly* factorizes the $PMI - \log k$ matrix (the M and PMI matrices are never constructed), where k is the number of negative samples.

Another family of methods (Bullinaria & Levy, 2007; Pennington, Socher, & Manning, 2014; Salle, Villavicencio, & Idiart, 2016; Shazeer, Doherty, Evans, & Waterson, 2016; Turney & Pantel, 2010; Xin, Yuan, He, & Jose, 2018) perform this factorization *explicitly*: M is constructed, re-weighted, then factorized. Note that although the popular GloVe model of Pennington et al. (2014) factorizes $\log M$ rather than PMI, bias terms learnt during factorization correlate strongly to the terms in the denominator of eq. (1) (Shazeer et al., 2016), suggesting that GloVe is also learning a factorization of a PMI variant.

Unfortunately, $PMI_{w,c}$ goes to negative infinity when the word-context pair (w,c) does not appear in the training corpus. Due to unreliable statistics, this happens very frequently in finite corpora: the matrix M is highly sparse. Models work around this issue by either: 1) altogether ignoring mnPMI values in the factorization (GloVe, Pennington et al. (2014)); 2) smoothing mnPMI values to make the numerator non-zero (Shazeer et al., 2016; Turney & Pantel, 2010); or 3) clipping nPMI values at 0, a measure known as Positive PMI (PPMI)[1] (Bullinaria & Levy, 2007; Salle, Villavicencio, & Idiart, 2016; Shazeer et al., 2016; Xin et al., 2018).

Although these approaches work well in practice, they raise the following questions about negative pointwise mutual information (for compactness referred to as *negative information* from hereon):

[1]Not to be confused with the pPMI set defined above.

(Q1) What is lost/gained by collapsing nPMI values to 0 as with the PPMI measure?
(Q2) More interestingly, what kind of information is captured in the word vectors if we exclusively consider nPMI *or* pPMI values when performing the factorization? In other words, if all we knew is that "word w [tends *not*/tends] to occur with words $c_i, \ldots, c_j$" (respectively for nPMI and pPMI), what can we learn about w? This can help us train better models, but perhaps more importantly, improve our understanding of natural language.
(Q3) Given that the factorization is low-rank, it leads to reconstruction errors: $PMI \approx WC^T$. A good factorization should minimize the loss function:

$$L_{factor} = \sum_{w,c \in V \times V} \lambda(w,c)(PMI_{w,c} - W_w C_c^T)^2 \tag{2}$$

where λ is the weight placed on the reconstruction error of pair (w,c). As previously stated, in the extreme case of GloVe, $\forall (w,c) \in mnPMI \quad \lambda(w,c) = 0$. In the context of this study, we ask what happens as λ increases for all (w,c) in nPMI? In other words, if a factorization accounts for more and more negative information, how are word vectors and the information they capture affected?

Answering these questions would grant greater insights into both (a) methods which perform explicit factorization of the *PMI* matrix, and the roles played by negative and positive information and (b) the implicit factorization performed by the Skip-gram model, whose strange geometry observed by Mimno and Thompson (2017) is supported by the results of this paper. It would also lead to a better understanding of (c) how different types of information are distributed in natural language.

In this paper, we give an initial answer to these three questions by training different word embedding models and evaluating them on tasks that test for semantics or syntax, focusing on English. As described in section 3.2, as representative of similar methods we use LexVec (Salle, Villavicencio, & Idiart, 2016) for greater experimental control and interpretability of results, since it allows us to factorize arbitrary PMI variants, selectively use only nPMI, pPMI, or both, and control the amount of negative information used in the factorization.

Although augmenting word embeddings with subword information – such variants include fastText (Skip-gram) (Bojanowski, Grave, Joulin, & Mikolov, 2017) and subword LexVec (Salle & Villavicencio, 2018) – can improve performance on some tasks, the sharing of information between words through subword vectors makes it impossible to isolate the effects of negative information on words of different frequencies (i.e. vector representations of rare word forms are improved *not only* because of improved corpus statistics, but because they also directly share vector representation with frequent forms), so we focus on the base variants that do not use subword information, but include in appendix C matching experimental results for LexVec and fastText subword models which support the conclusions from the main paper.

In summary, the contributions of this paper are:

- **The proposal of two PMI variants**, clipped PMI and normalized negative PMI, that account for the distribution of nPMI both in terms of the range and distribution of values within a set (*Section 3*). Results explicitly justify the popularity of PPMI by showing that collapsing the negative distribution to 0 does not substantially hurt results when compared to preserving it.

- An examination of the degree with which **nPMI and pPMI capture syntac-**

tic and semantic information (*Section 4*). Using only pPMI strongly captures both semantics and syntax, whereas using only nPMI captures some semantic information but (surprisingly) a lot of syntactic information. This deepens our understanding of distributional semantics and computational linguistics by extending Firth (1957)'s paraphrase of the distributional hypothesis to **"a word is not only characterized by the company that it keeps, *but also by the company it rejects*".**

- Empirical evaluation of **how the reconstruction error weights due to window sampling and negative sampling prioritize correct approximation of values in pPMI and nPMI** respectively in a model like LexVec (*Section 5*). We find that increasing the relative importance of negative information strengthens *geometric rank invariant properties* – vector norms and direction – of word vectors and improves the representation of rare words. Additionally, our analysis reveals that when word analogies are evaluated correctly (Schluter, 2018), performance improves as more negative information is used, suggesting that these geometric properties are connected to more strongly capturing the linear vector offsets used in answering analogies.

- Experiments reveal **similar results for Skip-gram, GloVe, and SVD models**, showing that the important role played by negative information in LexVec transfers well to these other models (*Section 6*).

2. Related Work

There is a long history of studying weightings (also known as association measures) of cooccurrence matrices in general, not only of word-context pairs; see Jurafsky (2000); Manning, Manning, and Schütze (1999); Schütze (1993) for an overview and Curran and Moens (2002) for comparison of different weightings. One widely adopted measure is PMI, and in fact, Bullinaria and Levy (2007) show that word vectors derived from PPMI matrices perform better than alternative weightings for word-context cooccurrence. Moreover, Levy and Goldberg (2014b) show theoretically that the popular Skip-gram model (Mikolov, Chen, et al., 2013) performs implicit factorization of shifted PMI. Another PMI variant is normalized PMI, which Bouma (2009) proposed for dealing with negative infinity ($-\infty$), for collocation extraction.

Recently, work in explicit low-rank matrix factorization of PMI variants has achieved state of the art results in word embeddings. GloVe (Pennington et al., 2014) performs weighted factorization of the log cooccurrence matrix with added bias terms, but does not account for zero cells. Shazeer et al. (2016) point out that GloVe's bias terms correlate strongly with unigram log counts, suggesting that GloVe is factorizing a variant of PMI. Their SwiVel model modifies the GloVe objective to use Laplace smoothing and hinge loss for zero counts of the cooccurrence matrix, directly factorizing the PMI matrix, sidestepping the negative infinity issue. An alternative is to use PPMI and variants as done by Kiela and Clark (2014); Milajevs, Sadrzadeh, and Purver (2016); Polajnar and Clark (2014); Salle, Villavicencio, and Idiart (2016); Xin et al. (2018). However, even though PPMI works well in practice, its use may seem unprincipled, as it is not clear what is lost by clipping the negative distribution of PMI. Accounting for zero cells in the factorizations of Salle, Villavicencio, and Idiart (2016); Shazeer et al. (2016); Xin et al. (2018) is motivated by better representing rare words. However,

they do not test the effect of nPMI and pPMI in isolation, or investigate the geometry of the resulting word vectors. Moreover, according to Schluter (2018), they do not perform correct evaluation of word analogies, as described in section 5.2.

The continued relevance of static word embeddings has led to a number of recent papers that aim to understand their properties. These include research on why word analogies ("*a* is to *b* as *c* is to *?*") hold (Allen, Balazevic, & Hospedales, 2019; Allen & Hospedales, 2019; Ethayarajh, Duvenaud, & Hirst, 2019; Hashimoto, Alvarez-Melis, & Jaakkola, 2016), or conversely why they do not hold (Linzen, 2016; Rogers, Drozd, & Li, 2017; Schluter, 2018); on their geometry (Mimno & Thompson, 2017); and on possible biases they incorporate (Bolukbasi, Chang, Zou, Saligrama, & Kalai, 2016; Gonen & Goldberg, 2019; Nissim, van Noord, & van der Goot, 2019). This paper follows this line of research into understanding the workings of these models. In particular, Mimno and Thompson (2017) analysis of the geometry of the Skip-gram model concludes that it is "strange" as word vectors occupy a narrow cone in space diametrically opposed to context vectors. Our analysis of model geometry is directly inspired by their work, but rather than analyzing the implicit PMI factorization of Skip-gram, we look at explicit factorizations, under which the "strange" geometry of Skip-gram can be explained by looking at the underlying PMI counts.

Schluter (2018) observed that in analogies of the form "*a* is to *b* as *c* is to ?" if *a,b,c* are not excluded from the set of possible answers, performance of the Skip-gram and GloVe models plummets. Linzen (2016); Rogers et al. (2017) made similar observations that these models do not quite seem to capture the geometry necessary to correctly answer analogies. However, none of these works *simultaneously* relate (a) increasing negative information with (b) word vector geometry for different frequencies and (c) analogy performance. Although Allen and Hospedales (2019); Ethayarajh et al. (2019) theoretically show that these linear analogies should hold in PMI factorizations, there is no investigation into how the geometry of different factorizations affects results. Allen et al. (2019) present a theoretical argument for why word embeddings that are linear projections of PMI capture certain semantic relationships, but do not look at the distribution of PMI and the importance of its negative values.

3. PMI and Matrix Factorization

In this section, we first look at the distribution of PMI values to get a sense for the sets nPMI, pPMI, and mnPMI. We then propose PMI variants to address the negative infinity issue for mnPMI values. Lastly, we describe the matrix factorization method we use in our investigation.

3.1. PMI & Negative Information

Before we can look at PMI values, a cooccurrence matrix M is needed. We construct it from a lowercased, alphanumerical 2015 English Wikipedia dump with 3.8B tokens, discarding tokens with frequency < 100, for a vocabulary V_w of $303,517$ words. Throughout this paper, we follow Mikolov, Sutskever, et al. (2013) in using a symmetric window of size 5 drawn from $U(1,5)$ for each target word. We also performed identical experiments using positional contexts and fixed window size of 2, as used in Salle, Idiart, and Villavicencio (2016). Results included in appendix B lead to the same conclusions as those for the larger randomized windows used in the main paper.

We use the additional heuristic of token *subsampling* (Mikolov, Sutskever, et al.,

2013) the training corpus: tokens for word w are randomly discarded with probability $p_w = \max(0, 1 - \sqrt{t/f_w})$, where t is the subsampling threshold (we follow Levy, Goldberg, and Dagan (2015); Mikolov, Sutskever, et al. (2013); Salle, Villavicencio, and Idiart (2016) and set $t = 10^{-5}$ throughout this paper) and f_w is the unigram frequency (tokens of w divided by total number of tokens in training corpus). For Skip-gram and LexVec, which perform factorization by sampling word-context pairs from the training corpus, subsampling accelerates training significantly. Mikolov, Chen, et al. (2013) also observe empirically that it improves the representation of uncommon words.

We refer to this cooccurrence matrix constructed from Wikipedia as M_{wiki}, and its PMI transformation as PMI_{wiki}.

Distribution of PMI: To better understand the distribution of PMI, we examine the PMI_{wiki} values of 10^5 non-zero pairs randomly sampled from M_{wiki}, shown in fig. 1. We sample only non-zero pairs because M_{wiki} is sparse: only 0.93% of cells are non-zero.

To the left of the 0 line, we can clearly see the distribution of nPMI that is collapsed when using the PPMI measure, which maps these negative values – $\sim 22.2\%$ of *cooccurring pairs* – to 0.

Preserving the distribution of negative information: To deal with values in mnPMI, we propose clipped PMI,

$$CPMI_{w,c}(z) = max(z, PMI_{w,c}) \tag{3}$$

which is equivalent to PPMI when $z = 0$ (z is the clipping threshold), and captures most of the nPMI distribution when $z \leq 2$.

We also experiment with normalized PMI ($NPMI$) (Bouma, 2009):

$$NPMI_{w,c} = PMI_{w,c} / -log(M_{w,c}/M_{**})$$

such that $NPMI(w, c) = -1$ when $(w, c) \in mnPMI$ (never cooccur), $NPMI(w, c) = 0$ when they are independent, and $NPMI(w, c) = 1$ when they always cooccur together. This effectively captures the entire negative distribution, but has the downside of normalization which discards scale information. In practice we find this works poorly if done symmetrically, so we introduce a variant called *NNEGPMI* which only normalizes nPMI:

$$NNEGPMI_{w,c} = \begin{cases} NPMI_{w,c} & \text{if } PMI_{w,c} < 0 \\ PMI_{w,c} & \text{otherwise} \end{cases}$$

We also experimented with Laplace smoothing as in Turney and Littman (2003) for various pseudocounts but found it to work consistently worse than both *CPMI* and *NNEGPMI* so we omit further discussion in this paper.

3.2. Matrix Factorization

As stated in section 1, low-rank word vectors obtained through the factorization of the *PMI* matrix are advantageous computationally and arguably lead to better generalization than directly using rows from *PMI* as word vectors. Since in our experiments we need to control whether only positive or negative information is used in the factorization, we cannot use the Skip-gram model since its implicit factorization does

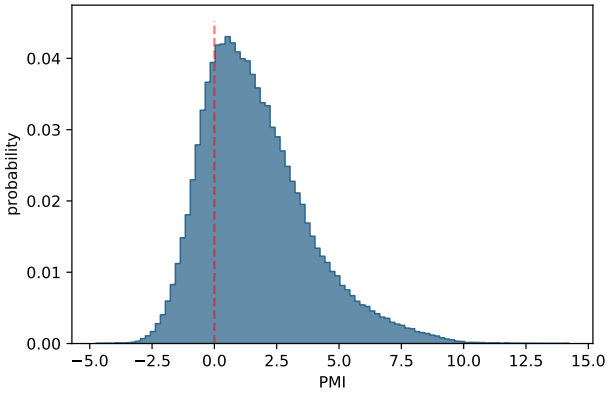


Figure 1. Histogram (bin width equal to 0.2) of 10^5 values *not in mnPMI* sampled from PMI_{wiki}. The negative distribution of PMI are the values to the left of the dashed line ($\sim$ 22.2% of sampled values), which are collapsed to 0 when using the popular PPMI association measure. Note that we exclude mnPMI (sample only non-zero cooccurrences) otherwise the graph would be a single vertical line at -2 (if graphing $CPMI(-2)$) since 99.07% of values in M_{wiki} are 0.

not allow us to access the underlying PMI values. This leaves as candidates models that perform explicit factorization: SVD (Deerwester et al., 1990; Levy & Goldberg, 2014b), GloVe (Pennington et al., 2014), Swivel (Shazeer et al., 2016), LexVec (Salle, Villavicencio, & Idiart, 2016), and AllVec (Xin et al., 2018).

Although SVD provably provides factorizations with the lowest possible squared loss $L_2(w,c) = \frac{1}{2}\lambda_{SVD}(w,c)(W_w C_c^\top - f(M)_{w,c})^2$, with $\lambda_{SVD}(w,c) = 1$ and an arbitrary transformation $f(\cdot)$ (Eckart & Young, 1936), Salle, Villavicencio, and Idiart (2016) show that, in word embedding where f is some variant of PMI, uniform weights λ significantly reduce the quality of the word vectors. The intuition is that if both word and context (w,c) are frequent, the estimated $PMI_{w,c}$ is more *statistically reliable* than if they were both rare, and the weights should reflect that. We call this the *reliability principle*: the loss function of a word embedding factorization of matrix M should have a weight λ on the reconstruction error of $PMI_{w,c}$ that is a monotonically increasing function of *both* $M_{w,*}$ and $M_{*,c}$.

We discard GloVe as a candidate because its weight function entirely ignores values in the mnPMI set (where $M_{w,c} = 0$):

$$L_{GloVe}(w,c) = \frac{1}{2}\lambda_{GloVe}(w,c)(W_w C_c^\top - f(M)_{w,c} + b_w + \tilde{b}_c)^2 \quad (4)$$

$$\lambda_{GloVe}(w,c) = \min(M_{w,c}^\alpha / x_{max}^\alpha, 1) \quad (5)$$

where $f(\cdot) = \log(\cdot)$, α is a constant and b and $\tilde{b}$ are bias terms.

Swivel is not used because of its $O(|V|^2)$ computational complexity from calculating *loss terms for every cell* in the matrix being factorized, thus requiring a large distributed computing environment to be practical: in our experiments $|V|^2 \approx 9.2e10$.

AllVec improves the GloVe objective by accounting for values in mnPMI:

$$L_{AllVec}(w,c) = \begin{cases} \frac{1}{2}\lambda_{GloVe}(w,c)(W_w C_c^\top - f(M)_{w,c})^2 & \text{if } (w,c) \notin mnPMI \\ \frac{1}{2}\lambda_{AllVec}(w,c)(W_w C_c^\top - r_\text{-})^2 & \text{otherwise} \end{cases} \tag{6}$$

$$\lambda_{AllVec}(w,c) = \alpha_0 \; M^\delta_{*,c} / \sum_{c \in V} M^\delta_{*,c} \tag{7}$$

where $r_\text{-}$ and α_0 are constants. While eq. (7) *does* account for values in mnPMI and allows us to choose what value to map $\text{-}\infty$ to (the term $r^\text{-}$), the weight term for these values depends only on c, not on both w *and* c, thus not obeying the reliability principle described previously.

LexVec performs matrix factorization by sliding a symmetric window over the training corpus (window sampling), in the exact same way as when M was constructed in section 3.1, and performing one Stochastic Gradient Descent (SGD) step every time a (w,c) pair is observed, minimizing

$$l(w,c) = \frac{1}{2}(W_w C_c^\top - f(M)_{w,c})^2 \tag{8}$$

Additionally, *negative sampling* is performed, where for every center word w, k negative samples (Mikolov, Chen, et al., 2013) are drawn from the unigram context distribution:

$$P_n(c) = (M_{*,c})^{\alpha_{cds}} / \sum_c (M_{*,c})^{\alpha_{cds}} \tag{9}$$

where α_{cds} is a smoothing factor – set to .75 in this paper following Levy et al. (2015); Mikolov, Sutskever, et al. (2013); Salle, Villavicencio, and Idiart (2016) – and SGD steps are taken to minimize

$$l_{neg}(w) = \sum_{i=1}^{k} \mathbf{E}_{c_i \sim P_n(c)} l(w, c_i) \tag{10}$$

The expected loss for a pair (w,c) in a single pass over the training corpus using both window sampling and negative sampling is:

$$\mathbf{E}[L_{LexVec}(w,c)] = M_{w,c} l(w,c) + \frac{M_{w,*}}{2l} \; k \; P_n(c) \; l(w,c) \tag{11}$$

$$= \frac{1}{2}\lambda_{LexVec}(w,c)(W_w C_c^\top - f(M)_{w,c})^2 \tag{12}$$

$$\lambda_{LexVec}(w,c) = \underbrace{M_{w,c}}_{\lambda_{ws}(w,c)} + \underbrace{\frac{M_{w,*}}{2l} \; k \; P_n(c)}_{\lambda_{ns}(w,c)} \tag{13}$$

In λ_{LexVec}, the first term prioritizes the correct approximation of frequently cooccurring pairs (window sampling), and the second term of pairs where either word occurs with high frequency (negative sampling), obeying the reliability principle previously described. *We thus use LexVec as the vehicle of our study* for this reason, and additionally because the complexity is linear in the size of the training corpus and not the

vocabulary size, which in the case of the English Wikipedia corpus we use here is 24x smaller than $|V|^2$.

Connection to Skip-gram: The Skip-gram loss function is very similar to the LexVec loss function. For each (w, c) pair observed in sliding a window over the training corpus, k negative samples are drawn from the unigram context distribution and the following objective function is *maximized*:

$$J_{sg}(w,c) = \log \sigma(W_w C_c^\top) + \sum_{i=1}^{k} \mathbf{E}_{c_i \sim P_n(c)} \log \sigma(-W_w C_{c_i}^\top) \tag{14}$$

where $\sigma(\cdot)$ is the logistic function. The expected loss for a single pass over the training corpus for a specific (w, c) is:

$$\mathbf{E}[J_{sg}(w,c)] = M_{w,c} \log \sigma(W_w C_c^\top) + M_{w,*}\ k\ P_n(c)\ \log \sigma(-W_w C_c^\top) \tag{15}$$

$$= \lambda_{ws}(w,c) \log \sigma(W_w C_c^\top) + 2l\ \lambda_{ns}(w,c) \log \sigma(-W_w C_c^\top) \tag{16}$$

Under the assumption that the embedding dimension is high enough such that $J(\cdot)$ terms can be maximized independently for different word-context pairs, Levy and Goldberg (2014b) show that the matrix $WC^T = PMI - \log k$ maximizes J. This, combined with the appearance of the LexVec weight terms $\lambda_{ws}, \lambda_{ns}$ in J suggests that the geometric observations made for the LexVec model will hold for the Skip-gram model as well. Both window and negative sampling will draw PMI values that have the same distribution as in fig. 2, and dot products between word and context vectors will approach these values minus the $\log k$ shift. However, in contrast to this apparent similarity, there is a subtle difference between the models in the way errors are weighted. Suppose we have found parameters that perfectly factorize the PMI matrices, so that in LexVec $WC^T = CPMI(-2)$ and in Skip-gram $WC^T = PMI - \log k$. We then introduce a small error ϵ into the dot product $W_w C_c^T$ so that

$$L_{LexVec+\epsilon}(w,c) = \frac{1}{2}(\lambda_{ws}(w,c) + \lambda_{ns}(w,c))\ \epsilon^2 \tag{17}$$

$$J_{sg+\epsilon}(w,c) = \lambda_{ws}(w,c) \log \sigma(PMI_{w,c} - \log k + \epsilon) \tag{18}$$

$$+ 2l\ \lambda_{ns}(w,c) \log \sigma(-(PMI_{w,c} - \log k + \epsilon)) \tag{19}$$

The LexVec loss depends only on the weights and the error, whereas the Skip-gram objective depends on the weight, the error, and *the PMI value*; because of the logistic function, reconstruction errors for PMI values near zero affect the objective much more than for high and low PMI values. If we re-word small reconstruction errors as *sharp* and large reconstruction errors as *fuzzy*, the fuzziness of LexVec approximations is entirely determined by the weights: the window sampling weight λ_{ws} encourages sharp approximations for high PMI values, the negative sampling weight λ_{ns} for low PMI values. The Skip-gram objective, in contrast, has opposing forces, with these same "sharpnesses" encouraged by $\lambda_{ws}, \lambda_{ns}$ being opposed by the fuzziness of the logistic function at extreme PMI values. Despite this subtle difference in error weightings, as we will see in section 6.2, the models behave very similarly empirically, such that the ideas developed here apply to the Skip-gram model as well.

Connection to GloVe: GloVe factorizes the logarithm of the cooccurrence matrix with added bias terms (eq. (5)). Although it is not clear what the optimal values of these bias terms are, Shazeer et al. (2016) observe that these terms are highly

correlated to the respective word and context corpus frequencies, such that the matrix being approximated could resemble PMI (eq. (1)). This, combined with the fact that GloVe's loss weighting resembles the window sampling weight λ_{ws}, suggests that results for the LexVec model with no negative sampling might hold for the GloVe model as well. That is, they might exhibit similar task performance and geometry.

Connection to SVD: Since $\lambda_{SVD}(w, c) = 1$ is uniform for all word-context pairs, the fraction of $|nPMI|/|nPMI \cup pPMI|$ gives the fraction of total loss weights assigned to negative information, which in the case of PMI_{wiki} is equal to 99.27%. The SVD is thus an extreme case of prioritizing negative information, and as we will see in experiments, behaves similarly to LexVec and Skip-gram models with high number of negative samples.

4. Semantics/syntax in negative/positive information

In this section, we examine the type of information captured by the PMI distribution, focusing on how syntactic and semantic information are reflected in positive and negative PMI. In particular, these experiments aim to answer the following questions: *Does negative information capture more semantic or syntactic information? Is the distribution of negative information important? What is captured by positive information? What is the benefit in using both?*

4.1. Materials

Models: In order to identify the role that nPMI and pPMI play in distributional semantics, we train two LexVec models:

- one that only considers negative information, nPMI, i.e. any pair in pPMI is skipped during factorization, or equivalently, $\lambda_{LexVec} = 0$ if $(w, c) \in pPMI$, and
- one model that only considers positive information, pPMI, i.e. any pair in nPMI is skipped during factorization, or equivalently $\lambda_{LexVec} = 0$ if $(w, c) \in nPMI$.

We compare these to models that include both negative and positive information ($nPMI \cup pPMI$) to see how the two interact. To account for values in mnPMI, we use the four PMI variants described in section 3.1: PPMI, CPMI(-2), NPMI, NNEGPMI.

We use the following LexVec configuration for all PMI variants: window size $l = 5$, embedding dimension of 300, 5 negative samples, learning rate of 0.025, no subword information, and negative distribution power $\alpha_{cds} = 0.75$.

For all experiments, we use the English Wikipedia corpus described in section 3.1, resulting in the same underlying M_{wiki} matrix for all models.

For comparison, we include results for a randomly initialized, untrained embedding to establish task baselines.

Semantic tasks: To evaluate word-level semantics, we use the SimLex (Hill, Reichart, & Korhonen, 2015) and Rare Word (RW) (Luong, Socher, & Manning, 2013) word similarity datasets. To evaluate word analogies, we use the Google Semantic (GSem) analogies (Mikolov, Chen, et al., 2013). We evaluate sentence-level semantics on the Semantic Textual Similarity (STSB) task (Cer, Diab, Agirre, Lopez-Gazpio, & Specia, 2017) with averaged bag of vectors representations (BoV, summing the vectors of each word in a sentence and dividing the sum by sentence length) using SentEval[2]

[2]Classifiers for all SentEval tasks are multilayer perceptrons with a single hidden layer of 100 units and

Table 1. Performance on tasks focused on semantics or syntax by models that use only positive information (p*), negative information (n*), or both (no prefix), and the random baseline. Using negative information alone performs far better than the random baseline, especially on the syntactic tasks. Metrics: Spearman rank correlation (×100) for SimLex and RW word similarity; Pearson correlation for STSB; % accuracy for GSem/GSyn/MSR word analogies, POS tagging and WC, Dep, TopC probing tasks. Best result for each column in bold, second best underlined.

model	SimLex	RW	GSem	STSB	GSyn	MSR	POS	Dep	TopC
pPPMI	**37.0**	40.1	58.8	**65.4**	52.7	35.1	92.0	27.1	30.4
nPPMI	4.0	1.8	0.0	48.6	0.0	0.0	16.3	17.9	5.0
nCPMI(-2)	22.6	25.2	18.3	41.4	24.5	18.3	90.6	**32.9**	**33.7**
nNPMI	9.9	21.8	8.2	38.3	9.6	5.8	89.0	<u>31.1</u>	<u>32.3</u>
PPMI	<u>34.0</u>	**45.3**	76.5	61.6	55.1	36.7	91.7	25.5	26.6
CPMI(-2)	<u>34.0</u>	41.8	**78.4**	<u>61.9</u>	**58.7**	**42.6**	**92.2**	27.3	28.4
NPMI	26.0	39.4	60.0	60.6	44.4	30.2	91.4	26.3	27.9
NNEGPMI	<u>34.0</u>	<u>43.0</u>	<u>78.3</u>	61.7	<u>56.3</u>	<u>39.8</u>	<u>92.0</u>	25.1	26.3
Random	4.0	1.9	0.0	45.3	0.0	0.0	59.1	17.9	5.0

(Conneau, Kruszewski, Lample, Barrault, & Baroni, 2018).

Syntactic tasks: Similarly, we use the Google Syntactic analogies (GSyn) (Mikolov, Chen, et al., 2013) and MRS syntactic analogies (Luong et al., 2013) to evaluate word-level syntactic information. Google Syntactic analogies are in fact morphological but many categories test for POS relations and are therefore syntactic in nature. We employ the Depth (Dep) and Top Constituent (TopC) (of the input sentence's constituent parse tree) probing tasks from SentEval (Conneau et al., 2018) to evaluate sentence-level syntax. Our final syntactic task is part-of-speech (POS) tagging using FLAIR (Akbik et al., 2019) with the same BiLSTM-CRF setup as Huang, Xu, and Yu (2015) but using only word embeddings (no hand-engineered features) as input, trained on the WSJ section of the Penn Treebank (Marcus, Marcinkiewicz, & Santorini, 1993).

4.2. Results

The results shown in table 1 provide insights into the role of negative and positive PMI for capturing semantic and syntactic information.

Negative PMI: $nCPMI(-2)$ and $nNPMI$ perform similarly to full distribution models in POS tagging and both syntactic probing tasks (Dep and TopC), but very poorly on all semantic tasks, suggesting that nPMI mostly encodes syntactic information. Our hypothesis to explain this phenomenon is that the grammar that generates language implicitly creates negative cooccurrence and so nPMI encodes this syntactic information. Interestingly, this idea creates a bridge between distributional semantics and the argument by Regier and Gahl (2004) that indirect negative evidence might play an important role in human language acquisition of grammar.

The $nPPMI$ model is almost identical to the random baseline; this is to be expected as random initialization gives in expectation perpendicular vectors or equivalently dot

dropout of 0.1.

products equal to zero, and if the only learning signal is to make dot products equal to zero nothing changes.

Positive PMI: The *pPPMI* model which accounts for only values in pPMI performs similarly to the full distribution models on most tasks, clearly indicating that pPMI encodes both semantic and syntactic information.

Why incorporate nPMI? *pPPMI* falters on the RW and analogy tasks, and accounting for nPMI significantly improves performance on both tasks. Section 5.3 explores how increasing the relative importance of negative information increases rank invariance in the word vectors and improves results on both these tasks relative to using only pPMI.

Full distribution models: Of the models which account for all PMI values ($nPMI \cup pPMI$), the PPMI, *CPMI-2*, and *NNEGPMI* models perform similarly, whereas the *NPMI* model is significantly worst on nearly all tasks. We thus conclude that accounting for scale in the positive distribution is more important than in the negative distribution.

Collapsing the negative distribution: The PPMI model, which collapses the negative distribution to zero, performs comparably to the *CPMI-2* and *NNEGPMI* models that account for the range of negative values on most tasks. However, preserving the scale and distribution of negative values (*CPMI-2* rather than $PPMI$) consistently improves performance on all syntactic tasks.

In summary, negative information alone strongly captures syntactic information, and preserving the distribution of negative values benefits tasks that are syntactic in nature. In contrast, positive information captures *both* semantic and syntactic information. Using both positive and negative information makes model performance more robust across tasks.

5. Negative information and geometry

Mimno and Thompson (2017) observed that as negative sampling is increased, Skip-gram word vectors occupy a narrowing cone in embedding in space and point away from context vectors. For further insight, it would be interesting to investigate additional effects of increased negative sampling, and generalize these results to other embedding models. Thus in this section, we conduct experiments to answer the following questions: *What is the relationship between negative sampling and negative information? How does increasing the relative weight (or importance) of negative information affect the geometry of the word vector space?*

5.1. Negative sampling vs. negative information

In our study, we need a way to gradually increase the loss weights in the negative distribution of PMI, λ_{LexVec} for $(w, c) \in nPMI$, relative to pairs in the positive distribution, $(w, c) \in pPMI$, to be able to answer the question above. Window sampling weights λ_{ws} are fixed given the data, so the only control we have over λ_{LexVec} is k, the number of negative samples. We need to show that increasing the number of negative samples in the LexVec factorization increases the relative weight of negative information.

To show this, we sample 10^5 values from $CPMI(-2)_{wiki}$ using window sampling and negative sampling and plot the distribution of these values in fig. 2. This plot shows the reconstruction error weights λ_* different sampling regimes assign to different PMI values. For example, the peak at -2 tells us that negative sampling weights λ_{ns} will be

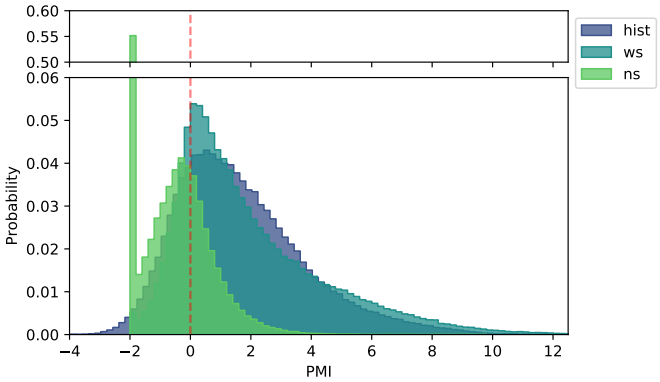


Figure 2. The distribution (bin width equal to 0.2) of sampled $CPMI_{wiki}(-2)$ values when using window sampling and negative sampling. Histogram from fig. 1 included for comparison. These window sampling and negative sampling distributions of PMI values correspond to the reconstruction error weights $\lambda_{ws}, \lambda_{ns}$ in eq. (13) as a function of PMI. Negative sampling assigns high weights to values in nPMI, and window sampling to values in pPMI.

higher for (w, c) pairs in nPMI than for (w, c) pairs in pPMI. The opposite is observed for window sampling weights, which assign more weight to values in pPMI. Table 2 shows this same result by aggregating values in fig. 2.

These results confirm that, as the number of *negative samples* is increased, so is the relative weight of *negative information* in the factorization. This enables us to use the LexVec model to investigate the impact of increasing negative information on resulting word vector geometry.

5.2. Materials

Models: As shown in section 5.1, when using both window sampling and negative sampling, increasing the number of negative samples effectively increases the relative importance of negative information. We use the default LexVec setting in which both nPMI and pPMI are used (no steps are skipped), and increase the number of negative samples from 0 to 1, 2, 4, 5, 10, 15, and 20. We focus on $CPMI(-2)$ since as described in section 3.1 it closely mimics the measure of ultimate interest which is PMI.

Analogies: Analogies of the form "a is to b as c is d" are evaluated by finding the word d^* such that:

$$d^* = argmax_{w \in V_w} Cos(W_w, W_c + W_b - W_a) \tag{20}$$

$$Cos(u, v) = \frac{u \cdot v}{|u||v|} \tag{21}$$

If $d^* = d$, the analogy is said to hold in the vector space. Schluter (2018) points out two flaws in the way this evaluation is conducted in works such as Mikolov, Chen, et al. (2013); Pennington et al. (2014):

(1) Normalization (Norm/N): The vector space is *distorted* by normalizing all word vectors to unit length before the term $W_c + W_b - W_a$ is calculated.
(2) Premise exclusion (Prem/P): The set $\{w \in V\}$ in eq. (20) is replaced by $\{w \in V \setminus \{a, b, c\}\}$ – the analogy's premises are excluded from the set of candidate answers. In practice this improves performance because it is often the case that

Table 2. Set (rows) membership of samples for various sampling methods (columns). Cell values are the percentage of samples for a given method that fall within a set, such that nPMI+pPMI sum to 100. **Full**: computed over all cells in PMI_{wiki}. **Hist**: computed over the same 10^5 sampled pairs plotted in fig. 1. **WS** and **NS**: computed over the same 10^5 sampled pairs used in fig. 2 using window sampling (ws) and negative sampling (ns). Observe that window sampling is heavily skewed towards pairs in pPMI, and negative sampling heavily skewed towards pairs in nPMI.

Name	Set	Full	Hist	WS	NS
nPMI: Negative information	$\{(w,c) \mid PMI_{w,c} \leq 0\}$	99.27	22.24	19.35	81.75
pPMI: Positive information	$\{(w,c) \mid PMI_{w,c} > 0\}$	0.73	77.76	80.65	18.25
mnPMI: Maximally-negative information	$\{(w,c) \mid M_{w,c} = 0\}$	99.07	0.00	0.00	51.33
nPMI\mnPMI: Collapsed negative information under PPMI	$\{(w,c) \mid PMI_{w,c} \leq 0 \wedge M_{w,c} > 0\}$	0.21	22.24	19.35	30.42

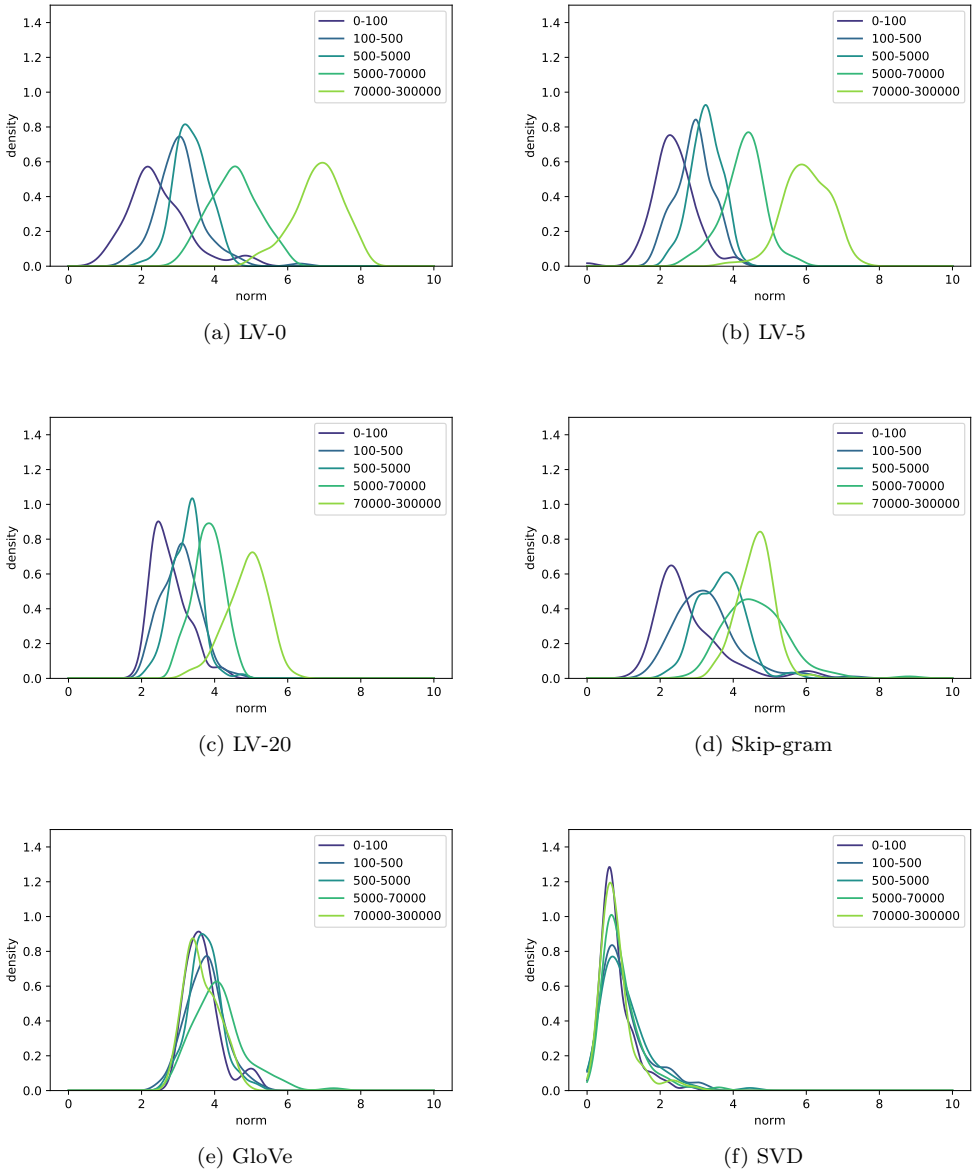
$W_b - W_a \approx 0$ and so $d* = c$ if c is not excluded from the candidates (Linzen, 2016).

Although these strategies work well in practice, significantly improving accuracy, they mask the degree to which the linear relationship $W_a - W_b \approx W_c - W_d$ is present in the vector space. This is particularly problematic as it may lead to wrong conclusions. For instance, using the popular GoogleNews word2vec vectors, the answer to the analogy "man is doctor as woman is to ?" is "nurse" (Bolukbasi et al., 2016) if both Norm and Prem are performed, when in fact, if analogies are evaluated correctly, the actual answer is "doctor" as well (Nissim et al., 2019).

In this work, we perform incorrect evaluation where both strategies are used (e.g. GSem, GSyn, MSR) partially correct evaluation where one strategy is excluded (e.g. GSem-N, MSR-P), and correct evaluation where both strategies are excluded (e.g. GSem-N-P).

Geometry × frequency: To understand how increased negative sampling affects the geometry of words of different frequencies, we evaluate performance using SimLex (which consists of frequent words) and RW (which consists of frequent-rare word) word similarity datasets. To perform the same frequency analysis on analogies, we order the analogies in the analogy datasets by the highest rank of any of the words in each analogy. We take the first 10% and last 10% analogies to create frequent (GSemF, GSynF, MSRF) and rare (GSemR, GSynR, MSRR) word analogy datasets, respectively, with -N and -N-P variants when strategies (N: normalization, P: premise exclusion) are excluded as described above. Table A1 in the appendix A gives percentile rank statistics for all datasets we use.

5.3. Results



(a) LV-0 (b) LV-5

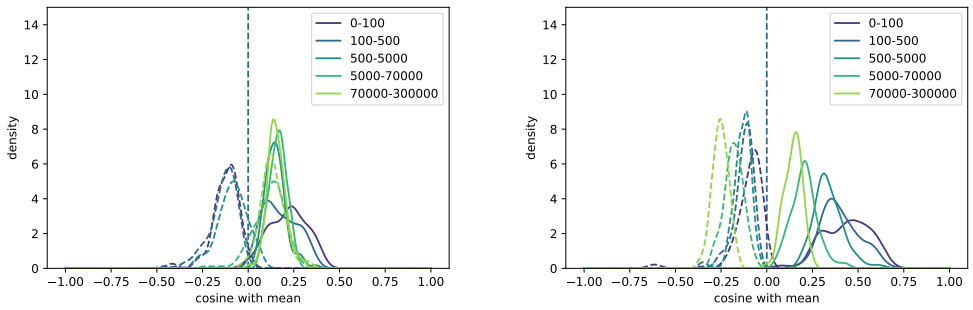
(c) LV-20 (d) Skip-gram

(e) GloVe (f) SVD

Figure 3. The distribution of vector L_2 norms for 100 words sampled from various frequency buckets for LexVec models using 0, 5, and 20 negative samples (LV-0, LV-5, LV-20). As the number of negative samples increases, the norms become more rank invariant, with means of the different buckets becoming increasingly closer and variance decreasing. The norm distributions for Skip-gram, GloVe, and SVD are shown for comparison.

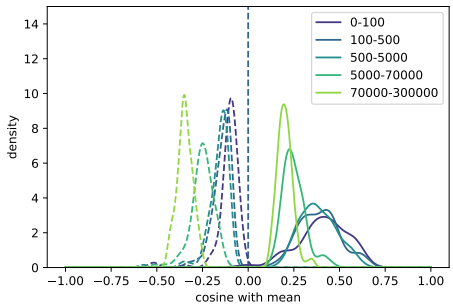
Norms: In fig. 3, we plot the distribution of vector L_2 norms for 100 words sampled from different frequency buckets for LexVec models using 0, 5, and 20 negative samples (LV-k denotes the model with k negative samples). We use the same buckets as Mimno and Thompson (2017), indexing words by inverse frequency (most frequent first), 0-100, 100-500, 500-5000, 5000-70000, and defining an additional bucket 70000-300000 for extremely rare words. With an increasing number of negative samples, the relative

weight of negative information is increased, and *vector norms become rank invariant*; the means of the different buckets becomes increasingly closer and variance decreases as negative samples are increased.

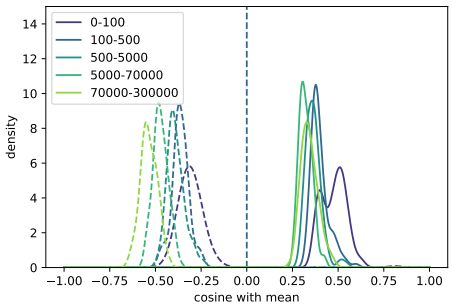


(a) LV-0

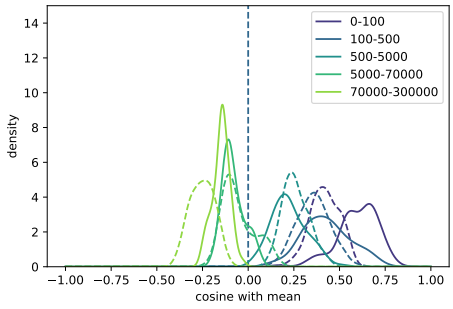
(b) LV-5



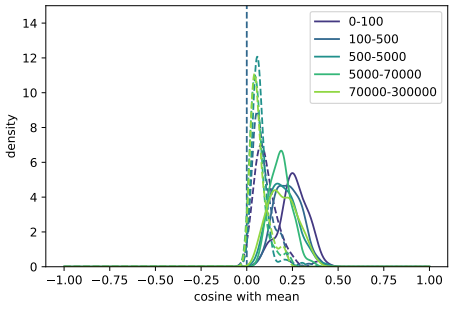
(c) LV-20



(d) Skip-gram



(e) GloVe



(f) SVD

Figure 4. In solid lines, the distributions of cosines (eq. (21)) of word vectors from different frequency buckets with the mean vector of word vectors from all buckets, and in dashed lines the distributions of cosines of *context* vectors from different frequency buckets with this *same mean word vector*. As the number of negative samples increases, word vectors increasingly point in the same direction and word-context vectors point in the opposite direction.

Direction: In fig. 4, we perform the experiment of Mimno and Thompson (2017), where using the same sampled words and frequency buckets as in fig. 3, we calculate the mean vector of all sampled words from all buckets, and plot the distribution of the cosine (eq. (21)) of sampled word vectors (solid line) and correponding context vectors (dashed lines) with this mean vector. Here we observe that as the number of

negative samples increases, word vectors increasingly point in the same direction, and context vectors point in opposite direction. As a preface to section 6.2, these are the same effects observed by Mimno and Thompson (2017) for the Skip-gram model.

Table 3. Nearest neighbors of words sampled from SimLex. Subscripts denote the percentile rank of a word (0.0 is most frequent word, 100.0 is rarest). All models exhibit semantically coherent neighbors, with the exception of the SG model which has some unrelated *intruders*, notably for the word "interest".

word	model	neighbors
interest $_{\mathbf{0.3}}$	LV-0	interests $_{\mathbf{1.0}}$, conflict $_{\mathbf{0.5}}$, interested $_{\mathbf{0.7}}$, scientific $_{\mathbf{0.5}}$, particular $_{\mathbf{0.3}}$, expertise $_{\mathbf{2.3}}$, reason $_{\mathbf{0.2}}$, wizardimps $_{\mathbf{67.2}}$, attention $_{\mathbf{0.5}}$, desire $_{\mathbf{1.3}}$
	LV-20	interests $_{\mathbf{1.0}}$, interested $_{\mathbf{0.7}}$, scientific $_{\mathbf{0.5}}$, attention $_{\mathbf{0.5}}$, particular $_{\mathbf{0.3}}$, conflict $_{\mathbf{0.5}}$, focus $_{\mathbf{0.5}}$, concerned $_{\mathbf{1.0}}$, nature $_{\mathbf{0.4}}$, piqued $_{\mathbf{26.1}}$
	SG	bshsu $_{\mathbf{90.2}}$, conflict $_{\mathbf{0.5}}$, btheeuropeanlibrary $_{\mathbf{87.6}}$, bwral $_{\mathbf{65.5}}$, bepochtimes $_{\mathbf{41.4}}$, richarddawkins $_{\mathbf{74.5}}$, thegauntlet $_{\mathbf{78.0}}$, interested $_{\mathbf{0.7}}$, wizardimps $_{\mathbf{67.2}}$, towsonedu $_{\mathbf{95.1}}$
	GloVe	interests $_{\mathbf{1.0}}$, interested $_{\mathbf{0.7}}$, concern $_{\mathbf{0.9}}$, attention $_{\mathbf{0.5}}$, focus $_{\mathbf{0.5}}$, knowledge $_{\mathbf{0.5}}$, conflict $_{\mathbf{0.5}}$, influence $_{\mathbf{0.6}}$, involvement $_{\mathbf{1.2}}$, subject $_{\mathbf{0.2}}$
	SVD	interests $_{\mathbf{1.0}}$, interested $_{\mathbf{0.7}}$, attention $_{\mathbf{0.5}}$, substantial $_{\mathbf{1.1}}$, share $_{\mathbf{0.7}}$, own $_{\mathbf{0.1}}$, credit $_{\mathbf{0.8}}$, benefit $_{\mathbf{1.0}}$, debt $_{\mathbf{1.6}}$, financial $_{\mathbf{0.5}}$
cup $_{\mathbf{0.1}}$	LV-0	champions $_{\mathbf{0.5}}$, uefa $_{\mathbf{1.3}}$, championship $_{\mathbf{0.2}}$, cups $_{\mathbf{2.7}}$, league $_{\mathbf{0.1}}$, finals $_{\mathbf{0.7}}$, trophy $_{\mathbf{1.1}}$, tournament $_{\mathbf{0.4}}$, fifa $_{\mathbf{1.1}}$, championships $_{\mathbf{0.3}}$
	LV-20	champions $_{\mathbf{0.5}}$, cups $_{\mathbf{2.7}}$, championship $_{\mathbf{0.2}}$, trophy $_{\mathbf{1.1}}$, finals $_{\mathbf{0.7}}$, league $_{\mathbf{0.1}}$, competitions $_{\mathbf{0.9}}$, runners $_{\mathbf{1.5}}$, scorer $_{\mathbf{2.2}}$, tournament $_{\mathbf{0.4}}$
	SG	cups $_{\mathbf{2.7}}$, champions $_{\mathbf{0.5}}$, championship $_{\mathbf{0.2}}$, finals $_{\mathbf{0.7}}$, trophy $_{\mathbf{1.1}}$, league $_{\mathbf{0.1}}$, supercup $_{\mathbf{8.5}}$, uefa $_{\mathbf{1.3}}$, scorers $_{\mathbf{4.7}}$, intertoto $_{\mathbf{12.4}}$
	GloVe	championship $_{\mathbf{0.2}}$, cups $_{\mathbf{2.7}}$, champions $_{\mathbf{0.5}}$, tournament $_{\mathbf{0.4}}$, league $_{\mathbf{0.1}}$, uefa $_{\mathbf{1.3}}$, finals $_{\mathbf{0.7}}$, championships $_{\mathbf{0.3}}$, trophy $_{\mathbf{1.1}}$, matches $_{\mathbf{0.4}}$
	SVD	runners $_{\mathbf{1.5}}$, nextseason $_{\mathbf{4.9}}$, champions $_{\mathbf{0.5}}$, prevseason $_{\mathbf{4.9}}$, cups $_{\mathbf{2.7}}$, scorers $_{\mathbf{4.7}}$, competitions $_{\mathbf{0.9}}$, scorer $_{\mathbf{2.2}}$, fifa $_{\mathbf{1.1}}$, matches $_{\mathbf{0.4}}$
soul $_{\mathbf{0.9}}$	LV-0	love $_{\mathbf{0.2}}$, blues $_{\mathbf{0.8}}$, heaven $_{\mathbf{1.5}}$, funk $_{\mathbf{2.7}}$, album $_{\mathbf{0.1}}$, mind $_{\mathbf{0.5}}$, spirit $_{\mathbf{0.8}}$, souls $_{\mathbf{3.0}}$, god $_{\mathbf{0.4}}$, gospel $_{\mathbf{1.6}}$
	LV-20	heaven $_{\mathbf{1.5}}$, funk $_{\mathbf{2.7}}$, love $_{\mathbf{0.2}}$, essence $_{\mathbf{3.0}}$, souls $_{\mathbf{3.0}}$, mind $_{\mathbf{0.5}}$, eternal $_{\mathbf{2.5}}$, forever $_{\mathbf{1.7}}$, spirit $_{\mathbf{0.8}}$, dreams $_{\mathbf{1.6}}$
	SG	soulful $_{\mathbf{11.0}}$, funk $_{\mathbf{2.7}}$, love $_{\mathbf{0.2}}$, heaven $_{\mathbf{1.5}}$, blues $_{\mathbf{0.8}}$, essence $_{\mathbf{3.0}}$, funky $_{\mathbf{6.1}}$, souls $_{\mathbf{3.0}}$, temptations $_{\mathbf{9.3}}$, changeless $_{\mathbf{89.4}}$
	GloVe	blues $_{\mathbf{0.8}}$, funk $_{\mathbf{2.7}}$, mind $_{\mathbf{0.5}}$, hop $_{\mathbf{1.1}}$, hip $_{\mathbf{1.0}}$, love $_{\mathbf{0.2}}$, rap $_{\mathbf{2.1}}$, spirit $_{\mathbf{0.8}}$, pop $_{\mathbf{0.4}}$, heaven $_{\mathbf{1.5}}$
	SVD	heaven $_{\mathbf{1.5}}$, forever $_{\mathbf{1.7}}$, dreams $_{\mathbf{1.6}}$, eternal $_{\mathbf{2.5}}$, dream $_{\mathbf{0.9}}$, love $_{\mathbf{0.2}}$, souls $_{\mathbf{3.0}}$, funk $_{\mathbf{2.7}}$, spirit $_{\mathbf{0.8}}$, destiny $_{\mathbf{3.0}}$

Nearest neighbors: We perform qualitative analysis of nearest neighbors of words sampled from SimLex in table 3 and RW in table 4, where neighbors are ordered by

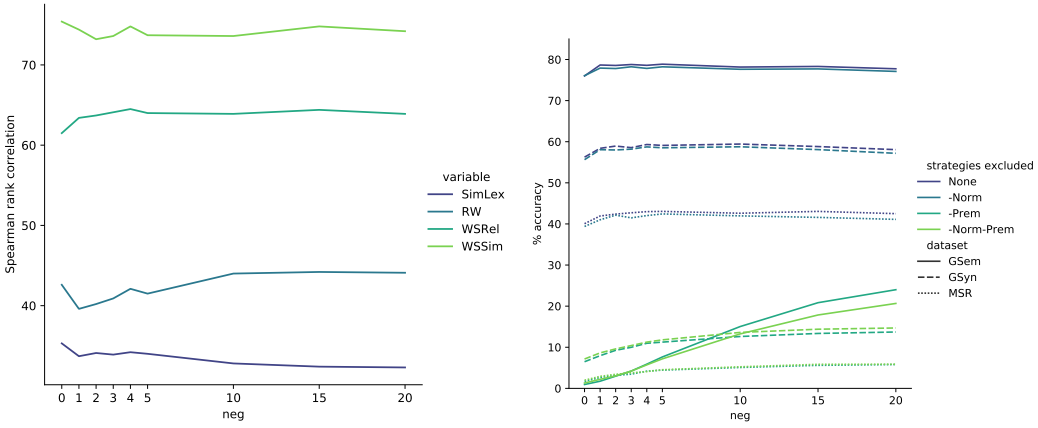
Table 4. Same as table 3, but with words sampled from RW. LV-0 neighbors have no clear semantic connection to the target word. In the LV-20 model, have clear semantic relations to the targets. GloVe behaves like LV-0, and Skip-gram and SVD like LV-20.

word	model	neighbors
rooters $_{77}$	LV-0	sonorella $_{98}$, bhavas $_{67}$, recreative $_{86}$, wwwjskscoin $_{98}$, wwwlegion $_{95}$, damsels $_{44}$, hassane $_{72}$, maniraptorans $_{99}$, abolboda $_{52}$, sympycnus $_{47}$
	LV-20	cheered $_{15}$, cheering $_{11}$, howled $_{73}$, jubilant $_{26}$, hissed $_{75}$, bosox $_{81}$, mcgreevy $_{53}$, rogell $_{62}$, jeers $_{43}$, mobbed $_{33}$
	SG	nuxhall $_{73}$, sparky $_{11}$, yanks $_{15}$, strupper $_{87}$, whitey $_{12}$, chisox $_{83}$, schoendienst $_{56}$, altrock $_{96}$, clendenon $_{85}$, campaneris $_{62}$
	GloVe	paiks $_{93}$, trashmen $_{96}$, wampanoags $_{93}$, mycenaeans $_{79}$, highnesses $_{60}$, perseids $_{89}$, clubmen $_{92}$, thalians $_{98}$, guelf $_{99}$, housecarls $_{93}$
	SVD	ballplayers $_{31}$, shibe $_{31}$, semipro $_{60}$, ebbets $_{29}$, phanatic $_{71}$, kekiongas $_{80}$, mudville $_{73}$, comiskey $_{18}$, mutuals $_{33}$, nabbp $_{48}$
monocultures $_{76}$	LV-0	pegomya $_{94}$, atara $_{81}$, shebang $_{40}$, subsidization $_{69}$, kiyomori $_{38}$, lucullus $_{23}$, intercal $_{76}$, paaerduag $_{99}$, dagbon $_{80}$, voluntas $_{89}$
	LV-20	monoculture $_{35}$, crops $_{2}$, agroforestry $_{34}$, replanting $_{35}$, silviculture $_{45}$, replanted $_{27}$, overgrazing $_{26}$, clearcutting $_{47}$, rainfed $_{65}$, seedlings $_{11}$
	SG	monoculture $_{35}$, overgrazed $_{81}$, polyculture $_{88}$, silvicultural $_{84}$, crops $_{2}$, woodlots $_{73}$, intercropping $_{90}$, overstory $_{91}$, allelopathic $_{83}$, croplands $_{60}$
	GloVe	chlamydospores $_{88}$, microbubbles $_{94}$, renunciations $_{92}$, insectoids $_{91}$, monoculture $_{35}$, plasmodesmata $_{87}$, relaxations $_{79}$, contactors $_{66}$, urocystis $_{57}$, vortexes $_{90}$
	SVD	monoculture $_{35}$, polyculture $_{88}$, dryland $_{33}$, silviculture $_{45}$, fuelwood $_{62}$, clearcutting $_{47}$, replanting $_{35}$, agroforestry $_{34}$, seedlings $_{11}$, swidden $_{59}$
flighted $_{83}$	LV-0	ablabesmyia $_{31}$, hydroptila $_{40}$, sphex $_{69}$, semiotus $_{82}$, sympycnus $_{47}$, coelichneumon $_{60}$, prajapati $_{32}$, diorhabda $_{98}$, quizzer $_{73}$, wwwtsuru $_{96}$
	LV-20	ratite $_{63}$, ratites $_{35}$, dromaeosaurids $_{50}$, flightless $_{14}$, tinamous $_{38}$, raptorial $_{60}$, parrots $_{8}$, psittaciformes $_{41}$, maniraptorans $_{99}$, psittacidae $_{32}$
	SG	raptorial $_{60}$, flightlessness $_{92}$, tibiotarsus $_{88}$, dromaeosaurids $_{50}$, zygodactyl $_{57}$, maniraptorans $_{99}$, apomorphic $_{93}$, ratite $_{63}$, rectrices $_{65}$, hindlimbs $_{39}$
	GloVe	sunbathe $_{92}$, hypnotise $_{99}$, maniraptorans $_{99}$, illidan $_{80}$, languorous $_{98}$, githyanki $_{95}$, dichotomius $_{73}$, tmesisternus $_{54}$, chloroceryle $_{89}$, quadroon $_{72}$
	SVD	zygodactyl $_{57}$, raptorial $_{60}$, beaks $_{13}$, flightless $_{14}$, avians $_{73}$, opposable $_{42}$, featherless $_{60}$, ratite $_{63}$, forelimbs $_{17}$, pronated $_{92}$

descending cosine similarity (eq. (21)) – similarity decreases from left to right – and

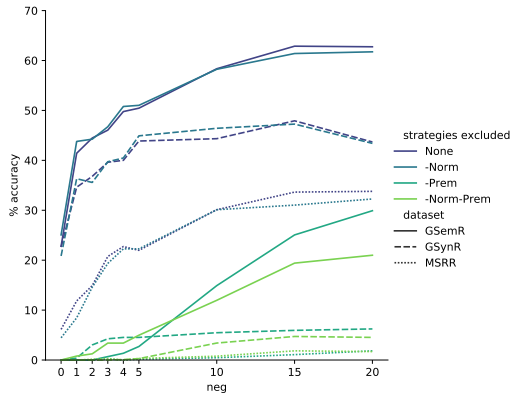
subscripts denote the percentile rank of a word (0.0 is most frequent word, 100.0 is rarest).

In both the LV-0 and LV-20 model, frequent words have semantically related, frequent word neighbors, showing that increasing negative information has no effect on the semantic similarity of frequent words. Results change completely with rare words, where LV-0 neighbors are rare words and – barring a few exceptions – have no obvious semantic connection to the target word, e.g. the nearest neighbor of "monocultures" (an agriculutural practice) is "fieldensis" (a species of arthropod). With the LV-20 model, on the other hand, the neighbors of target rare words are generally of higher frequency than the targets, and have clear semantic relations, as with "monocultures" and "monoculture" (singular) or "seedlings" (young plants). Qualitatively, and as we will see in the next section, quantitatively, increasing the importance of negative information has a positive impact on the representation of rare words.

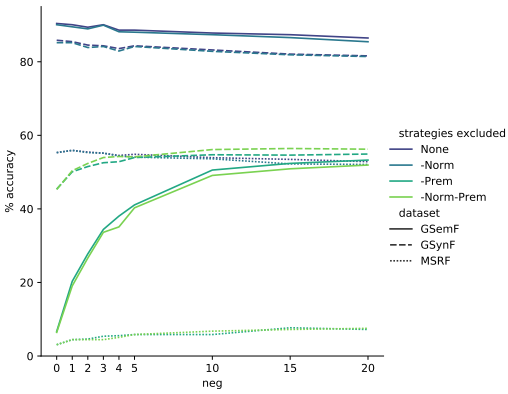


(a) Word similarity

(b) Analogies



(c) Analogies, rare words



(d) Analogies, frequent words

Figure 5. (a) Word similarity results: RW improves as the number of negative samples increases. There is a small but consistent drop in performance on SimLex when increasing the number of negative samples. Scores in WSRel do indeed increase, but there is no clear drop in similarity scores in WSSim. (b) GSem/GSyn/MSR: Performance increases as the number of negative samples increases. Normalization has a minor effect, whereas premise exclusion is critical to performance. (c) GSem/GSyn/MSR Rare words: Consistent improvements in all conditions because rare words are better represented. (d) GSem/GSyn Frequent words: With both strategies or only the normalization strategy performance is nearly constant. For conditions without premise exclusion strategy, there is remarkable improvement in analogy performance as negative samples increase, especially for semantic analogies. Metrics: for (a) Spearman rank correlation (×100), for (b,c,d), Accuracy.

Word similarity tasks: Shown in fig. 5a, results on RW improve consistently as the number of negative samples is increased. There is a small but consistent drop in performance on SimLex when increasing the number of negative samples. Looking at the SimLex nearest neighbor examples in table 3, the semantic similarities are indistinguishable between the LV-0 and LV-20 models, even though the LV-0 model has a marginally higher SimLex score. We tested the WS353 word similarity dataset (Finkelstein et al., 2001), split into relatedness (WSRel) and similarity (WSSim), to see if this drop on SimLex scores, which measures similarity, is due to an increase in relatedness; that is, if related pairs such as (psychology,Freud) are being drawn closer together in space to the detriment of semantically similar pairs such as (psychology,psychiatry). We observe in fig. 5a that relatedness scores in WSRel do indeed increase, and that there is a small drop in similarity scores in WSSim. This suggests that as relatedness increases – which happens as the relative importance of negative information is increased – true similarity drops. This possible tradeoff between similarity and relatedness is an interesting investigation for future work.

Word analogies: Figure 5b shows how analogy performance varies as the number of negative samples is increased. Clearly performance increases as the number of negative samples increases, in particular for the semantic analogies. Removing the normalization strategy has a minor effect on task performance. Removing the premise exclusion strategy, however, leads to a tremendous drop in performance. This is why it is crucial to perform correct evaluation of word analogies without additional strategies, for an accurate assessment of the semantic and syntactic information represented by the word embeddings.

The strong results with the premise exclusion strategy might lead one to believe that the linear offsets corresponding to the analogies are straightforwardly accessible in the vector space when they clearly are not. However, if the number of negative samples is increased, we see a stark improvement under the correct evaluation, suggesting that these linear offsets/regularities manifest more clearly in the vector space.

To determine if this improvement is not merely due to better representations of rare words induced by increased negative sampling, as discussed previously, we look at the (GSem/GSyn/MSR)F/R datasets which contain the analogies with most/least frequent words. For (GSem/GSyn/MSR)R, we see the expected results in fig. 5c: consistent improvements with and without strategies because rare words are better represented. The surprising result is for (GSem/GSyn/MSR)F in fig. 5d: with both strategies or only the normalization strategy performance is nearly constant. Without the premise exclusion strategy, however, there is a remarkable improvement in analogy performance, especially for semantic analogies. This suggests that negative sampling is altering the geometry of the vector space in such a way that the linear offsets used to solve analogies hold more strongly, without resorting to strategies.

In summary, we established a direct relationship between negative sampling and negative information, and showed that increasing the relative importance of negative information within the LexVec model increases the rank invariance of vector direction and norms, improves the representation of rare words (which can also be cast as an increase in rank invariance of the coherence of semantic representations), and significantly improves analogy performance under correct evaluation.

6. Connection to other models

In this section, we conduct experiments aimed at answering the question: *Are these conclusions generalizable?* In particular, *do conclusions for the LexVec model apply to popular models such as Skip-gram, GloVe, and SVD?* Given that Skip-gram implicitly factorizes a shifted PMI matrix, GloVe's factorization is related to a PMI factorization, and SVD performs optimal unweighted factorization, we investigate how these factorizations compare to the explicit PMI factorization performed by LexVec as regards increasing negative information. In particular, we are interested in the (dis)similarities of the geometry of the resulting word vector spaces.

6.1. Materials

Skip-gram: We train a Skip-gram model using the same parameters from the original paper (Mikolov, Chen, et al., 2013), with window size $l = 5$ and number of negative samples in $\{1, 2, 5\}$ (we refer to these models as SG-1, SG-2 and SG-5). Note that Skip-gram with 5 negative samples performs the same amount of computation as the LexVec model with 20 negative samples: Skip-gram draws 5 negative samples per target-context pair ($2 \times 2 \times 5 = 20$ for each window), whereas LexVec draws 20 negative samples per window. Analogous parameters have the same values as the LexVec models: embedding dimension of size 300, learning rate of 0.025, negative distribution power of 0.75, subsampling threshold of $1e-5$.

GloVe: The GloVe configuration follows the configuration of the original paper (Pennington et al., 2014), but with three changes to make the results directly comparable to LexVec: (1) unlike in the original paper, the corpus is subsampled using a threshold of $1e-5$ before constructing the cooccurrence matrix. (2) Window size $l = 5$ to match LexVec and Skip-gram models. (3) Word vectors are output without averaging with context vectors, so that word vectors and context vectors can be analyzed separately. All other parameters are kept: embeddings of size 300, 100 training epochs, and learning rate of 0.05.

SVD: A limitation of the truncated SVD is that its computational efficiency is contingent on the sparsity of the input matrix. This sparsity is lost when using the $CPMI(-2)$, so we must use the zero-preserving transform $PPMI$. Given the truncated SVD, $PPMI = U_d \Sigma_d V_d^T$, which discards all but the top d singular values, we follow Levy et al. (2015) and set word and context matrices to $W = U_d\sqrt{\Sigma_d}$, $C = V_d\sqrt{\Sigma_d}$ respectively. We factorize the $PPMI$ transform of M_{wiki}, setting $d = 300$.

6.2. Results

Given these similarities in loss functions presented in section 3.2, we expect the Skip-gram model trained using 5 negative samples to resemble the LexVec LV-20 model that draws 20 negative samples (since, as stated in section 6.1, this leads to the same number of total negative samples) and the GloVe model to resemble the LV-0 which uses only window sampling. We expect the patterns that emerge as the number of negative samples increase to manifest clearly in the SVD model, which assigns the vast majority of its loss weights to negative values. Table 5 shows task results for all compared models.

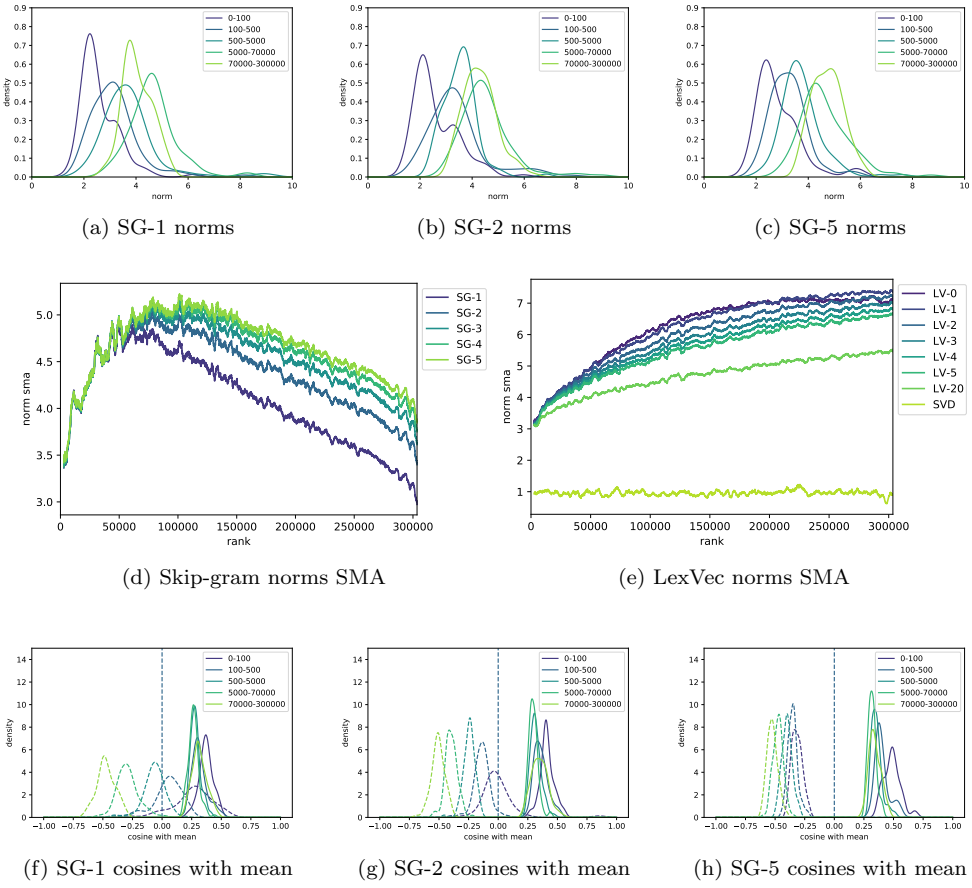
Table 5. Comparing task performance of LexVec, Skip-gram, GloVe, and SVD models in Word similarity and analogy tasks (-: no strategies excluded, -N: norm strategy excluded, -P: premise strategy excluded, -N-P: both strategies excluded). The LV-0 model is similar to the GloVe model: strong SimLex results, weaker RW results, and similar across all analogy evaluations, barring MSR where GloVe is considerably stronger than all other models. Analogously, the LV-20 model is similar to the Skip-gram and models: strong SimLex and RW results, and consistently strong performance on all analogy evaluations. SVD has the lowest performance, but note that it suffers the smallest drop when the -P strategy is excluded. Metrics: Spearman rank correlation (×100) for SimLex and RW, Accuracy for analogies.

model	SimLex	RW	GSem	GSem^{-N}	GSem^{-P}	GSem^{-N-P}
LV-0	35.3	42.6	76.0	76.1	0.9	1.3
LV-5	34.0	41.5	<u>78.9</u>	<u>78.2</u>	7.6	7.2
LV-20	32.3	44.1	77.7	77.1	**24.0**	**20.7**
SG-1	<u>36.0</u>	45.5	75.3	75.3	11.2	10.5
SG-2	**36.4**	<u>46.5</u>	77.4	76.9	11.7	10.6
SG-5	35.9	**46.9**	**78.9**	**78.4**	11.2	10.1
GloVe	35.4	40.6	74.4	74.3	5.6	6.0
SVD	28.6	41.9	43.9	41.7	<u>20.5</u>	<u>20.6</u>

model	GSyn	GSyn^{-N}	GSyn^{-P}	GSyn^{-N-P}	MSR	MSR^{-N}	MSR^{-P}	MSR^{-N-P}
LV-0	56.2	55.6	6.5	7.1	40.1	39.4	1.7	2.0
LV-5	59.1	58.5	11.2	11.8	43.0	42.4	4.4	4.5
LV-20	58.1	57.2	**13.7**	**14.7**	42.5	41.1	<u>5.8</u>	<u>5.9</u>
SG-1	61.9	60.4	11.4	12.2	45.6	44.8	3.7	3.7
SG-2	<u>63.3</u>	<u>61.8</u>	11.3	12.4	<u>47.3</u>	<u>46.6</u>	4.0	4.1
SG-5	63.0	61.1	10.9	11.8	46.3	45.2	3.1	3.7
GloVe	**64.5**	**64.0**	<u>13.3</u>	<u>14.0</u>	**58.1**	**57.6**	**7.9**	**8.5**
SVD	42.0	40.8	9.2	8.9	26.9	23.1	4.8	3.9

6.2.1. Skip-gram

- **Norms:** figs. 6a to 6c show the word vector norm distributions for the Skip-gram models. In contrast to the same figures for the LexVec models (figs. 3a to 3c), for Skip-gram it is not *clear* if increasing the number of negative samples increases the rank invariance of vector norms. Delving deeper, we plot a simple moving average of period 100 ($SMA_{100}(i) = \sum_{t=0}^{99} |W_{i-t}|/100$) of vector norms as a function of word rank in figs. 6d and 6e. Although the shape of both functions is different, it is clear for both Skip-gram and LexVec that as the number of negative samples increases the functions become flatter, indicating the increase in rank invariance of vector norms.
- **Directions:** In figs. 6f to 6h, we plot the distribution of cosines between words of different frequency buckets with the mean vector of all buckets for the SG models, as was drawn in figs. 4a to 4c for the LexVec models. Just as with



(a) SG-1 norms (b) SG-2 norms (c) SG-5 norms

(d) Skip-gram norms SMA (e) LexVec norms SMA

(f) SG-1 cosines with mean (g) SG-2 cosines with mean (h) SG-5 cosines with mean

Figure 6. (a,b,c): Word vector norm distributions for Skip-gram models with 1, 2 and 5 negative samples (SG-1, SG-2, SG-5). In contrast to the same figures for the LexVec models (figs. 3a to 3c), it is not clear if increasing the number of negative samples increases the rank invariance of vector norms. (d,e): Simple moving average of period 100 (with accompanying scatter plot of points used in calculating this average) of word vector norms as a function of word rank. We include additional SG-k and LV-k models for various k to make the trend clear. Although the shape of (d) and (e) is different, it is evident for both Skip-gram and LexVec that as the number of negative samples increases the functions become flatter, indicating the increase in rank invariance of vector norms. SVD displays the extreme case where nearly all loss weight is assigned to negative information, leading to ideal rank invariance. (f,g,h): Solid lines show distribution of cosines between vectors of words of different frequency buckets with the mean vector of all buckets, and dashed lines the of cosines of context vectors of these same words with the mean word vector, as in figs. 4a to 4c. Word vectors increasingly point in the same direction and word vectors point away from context vectors as the number of negative samples increases.

the LexVec models, word vectors point in the same direction and word vectors point away from context vectors as the number of negative samples increases. This is precisely what was observed as the "strange" geometry of Skip-gram in Mimno and Thompson (2017), here explained by increasing importance of negative information.

- **Word similarity and analogies:** Overall SG results in table 5 are similar to the LV-20 models, which is to be expected given the similarity in loss functions. All SG models achieve similar results, with the only clear trend being marginal improvements in RW performance as the number of negative samples increases. Thus increasing negative information has little effect on task results. We attribute this to the fact that even with the minimum number of negative samples

($k = 1$), there is one negative sample per observed word-context pair, in contrast to LexVec where with $k = 1$ there are $2l$ observed pairs. In other words, the *minimum* relative importance of negative information within the Skip-gram model is sufficiently high so as not to observe the poor rare word representations observed in the LV-0 and GloVe models. Note that setting $k = 0$ makes the Skip-gram objective (eq. (15)) ill-defined since it can be made arbitrarily high by aligning all word vectors and increasing their norms.

- **Nearest neighbors:** We performed qualitative analysis of word neighbors in tables 3 and 4 as was done with the LexVec models. Skip-gram neighbors are semantically related for both frequent and rare words, as is the case with other models that use negative information (LV-20 and SVD). The exception is the frequent word "interest" which has some incoherent neighbors. This irregularity deserves future investigation, but we suspect it is due to a weakness in the local nature of the Skip-gram model where, independent of the global PMI value for an observed pair (even if it is negative), a single step of optimization draws together the corresponding vectors. We omit results for SG-1,2 for there is no qualitative difference between the SG models as negative samples increase (our hypothesis for this is described above).

6.2.2. GloVe and SVD

- **GloVe:** In figs. 4a and 4e, LV-0 and GloVe behave similarly: vectors do not have a directional preference in respect to the mean vector and to context vectors. The similarity breaks in the distribution of vector norms in fig. 4e, which in LV-0 are far less rank invariant than in GloVe. We hypothesize that GloVe's vector norm rank invariance is due to bias terms which are responsible for scaling word-context dot products to approximate log cooccurrence count, allowing word/context vectors to have a similar norm.

  Word similarity and analogy results are given in table 5. The LV-0 model is similar to the GloVe model: good SimLex results, weak RW results, and consistently similar across all analogy evaluations, with the exception of MSR where the GloVe model outperforms all other models by a wide margin. The similarity is even clearer in the nearest neighbor samples in tables 3 and 4. For both models, frequent words have semantically related neighbors, and rare words have incoherent neighbors.

  Overall, despite minor differences in their objectives, the GloVe and LV-0 models – which perform only window sampling – behave similarly.
- **SVD:** As can be seen in fig. 3f, the SVD vector norms are invariant to rank. When moving from the LV-0 to LV-20 model, increasing the relative importance of negative information increases rank invariance of vector norms, and the SVD model which weighs negative information more heavily than any other model shows this effect to the extreme.

  Figure 4f shows less separation of word and context vectors than seen in the LV-20 and SG models. However, there is a clear separation in modes, with all word vector buckets having positive modes and all context vector buckets nearing zero. This is explained by the SVD model using the $PPMI$ transform, which drives dot products of negative cooccurring pairs to 0 rather than to negative values as with the $CPMI(-2)$ transform. Cosines of context vectors and the mean word vector are thus distributed near zero, rather than at negative values.

  Looking at table 5, the SVD model is significantly weaker on the SimLex

task than all other models. We attribute this to its indiscriminant weighting of reconstruction errors (not obeying the reliability principle). Weak results on RW are similarly attributed. Note that one might suspect the $PPMI$ transform to be at fault, but observe that in table 1 the $PPMI$ variant of LexVec is comparable to other transforms that do not collapse the negative distribution of PMI.

Under incorrect evaluation – which includes norm and premise exclusion strategies – SVD has the weakest performance on analogies of all models tested. However, if both strategies are excluded, it performs nearly as well as the LV-20 model on GSem$^{\text{-N-P}}$, and marginally worse on GSyn$^{\text{-N-P}}$ and MSR$^{\text{-N-P}}$. We attribute the weaker performance on syntactic analogies to the $PPMI$ metric (in table 1, $PPMI$ underperforms both $CPMI(-2)$ and $NNEGPMI$ on GSyn and MSR), and the strong performance *with strategy exclusion* to the majority weighting of negative information in the loss function.

Under qualitative analysis of nearest neighbors in tables 3 and 4, the SVD model returns semantically related words for both frequent and rare words, similar to the LV-20 model.

In summary, the SVD model, which accounts for negative information more strongly than any other model, magnifies the effects of negative information observed in the LexVec models.

7. Conclusions and Future Work

In this paper, we investigated the role that negative and positive information each play in distributional semantic models. We evaluated existing and novel ways of incorporating negative information into word embedding models based on explicit weighted matrix factorization. Results show that only accounting for positive PMI in the factorization strongly captures both semantics and syntax, whereas using only negative PMI captures some semantics but (surprisingly) a lot of syntactic information. Our findings indicate that *"a word is not only characterized by the company that it keeps, but also by the company it rejects"*.

Additionally, we investigated how increasing the relative importance of negative information affects the geometry of word embeddings. We observed that negative information improves rank invariance of vector geometry, with increase in word similarity and analogy task performance suggesting the importance of this invariance. We showed empirically that similar conclusions hold for the popular Skip-gram, GloVe, and SVD models. An important question for future work is why does word analogy performance *under correct evaluation* improve dramatically as negative sampling increases.

Finally, understanding the type of information captured by negative and positive PMI may also be relevant for studies on the role of negative indirect information for language acquisition. For example, Regier and Gahl (2004) argue that indirect negative evidence might play an important role in human acquisition of grammar, but do not link this idea to distributional semantics.

Acknowledgments

This research was partly supported by CAPES, CNPq, and EPSRC (projects 312114/2015-0, 423843/2016-8, 140402/2018-7, and EP/T02450X/1). There is no potential competing interest.

References

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., & Vollgraf, R. (2019, June). FLAIR: An easy-to-use framework for state-of-the-art NLP. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics (demonstrations)* (pp. 54–59). Minneapolis, Minnesota: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/N19-4010`

Allen, C., Balazevic, I., & Hospedales, T. (2019). What the vec? towards probabilistically grounded embeddings. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc. Retrieved from `https://proceedings.neurips.cc/paper/2019/file/23755432da68528f115c9633c0d7834f-Paper.pdf`

Allen, C., & Hospedales, T. M. (2019). Analogies explained: Towards understanding word embeddings. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (Vol. 97, pp. 223–231). PMLR. Retrieved from `http://proceedings.mlr.press/v97/allen19a.html`

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *TACL*, *5*, 135–146. Retrieved from `https://transacl.org/ojs/index.php/tacl/article/view/999`

Bolukbasi, T., Chang, K., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems 29: Annual conference on neural information processing systems* (pp. 4349–4357). Retrieved from `http://papers.nips.cc/paper/6228-man-is-to-computer-programmer-as-woman-is-to-homemaker-debiasing-word-embeddings`

Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, 31–40.

Bullinaria, J. A., & Levy, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, *39*(3), 510–526.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th international workshop on semantic evaluation* (pp. 1–14).

Church, K. W., & Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational linguistics*, *16*(1), 22–29.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., & Baroni, M. (2018). What you can cram into a single vector: Probing sentence embeddings for linguistic properties. *arXiv preprint arXiv:1805.01070*.

Curran, J. R., & Moens, M. (2002). Improvements in automatic thesaurus extraction. In *Proceedings of the acl-02 workshop on unsupervised lexical acquisition - volume 9* (pp. 59–66). Stroudsburg, PA, USA: Association for Computational Linguistics.

Deerwester, S. C., Dumais, S. T., Landauer, T. K., Furnas, G. W., & Harshman, R. A. (1990). Indexing by latent semantic analysis. *JAsIs*, *41*(6), 391–407.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, & T. Solorio (Eds.), *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)* (pp. 4171–4186). Association for Computational Linguistics.

Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, *1*(3), 211–218.

Ethayarajh, K., Duvenaud, D., & Hirst, G. (2019). Towards understanding linear word analogies. In A. Korhonen, D. R. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th conference of the association for computational linguistics, volume 1: Long papers* (pp. 3253–3262). Association for Computational Linguistics.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., & Ruppin,

E. (2001). Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on world wide web* (pp. 406–414).

Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Goldberg, Y., & Levy, O. (2014). word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*.

Gonen, H., & Goldberg, Y. (2019). Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In J. Burstein, C. Doran, & T. Solorio (Eds.), *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)* (pp. 609–614). Association for Computational Linguistics.

Harris, Z. S. (1954). Distributional structure. *Word*.

Hashimoto, T. B., Alvarez-Melis, D., & Jaakkola, T. S. (2016). Word embeddings as metric recovery in semantic spaces. *TACL*, *4*, 273–286. Retrieved from `https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/809`

Hill, F., Reichart, R., & Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, *41*(4), 665–695.

Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. *CoRR*, *abs/1508.01991*.

Jurafsky, D. (2000). *Speech & language processing*. Pearson Education India.

Kiela, D., & Clark, S. (2014, April). A systematic study of semantic vector space model parameters. In *Proceedings of the 2nd workshop on continuous vector space models and their compositionality (cvsc)* (pp. 21–30). Gothenburg, Sweden: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/W14-1503`

Lenci, A., Sahlgren, M., Jeuniaux, P., Gyllensten, A. C., & Miliani, M. (2021). A comprehensive comparative evaluation and analysis of distributional semantic models. *CoRR*, *abs/2105.09825*. Retrieved from `https://arxiv.org/abs/2105.09825`

Levy, O., & Goldberg, Y. (2014a, June). Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning* (pp. 171–180). Ann Arbor, Michigan: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/W14-1618`

Levy, O., & Goldberg, Y. (2014b). Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems* (pp. 2177–2185).

Levy, O., Goldberg, Y., & Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, *3*, 211–225.

Linzen, T. (2016). Issues in evaluating semantic spaces using word analogies. In *Proceedings of the 1st workshop on evaluating vector-space representations for nlp* (pp. 13–18). Association for Computational Linguistics.

Luong, T., Socher, R., & Manning, C. (2013, August). Better word representations with recursive neural networks for morphology. In *Proceedings of the seventeenth conference on computational natural language learning* (pp. 104–113). Sofia, Bulgaria: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/W13-3512`

Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Marcus, M. P., Marcinkiewicz, M. A., & Santorini, B. (1993, June). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, *19*(2), 313–330. Retrieved from `http://dl.acm.org/citation.cfm?id=972470.972475`

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111–3119).

Milajevs, D., Sadrzadeh, M., & Purver, M. (2016, August). Robust co-occurrence quantifi-

cation for lexical distributional semantics. In *Proceedings of the acl 2016 student research workshop* (pp. 58–64). Berlin, Germany: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/P16-3009

Mimno, D., & Thompson, L. (2017, September). The strange geometry of skip-gram with negative sampling. In *Proceedings of the 2017 conference on empirical methods in natural language processing* (pp. 2873–2878). Copenhagen, Denmark: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/D17-1308

Nissim, M., van Noord, R., & van der Goot, R. (2019). Fair is better than sensational: Man is to doctor as woman is to doctor. *CoRR*, *abs/1905.09866*. Retrieved from http://arxiv.org/abs/1905.09866

Pennington, J., Socher, R., & Manning, C. (2014, October). GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing* (pp. 1532–1543). Doha, Qatar: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/D14-1162

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In M. A. Walker, H. Ji, & A. Stent (Eds.), *Proceedings of the 2018 conference of the north american chapter of the association for computational linguistics: Human language technologies, volume 1 (long papers)* (pp. 2227–2237). Association for Computational Linguistics.

Polajnar, T., & Clark, S. (2014, April). Improving distributional semantic vectors through context selection and normalisation. In *Proceedings of the 14th conference of the european chapter of the association for computational linguistics* (pp. 230–238). Gothenburg, Sweden: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/E14-1025

Regier, T., & Gahl, S. (2004). Learning the unlearnable: The role of missing evidence. *Cognition*, *93*(2), 147–155.

Rogers, A., Drozd, A., & Li, B. (2017, August). The (too many) problems of analogical reasoning with word vectors. In *Proceedings of the 6th joint conference on lexical and computational semantics* (pp. 135–148). Vancouver, Canada: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/S17-1017

Salle, A., Idiart, M., & Villavicencio, A. (2016). Enhancing the lexvec distributed word representation model using positional contexts and external memory. *CoRR*, *abs/1606.01283*. Retrieved from http://arxiv.org/abs/1606.01283

Salle, A., & Villavicencio, A. (2018, June). Incorporating subword information into matrix factorization word embeddings. In *Proceedings of the second workshop on subword/character LEvel models* (pp. 66–71). New Orleans: Association for Computational Linguistics. Retrieved from https://aclanthology.org/W18-1209

Salle, A., Villavicencio, A., & Idiart, M. (2016, August). Matrix factorization using window sampling and negative sampling for improved word representations. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 419–424). Berlin, Germany: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/P16-2068

Schluter, N. (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)* (pp. 242–246).

Schütze, H. (1993). Word space. In *Advances in neural information processing systems* (pp. 895–902).

Shazeer, N., Doherty, R., Evans, C., & Waterson, C. (2016). Swivel: Improving embeddings by noticing what's missing. *arXiv preprint arXiv:1602.02215*.

Turney, P. D., & Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, *21*(4), 315–346.

Turney, P. D., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, *37*, 141–188.

Xin, X., Yuan, F., He, X., & Jose, J. M. (2018, July). Batch IS NOT heavy: Learning word representations from all samples. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)* (pp. 1853–1862). Melbourne, Australia: Association for Computational Linguistics. Retrieved from https://www.aclweb.org/anthology/P18-1172

Appendix A. Dataset Rank Statistics

Table A1. Percentile rank statistics for words in the datasets used in this paper. Note: μ- mean, σ- standard deviation, Qk- k-th quartile.

dataset	μ	σ	min	Q1	median	Q3	max
Rare Words (RW)	19.3	24.4	0.0	1.8	7.9	27.8	100.0
SimLex	2.3	3.1	0.0	0.6	1.5	3.0	46.4
WordSim-Relatedness (WSRel)	1.6	2.3	0.0	0.3	0.8	1.9	22.4
WordSim-Similarity (WSSim)	2.0	2.9	0.0	0.3	0.7	2.3	21.8
MSR Syntactic Analogies (MSR)	5.2	13.2	0.0	0.3	0.7	2.5	97.0
Google Semantic Analogies (GSem)	10.4	15.3	0.0	0.5	2.8	13.5	69.2
Google Syntactic Analogies (GSyn)	3.7	5.2	0.0	0.7	1.8	4.6	38.4
Google Semantic Analogies, Frequent Split (GSemF)	0.9	0.6	0.0	0.3	0.7	1.3	2.3
Google Syntactic Analogies, Frequent Split (GSynF)	0.4	0.3	0.0	0.2	0.3	0.6	1.0
Google Semantic Analogies, Rare Split (GSemR)	6.1	9.3	0.0	1.1	2.7	6.1	69.2
Google Syntactic Analogies, Rare Split(GSynR)	3.5	5.3	0.0	0.5	1.5	3.6	38.4
Penn Treebank (POS)	12.3	16.5	0.0	2.2	6.1	15.0	99.8
Tree Depth (Dep)	14.4	17.5	0.0	2.8	7.8	18.8	99.6
Top Constituent (TopC)	14.1	17.2	0.0	2.8	7.6	18.2	99.9

Appendix B. Fixed Window Size $l = 2$ and Positional Contexts

We performed identical experiments to the main paper, but using positional contexts and fixed window size of 2, as used in Salle, Idiart, and Villavicencio (2016). Results lead to matching conclusions as those for the larger randomized windows used in the main paper, and further highlight the role negative information: accounting for the zero cooccurrence (pairs in mnPMI) is even more important when using this smaller window size and positional contexts, which increase the sparsity of the cooccurrence matrix. Under this increased sparsity, models which ignore pairs in mnPMI (such as GloVe and LexVec with no negative sampling) see severe degradation in rare word representations.

Appendix C. Subword Information

Here we repeat the experiments from appendix B, but incorporate subword information into LexVec (Subword LexVec; SLV) and Skip-gram (fastText; FT). Results follow the same trend of the main paper, leading to matching conclusions. However, note that whereas in the main paper and appendix B we are able to isolate the effects of negative information on words of different frequencies, using subword information breaks this

Table B1. Same as table 2, but using **positional contexts and symmetric context window of fixed size 2**.

Name	Set	Full	Hist	WS	NS
nPMI: Negative information	$\{(w,c) \mid PMI_{w,c} \leq 0\}$	99.75	14.53	12.44	85.42
pPMI: Positive information	$\{(w,c) \mid PMI_{w,c} > 0\}$	0.25	85.47	87.56	14.58
mnPMI: Maximally-negative information	$\{(w,c) \mid M_{w,c} = 0\}$	99.71	0.00	0.00	69.08
nPMI\mnPMI: Collapsed negative information under PPMI	$\{(w,c) \mid PMI_{w,c} \leq 0 \wedge M_{w,c} > 0\}$	0.04	14.53	12.44	16.34

isolation by sharing information between frequent and rare word forms. Nevertheless, despite this confounding factor, results follow a remarkably similar trend.

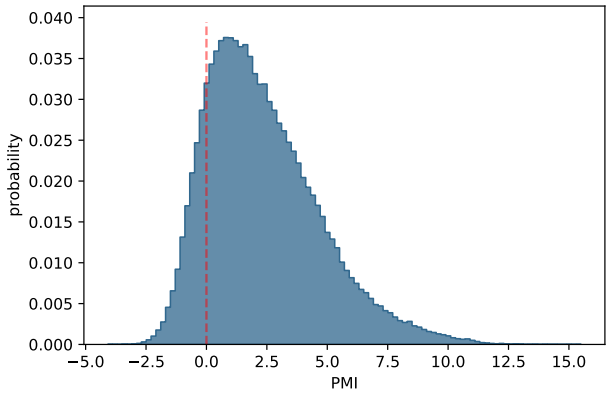


Figure B1. Same as fig. 1, but using **positional contexts and symmetric context window of fixed size 2**.

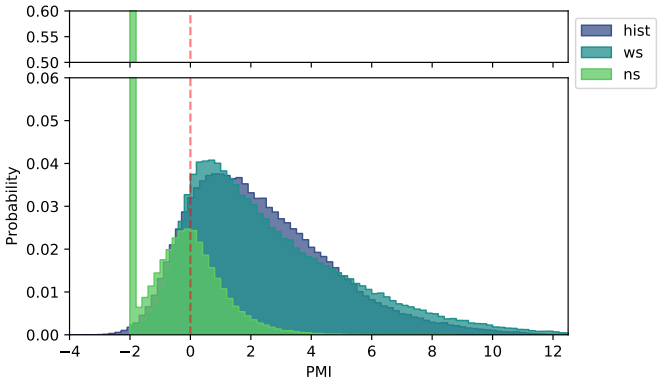
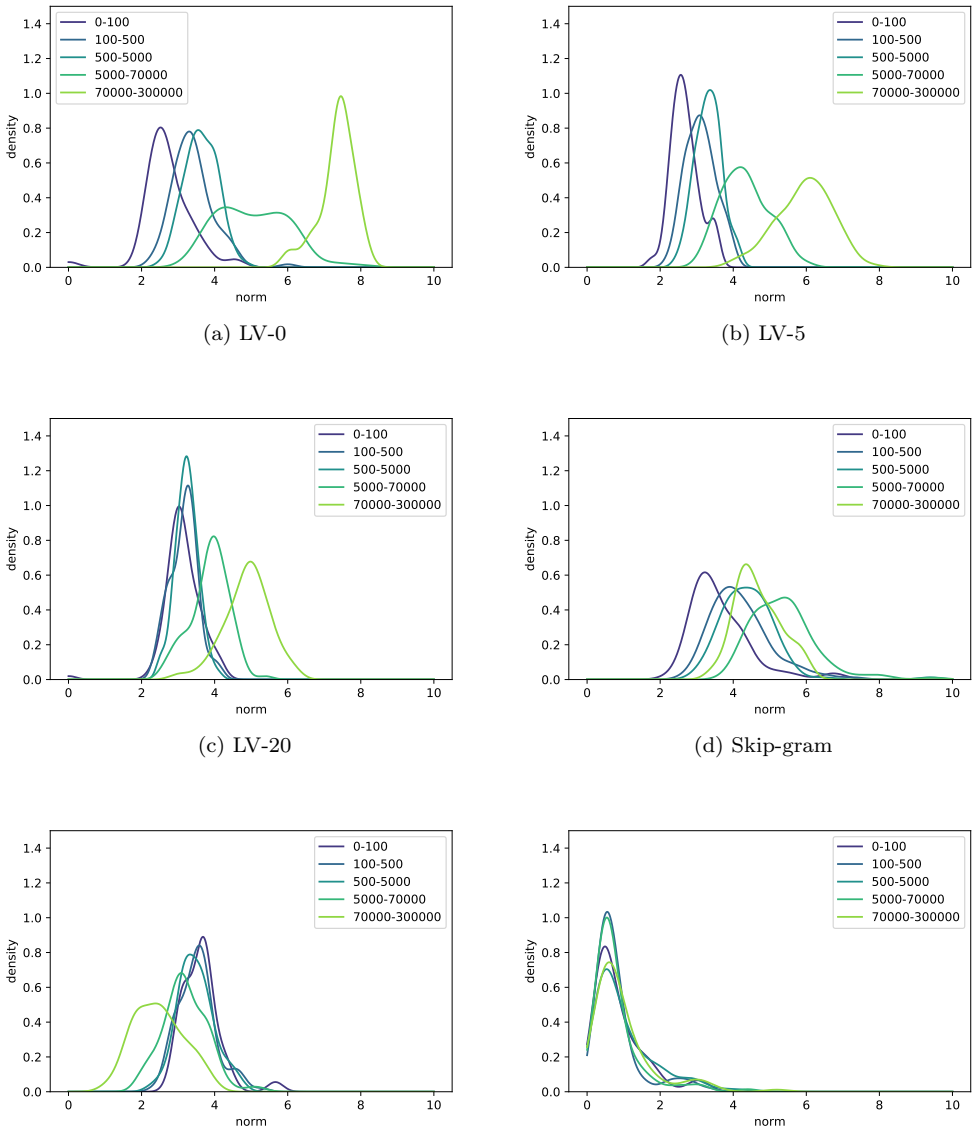


Figure B2. Same as fig. 2, but using **positional contexts and symmetric context window of fixed size 2**.

Table B2. Same as table 1, but using **positional contexts and symmetric context window of fixed size 2**.

model	SimLex	RW	GSem	STSB	GSyn	MSR	POS	Dep	TopC
pPPMI	**36.9**	34.1	52.8	63.3	47.4	34.0	92.3	31.7	33.7
nPPMI	1.2	-1.2	0.0	48.1	0.0	0.0	16.3	17.9	5.0
nCPMI(-2)	18.3	24.0	6.4	41.0	13.9	13.9	90.8	**32.8**	**35.3**
nNPMI	13.7	23.9	4.5	40.1	8.9	7.8	89.8	31.7	34.0
PPMI	36.6	**45.1**	79.5	**63.4**	61.3	45.6	92.4	27.5	30.1
CPMI(-2)	35.8	43.1	80.4	63.0	**65.2**	**51.8**	**92.5**	28.0	31.3
NPMI	32.5	43.6	62.4	57.1	57.4	44.8	92.4	29.4	31.7
NNEGPMI	36.2	43.5	**80.7**	63.3	63.6	49.5	92.4	27.8	30.1
Random	1.2	-1.2	0.0	45.3	0.0	0.0	16.3	17.9	5.0

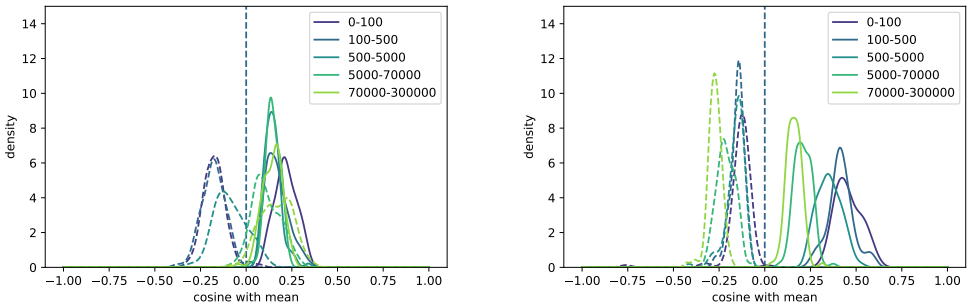


(a) LV-0 (b) LV-5

(c) LV-20 (d) Skip-gram

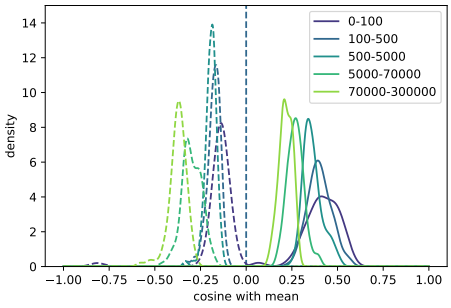
(e) GloVe (f) SVD

Figure B3. Same as fig. 3, but using **positional contexts and symmetric context window of fixed size 2**.

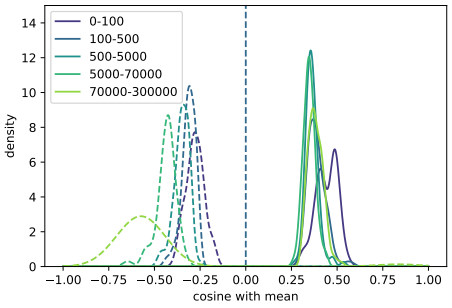


(a) LV-0

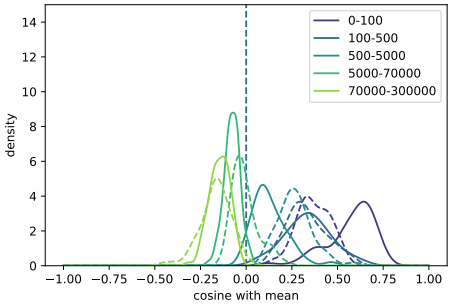
(b) LV-5



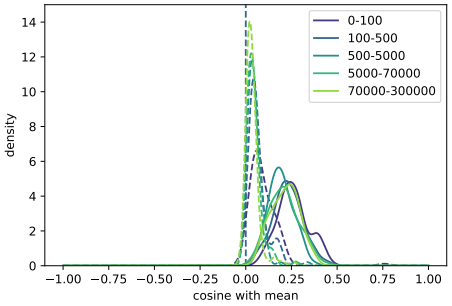
(c) LV-20



(d) Skip-gram

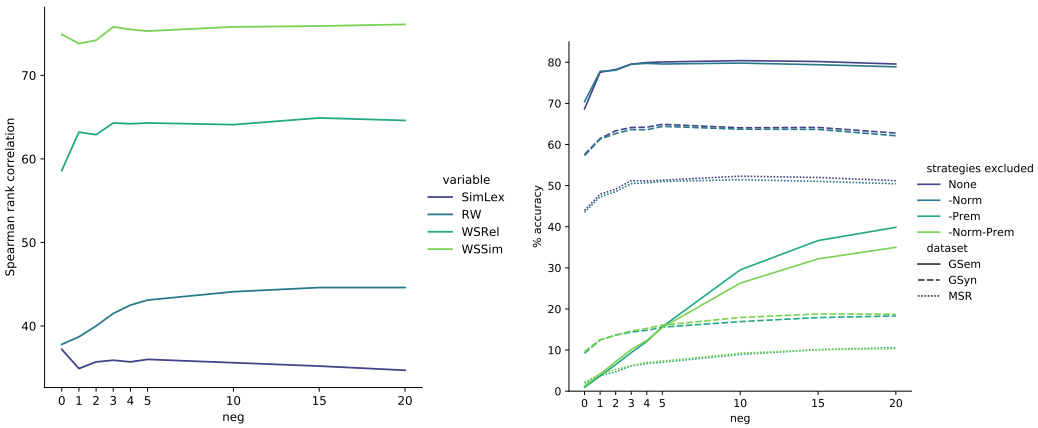


(e) GloVe



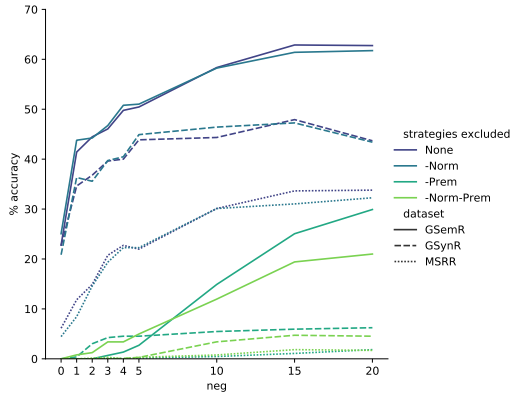
(f) SVD

Figure B4. Same as fig. 4, but using **positional contexts and symmetric context window of fixed size 2**.

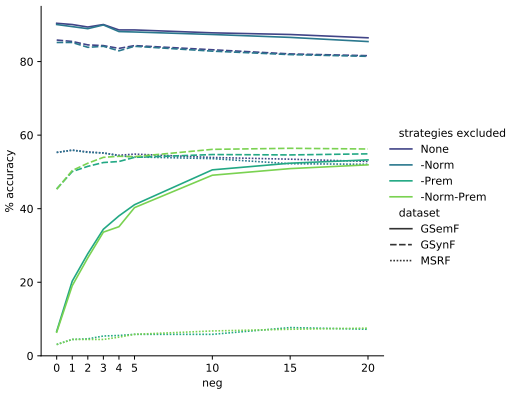


(a) Word similarity

(b) Analogies



(c) Analogies, rare words



(d) Analogies, frequent words

Figure B5. Same as fig. 5, but using **positional contexts and symmetric context window of fixed size 2**.

Table B3. Same as table 3, but using **positional contexts and symmetric context window of fixed size 2**.

word	model	neighbors
interest $_{\mathbf{0.3}}$	LV-0	interests $_{\mathbf{1.0}}$, conflict $_{\mathbf{0.5}}$, coi $_{\mathbf{1.2}}$, interested $_{\mathbf{0.7}}$, thegauntlet $_{\mathbf{78.0}}$, scientific $_{\mathbf{0.5}}$, passion $_{\mathbf{2.0}}$, attention $_{\mathbf{0.5}}$, concern $_{\mathbf{0.9}}$, activity $_{\mathbf{0.6}}$
	LV-20	interests $_{\mathbf{1.0}}$, interested $_{\mathbf{0.7}}$, conflict $_{\mathbf{0.5}}$, attention $_{\mathbf{0.5}}$, enthusiasm $_{\mathbf{3.9}}$, expertise $_{\mathbf{2.3}}$, concern $_{\mathbf{0.9}}$, appreciation $_{\mathbf{3.5}}$, scientific $_{\mathbf{0.5}}$, involvement $_{\mathbf{1.2}}$
	SG	interests $_{\mathbf{1.0}}$, intrest $_{\mathbf{50.0}}$, wizardimps $_{\mathbf{67.2}}$, interested $_{\mathbf{0.7}}$, richarddawkins $_{\mathbf{74.5}}$, thegauntlet $_{\mathbf{78.0}}$, interst $_{\mathbf{77.4}}$, bluntsde $_{\mathbf{91.6}}$, conflict $_{\mathbf{0.5}}$, bizjournals $_{\mathbf{42.8}}$
	GloVe	interests $_{\mathbf{1.0}}$, interested $_{\mathbf{0.7}}$, conflict $_{\mathbf{0.5}}$, attention $_{\mathbf{0.5}}$, influence $_{\mathbf{0.6}}$, concern $_{\mathbf{0.9}}$, expertise $_{\mathbf{2.3}}$, involvement $_{\mathbf{1.2}}$, popularity $_{\mathbf{1.2}}$, passion $_{\mathbf{2.0}}$
	SVD	interests $_{\mathbf{1.0}}$, attention $_{\mathbf{0.5}}$, interested $_{\mathbf{0.7}}$, importance $_{\mathbf{0.4}}$, profits $_{\mathbf{2.9}}$, debt $_{\mathbf{1.6}}$, expertise $_{\mathbf{2.3}}$, benefit $_{\mathbf{1.0}}$, contributions $_{\mathbf{0.5}}$, contribution $_{\mathbf{1.0}}$
cup $_{\mathbf{0.1}}$	LV-0	championship $_{\mathbf{0.2}}$, league $_{\mathbf{0.1}}$, champions $_{\mathbf{0.5}}$, cups $_{\mathbf{2.7}}$, finals $_{\mathbf{0.7}}$, trophy $_{\mathbf{1.1}}$, uefa $_{\mathbf{1.3}}$, final $_{\mathbf{0.1}}$, tournament $_{\mathbf{0.4}}$, championships $_{\mathbf{0.3}}$
	LV-20	champions $_{\mathbf{0.5}}$, cups $_{\mathbf{2.7}}$, championship $_{\mathbf{0.2}}$, finals $_{\mathbf{0.7}}$, trophy $_{\mathbf{1.1}}$, league $_{\mathbf{0.1}}$, competitions $_{\mathbf{0.9}}$, uefa $_{\mathbf{1.3}}$, tournament $_{\mathbf{0.4}}$, runners $_{\mathbf{1.5}}$
	SG	cups $_{\mathbf{2.7}}$, trophy $_{\mathbf{1.1}}$, championship $_{\mathbf{0.2}}$, champions $_{\mathbf{0.5}}$, finals $_{\mathbf{0.7}}$, supercup $_{\mathbf{8.5}}$, pokal $_{\mathbf{9.4}}$, championships $_{\mathbf{0.3}}$, uhrencup $_{\mathbf{90.3}}$, uefa $_{\mathbf{1.3}}$
	GloVe	championship $_{\mathbf{0.2}}$, champions $_{\mathbf{0.5}}$, finals $_{\mathbf{0.7}}$, cups $_{\mathbf{2.7}}$, league $_{\mathbf{0.1}}$, trophy $_{\mathbf{1.1}}$, uefa $_{\mathbf{1.3}}$, tournament $_{\mathbf{0.4}}$, final $_{\mathbf{0.1}}$, fifa $_{\mathbf{1.1}}$
	SVD	runners $_{\mathbf{1.5}}$, nextseason $_{\mathbf{4.9}}$, champions $_{\mathbf{0.5}}$, cups $_{\mathbf{2.7}}$, trophy $_{\mathbf{1.1}}$, scorers $_{\mathbf{4.7}}$, competitions $_{\mathbf{0.9}}$, matches $_{\mathbf{0.4}}$, squad $_{\mathbf{0.7}}$, fifa $_{\mathbf{1.1}}$
soul $_{\mathbf{0.9}}$	LV-0	souls $_{\mathbf{3.0}}$, blues $_{\mathbf{0.8}}$, funk $_{\mathbf{2.7}}$, gospel $_{\mathbf{1.6}}$, hop $_{\mathbf{1.1}}$, mind $_{\mathbf{0.5}}$, spirit $_{\mathbf{0.8}}$, reggae $_{\mathbf{3.4}}$, jazz $_{\mathbf{0.6}}$, rap $_{\mathbf{2.1}}$
	LV-20	funk $_{\mathbf{2.7}}$, blues $_{\mathbf{0.8}}$, heaven $_{\mathbf{1.5}}$, essence $_{\mathbf{3.0}}$, souls $_{\mathbf{3.0}}$, mind $_{\mathbf{0.5}}$, spirit $_{\mathbf{0.8}}$, love $_{\mathbf{0.2}}$, dreams $_{\mathbf{1.6}}$, jazz $_{\mathbf{0.6}}$
	SG	funk $_{\mathbf{2.7}}$, souls $_{\mathbf{3.0}}$, blues $_{\mathbf{0.8}}$, soulful $_{\mathbf{11.0}}$, seekerz $_{\mathbf{81.3}}$, changeless $_{\mathbf{89.4}}$, makossa $_{\mathbf{58.0}}$, spirit $_{\mathbf{0.8}}$, essence $_{\mathbf{3.0}}$, jazzmatazz $_{\mathbf{98.9}}$
	GloVe	blues $_{\mathbf{0.8}}$, funk $_{\mathbf{2.7}}$, souls $_{\mathbf{3.0}}$, spirit $_{\mathbf{0.8}}$, mind $_{\mathbf{0.5}}$, gospel $_{\mathbf{1.6}}$, hop $_{\mathbf{1.1}}$, pop $_{\mathbf{0.4}}$, jazz $_{\mathbf{0.6}}$, love $_{\mathbf{0.2}}$
	SVD	heaven $_{\mathbf{1.5}}$, eternal $_{\mathbf{2.5}}$, forever $_{\mathbf{1.7}}$, dreams $_{\mathbf{1.6}}$, dream $_{\mathbf{0.9}}$, funk $_{\mathbf{2.7}}$, love $_{\mathbf{0.2}}$, souls $_{\mathbf{3.0}}$, spirit $_{\mathbf{0.8}}$, blues $_{\mathbf{0.8}}$

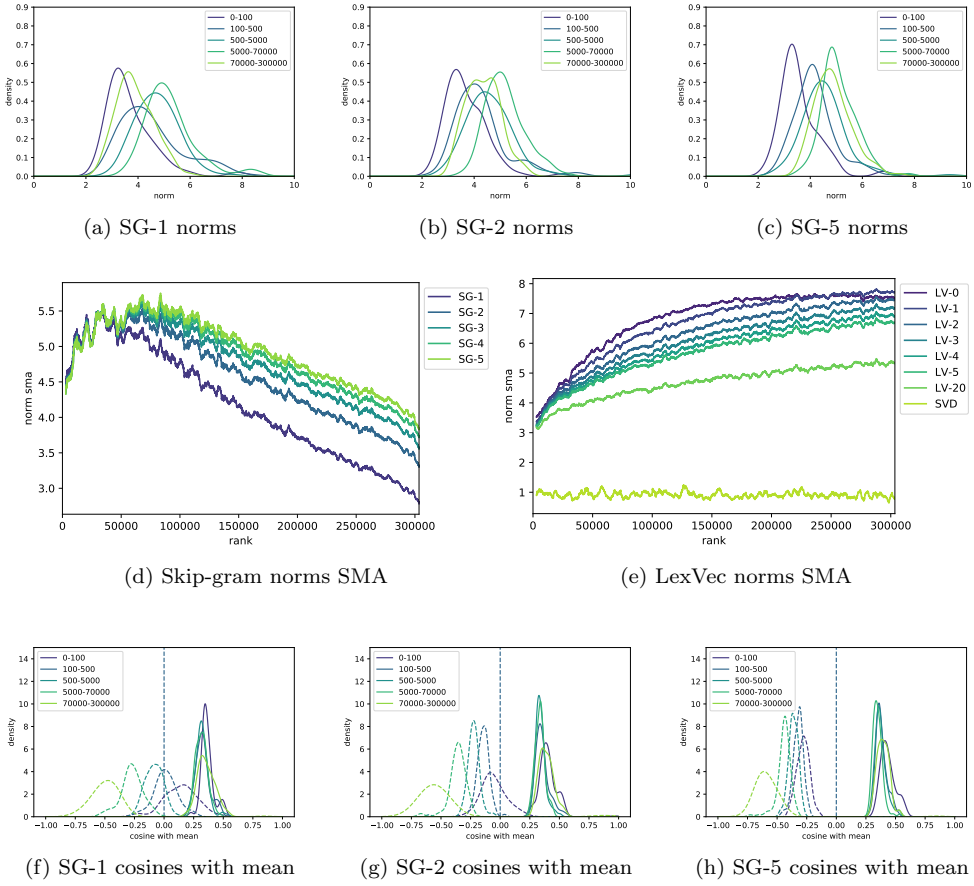
Table B4. Same as table 4, but using **positional contexts and symmetric context window of fixed size 2**.

word	model	neighbors
rooters 77	LV-0	bajir 66, ravimy 87, hamerkop 83, argyrodes 68, thaumasia 90, brooklin 43, karuma 99, sorlle 86, valise 69, roaring 7
	LV-20	howled 73, cheering 11, roars 27, bosox 81, monumentals 68, cheered 15, nuxhall 73, booing 30, phillie 56, atlantics 26
	SG	cheering 11, bosox 81, howled 73, fans 0, lynah 86, cheered 15, landrith 57, clendenon 85, phillie 56, semipro 60
	GloVe	specifc 84, unserious 99, cusumano 87, whch 65, uninviting 91, overexcited 99, imnsho 89, preffered 91, alread 94, untypical 94
	SVD	ebbets 29, comiskey 18, ballplayers 31, batboy 62, crawfords 37, chisox 83, bosox 81, gothams 59, semipro 60, krichell 79
monocultures 76	LV-0	fieldensis 87, cantillans 87, berbers 13, ritsema 57, boutonii 76, tmutarakan 74, shuhada 67, chisocheton 74, poepp 77, approvals 12
	LV-20	monoculture 35, seedlings 11, saplings 24, conifers 13, hardwoods 19, crops 2, agroforestry 34, cultivations 64, understory 16, broadleaved 47
	SG	monoculture 35, polyculture 88, overgrazed 81, fuelwood 62, intercropping 90, rainfed 65, overharvesting 83, cucurbits 81, croplands 60, silvicultural 84
	GloVe	hereabouts 74, upend 97, controvertial 87, enlivening 79, pluralisation 92, selfsame 98, herrod 85, overspend 83, unserious 99, liquify 89
	SVD	monoculture 35, cropland 27, windbreaks 68, replanted 27, orchards 6, silviculture 45, plantations 3, intercropping 90, cultivations 64, seedlings 11
flighted 83	LV-0	uproot 34, okumoto 96, ratcheted 94, revealer 79, pandey 10, flagpole 15, halvard 66, bersetzungsstufen 74, swiftest 84, stairlift 86
	LV-20	feathered 11, flightless 14, quadrupedal 30, ratites 35, bipedal 16, raptorial 60, necked 6, beak 6, beaks 13, prehensile 26
	SG	digitigrade 84, plantigrade 84, zygodactyl 57, raptorial 60, woodcreepers 94, chelae 81, pronated 92, forelegs 30, apomorphic 93, stockier 86
	GloVe	wikispeak 87, similary 82, vandelism 97, specifc 84, validations 76, demagogic 77, imnsho 89, incentivise 100, smidge 97, geneological 98
	SVD	flightless 14, shoebill 71, tinamous 38, corvid 76, curassows 78, pratincoles 45, toucans 40, turacos 56, hoatzin 65, anseriformes 34

Table B5. Same as table 5, but using **positional contexts and symmetric context window of fixed size 2**.

model	SimLex	RW	GSem	GSem^{-N}	GSem^{-P}	GSem^{-N-P}
LV-0	37.2	37.8	68.7	70.4	0.9	1.3
LV-5	36.0	43.1	**80.1**	79.6	15.7	15.5
LV-20	34.7	44.6	79.6	78.9	**39.9**	**35.0**
SG-1	38.5	46.4	74.7	74.3	18.0	17.8
SG-2	39.0	47.7	77.5	77.5	19.3	18.6
SG-5	**39.4**	**48.8**	79.8	**79.9**	18.9	18.2
GloVe	35.2	36.1	74.8	73.0	3.0	3.5
SVD	31.6	44.0	48.5	41.1	30.9	26.8

model	GSyn	GSyn^{-N}	GSyn^{-P}	GSyn^{-N-P}	MSR	MSR^{-N}	MSR^{-P}	MSR^{-N-P}
LV-0	57.6	57.3	9.2	9.6	44.0	43.5	2.0	2.3
LV-5	64.9	64.4	15.6	16.1	51.3	51.0	7.0	7.3
LV-20	62.8	62.1	**18.3**	**18.7**	51.2	50.5	**10.6**	**10.3**
SG-1	67.7	66.8	14.6	15.2	53.7	52.4	6.1	6.4
SG-2	**68.7**	**67.8**	14.8	15.6	54.8	53.5	6.0	6.7
SG-5	68.2	67.6	14.9	15.7	**56.0**	**54.7**	6.2	6.7
GloVe	59.2	58.2	9.7	9.7	47.5	45.7	2.8	3.2
SVD	49.3	46.5	17.1	15.8	37.3	31.5	9.9	8.5



(a) SG-1 norms (b) SG-2 norms (c) SG-5 norms

(d) Skip-gram norms SMA (e) LexVec norms SMA

(f) SG-1 cosines with mean (g) SG-2 cosines with mean (h) SG-5 cosines with mean

Figure B6. Same as fig. 6, but using **positional contexts and symmetric context window of fixed size 2**.

Table C1. Same as table 2, but using **subword information**, positional contexts and symmetric context window of fixed size 2.

Name	Set	Full	Hist	WS	NS
nPMI: Negative information	$\{(w,c) \mid PMI_{w,c} \leq 0\}$	99.75	14.53	12.44	85.42
pPMI: Positive information	$\{(w,c) \mid PMI_{w,c} > 0\}$	0.25	85.47	87.56	14.58
mnPMI: Maximally-negative information	$\{(w,c) \mid M_{w,c} = 0\}$	99.71	0.00	0.00	69.08
nPMI\mnPMI: Collapsed negative information under PPMI	$\{(w,c) \mid PMI_{w,c} \leq 0 \wedge M_{w,c} > 0\}$	0.04	14.53	12.44	16.34

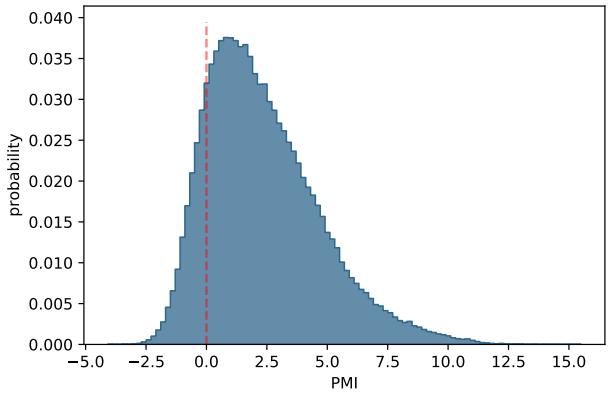


Figure C1. Same as fig. 1, but using **subword information**, positional contexts and symmetric context window of fixed size 2.

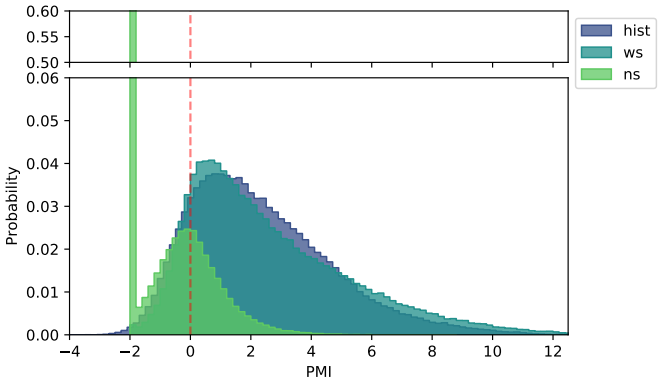
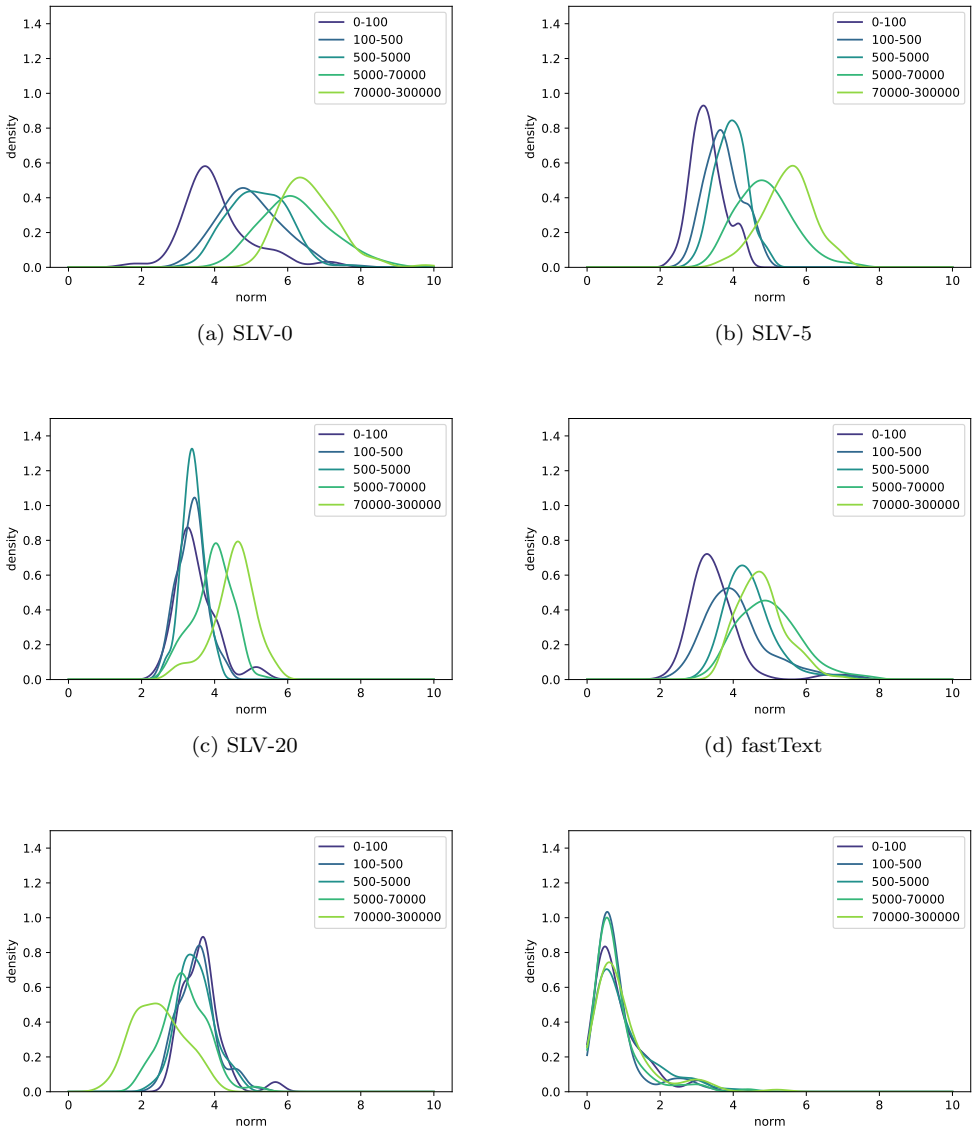


Figure C2. Same as fig. 2, but using **subword information**, positional contexts and symmetric context window of fixed size 2.

Table C2. Same as table 1, but using **subword information**, positional contexts and symmetric context window of fixed size 2.

model	SimLex	RW	GSem	STSB	GSyn	MSR	POS	Dep	TopC
pPPMI	33.9	37.3	21.9	**64.3**	51.4	41.1	92.5	30.6	34.2
nPPMI	2.8	13.5	0.2	30.8	26.1	24.1	16.3	17.9	5.0
nCPMI(-2)	18.2	25.6	2.9	39.9	14.3	13.5	91.1	**33.0**	**35.5**
nNPMI	13.8	26.5	1.2	38.8	21.5	24.2	90.9	32.4	34.5
PPMI	**38.1**	**51.4**	72.2	63.5	67.5	52.7	92.5	27.9	31.1
CPMI(-2)	37.1	49.6	**77.3**	64.1	71.7	59.5	**92.8**	29.0	32.6
NPMI	32.8	46.6	32.8	54.1	**74.0**	**62.5**	92.4	31.5	32.4
NNEGPMI	37.2	49.6	76.5	64.0	70.5	57.3	92.6	28.5	31.5
Random	2.9	13.5	0.2	30.1	26.6	24.6	16.3	17.9	5.0



(a) SLV-0

(b) SLV-5

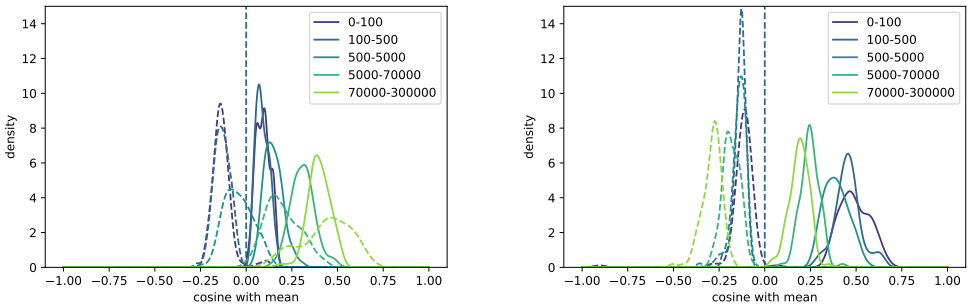
(c) SLV-20

(d) fastText

(e) GloVe

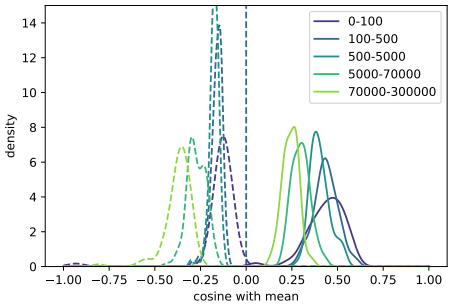
(f) SVD

Figure C3. Same as fig. 3, but using **subword information**, positional contexts and symmetric context window of fixed size 2.

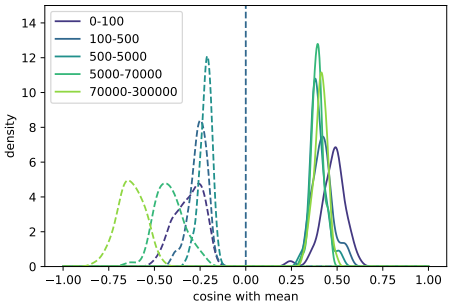


(a) SLV-0

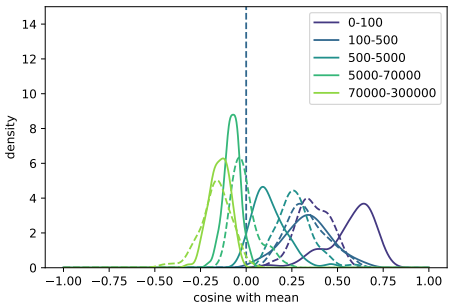
(b) SLV-5



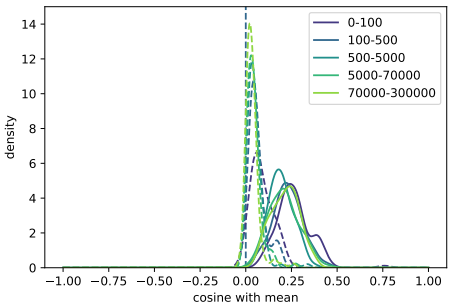
(c) SLV-20



(d) fastText

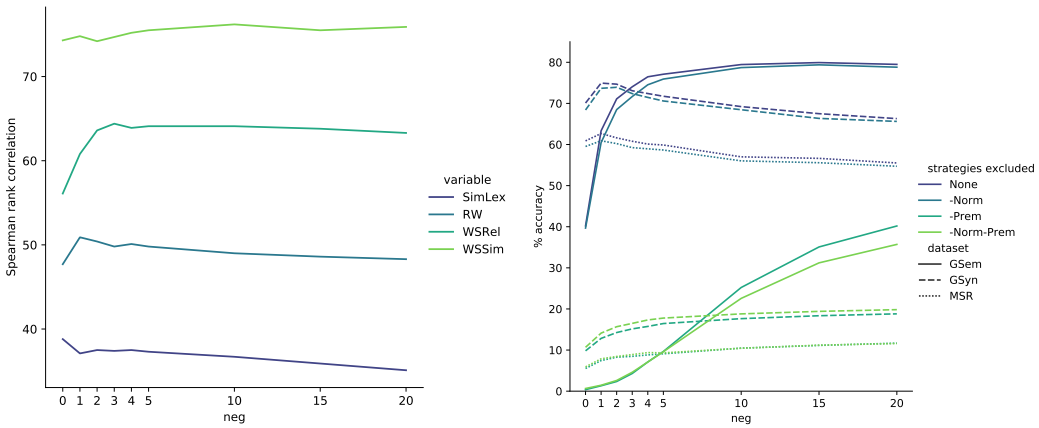


(e) GloVe



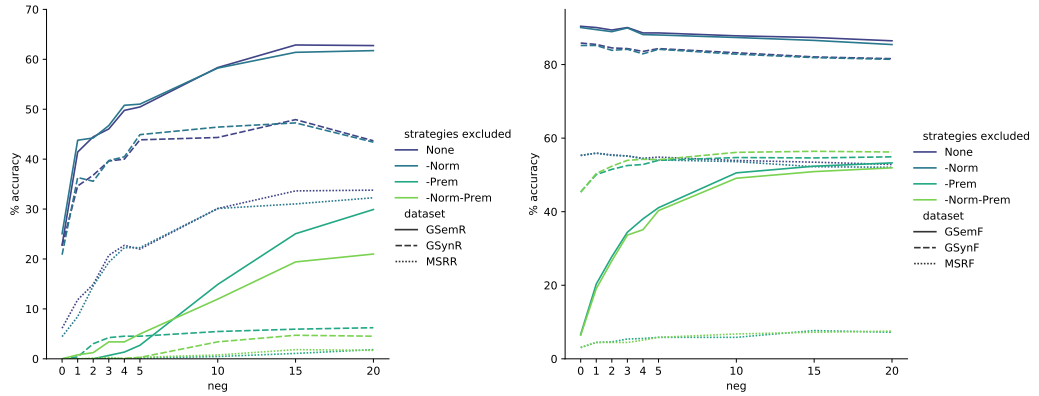
(f) SVD

Figure C4. Same as fig. 4, but using **subword information**, positional contexts and symmetric context window of fixed size 2.



(a) Word similarity

(b) Analogies



(c) Analogies, rare words

(d) Analogies, frequent words

Figure C5. Same as fig. 5, but using **subword information**, positional contexts and symmetric context window of fixed size 2.

Table C3. Same as table 3, but using **subword information**, positional contexts and symmetric context window of fixed size 2.

word	model	neighbors
interest $_{\mathbf{0.3}}$	SLV-0	pinterest $_{\mathbf{29.2}}$, interests $_{\mathbf{1.0}}$, interested $_{\mathbf{0.7}}$, disinterest $_{\mathbf{26.1}}$, conflict $_{\mathbf{0.5}}$, stuffofinterest $_{\mathbf{25.6}}$, interesse $_{\mathbf{66.5}}$, bitterest $_{\mathbf{51.3}}$, merest $_{\mathbf{63.5}}$, chemicalinterest $_{\mathbf{23.2}}$
	SLV-20	interests $_{\mathbf{1.0}}$, interested $_{\mathbf{0.7}}$, conflict $_{\mathbf{0.5}}$, attention $_{\mathbf{0.5}}$, expertise $_{\mathbf{2.3}}$, involvement $_{\mathbf{1.2}}$, enthusiasm $_{\mathbf{3.9}}$, disinterest $_{\mathbf{26.1}}$, concern $_{\mathbf{0.9}}$, appreciation $_{\mathbf{3.5}}$
	FT	interests $_{\mathbf{1.0}}$, disinterest $_{\mathbf{26.1}}$, interested $_{\mathbf{0.7}}$, pinterest $_{\mathbf{29.2}}$, enthusiasm $_{\mathbf{3.9}}$, fascination $_{\mathbf{7.0}}$, attention $_{\mathbf{0.5}}$, enthusiasms $_{\mathbf{56.1}}$, disinterestedness $_{\mathbf{98.5}}$, intrest $_{\mathbf{50.0}}$
	GloVe	interests $_{\mathbf{1.0}}$, interested $_{\mathbf{0.7}}$, conflict $_{\mathbf{0.5}}$, attention $_{\mathbf{0.5}}$, influence $_{\mathbf{0.6}}$, concern $_{\mathbf{0.9}}$, expertise $_{\mathbf{2.3}}$, involvement $_{\mathbf{1.2}}$, popularity $_{\mathbf{1.2}}$, passion $_{\mathbf{2.0}}$
	SVD	interests $_{\mathbf{1.0}}$, attention $_{\mathbf{0.5}}$, interested $_{\mathbf{0.7}}$, importance $_{\mathbf{0.4}}$, profits $_{\mathbf{2.9}}$, debt $_{\mathbf{1.6}}$, expertise $_{\mathbf{2.3}}$, benefit $_{\mathbf{1.0}}$, contributions $_{\mathbf{0.5}}$, contribution $_{\mathbf{1.0}}$
cup $_{\mathbf{0.1}}$	SLV-0	cups $_{\mathbf{2.7}}$, championship $_{\mathbf{0.2}}$, champions $_{\mathbf{0.5}}$, cupfb $_{\mathbf{64.7}}$, trophy $_{\mathbf{1.1}}$, cupen $_{\mathbf{17.1}}$, cupa $_{\mathbf{23.7}}$, uefa $_{\mathbf{1.3}}$, league $_{\mathbf{0.1}}$, championships $_{\mathbf{0.3}}$
	SLV-20	cups $_{\mathbf{2.7}}$, champions $_{\mathbf{0.5}}$, championship $_{\mathbf{0.2}}$, finals $_{\mathbf{0.7}}$, trophy $_{\mathbf{1.1}}$, league $_{\mathbf{0.1}}$, competitions $_{\mathbf{0.9}}$, qualifiers $_{\mathbf{4.7}}$, tournament $_{\mathbf{0.4}}$, runners $_{\mathbf{1.5}}$
	FT	cups $_{\mathbf{2.7}}$, championship $_{\mathbf{0.2}}$, supercups $_{\mathbf{45.3}}$, supercup $_{\mathbf{8.5}}$, trophy $_{\mathbf{1.1}}$, champions $_{\mathbf{0.5}}$, finals $_{\mathbf{0.7}}$, supercupen $_{\mathbf{55.0}}$, uhrencup $_{\mathbf{90.3}}$, pokal $_{\mathbf{9.4}}$
	GloVe	championship $_{\mathbf{0.2}}$, champions $_{\mathbf{0.5}}$, finals $_{\mathbf{0.7}}$, cups $_{\mathbf{2.7}}$, league $_{\mathbf{0.1}}$, trophy $_{\mathbf{1.1}}$, uefa $_{\mathbf{1.3}}$, tournament $_{\mathbf{0.4}}$, final $_{\mathbf{0.1}}$, fifa $_{\mathbf{1.1}}$
	SVD	runners $_{\mathbf{1.5}}$, nextseason $_{\mathbf{4.9}}$, champions $_{\mathbf{0.5}}$, cups $_{\mathbf{2.7}}$, trophy $_{\mathbf{1.1}}$, scorers $_{\mathbf{4.7}}$, competitions $_{\mathbf{0.9}}$, matches $_{\mathbf{0.4}}$, squad $_{\mathbf{0.7}}$, fifa $_{\mathbf{1.1}}$
soul $_{\mathbf{0.9}}$	SLV-0	souls $_{\mathbf{3.0}}$, soule $_{\mathbf{16.8}}$, souli $_{\mathbf{48.4}}$, nsoul $_{\mathbf{52.5}}$, sould $_{\mathbf{40.0}}$, souled $_{\mathbf{37.4}}$, soulchild $_{\mathbf{51.7}}$, soulive $_{\mathbf{72.5}}$, soulful $_{\mathbf{11.0}}$, soult $_{\mathbf{18.6}}$
	SLV-20	funk $_{\mathbf{2.7}}$, blues $_{\mathbf{0.8}}$, souls $_{\mathbf{3.0}}$, heaven $_{\mathbf{1.5}}$, essence $_{\mathbf{3.0}}$, spirit $_{\mathbf{0.8}}$, mind $_{\mathbf{0.5}}$, love $_{\mathbf{0.2}}$, jazz $_{\mathbf{0.6}}$, reggae $_{\mathbf{3.4}}$
	FT	souls $_{\mathbf{3.0}}$, funk $_{\mathbf{2.7}}$, soulchild $_{\mathbf{51.7}}$, soulful $_{\mathbf{11.0}}$, blues $_{\mathbf{0.8}}$, temptations $_{\mathbf{9.3}}$, soulfulness $_{\mathbf{97.6}}$, salsoul $_{\mathbf{42.7}}$, soulmates $_{\mathbf{48.9}}$, reggae $_{\mathbf{3.4}}$
	GloVe	blues $_{\mathbf{0.8}}$, funk $_{\mathbf{2.7}}$, souls $_{\mathbf{3.0}}$, spirit $_{\mathbf{0.8}}$, mind $_{\mathbf{0.5}}$, gospel $_{\mathbf{1.6}}$, hop $_{\mathbf{1.1}}$, pop $_{\mathbf{0.4}}$, jazz $_{\mathbf{0.6}}$, love $_{\mathbf{0.2}}$
	SVD	heaven $_{\mathbf{1.5}}$, eternal $_{\mathbf{2.5}}$, forever $_{\mathbf{1.7}}$, dreams $_{\mathbf{1.6}}$, dream $_{\mathbf{0.9}}$, funk $_{\mathbf{2.7}}$, love $_{\mathbf{0.2}}$, souls $_{\mathbf{3.0}}$, spirit $_{\mathbf{0.8}}$, blues $_{\mathbf{0.8}}$

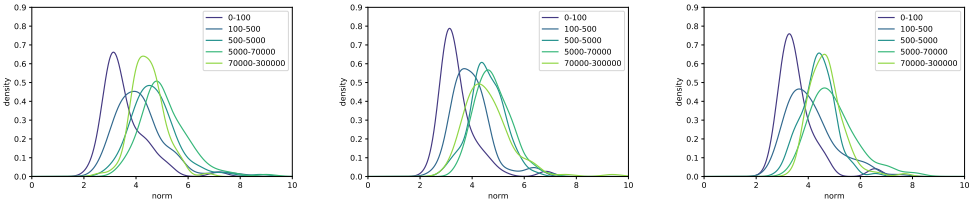
Table C4. Same as table 4, but using **subword information**, positional contexts and symmetric context window of fixed size 2.

word	model	neighbors
rooters $_{77}$	SLV-0	cooters $_{97}$, looters $_{21}$, booters $_{75}$, rooter $_{61}$, hooters $_{18}$, footers $_{25}$, scooters $_{15}$, freebooters $_{74}$, rootes $_{24}$, troubleshooters $_{55}$
	SLV-20	rooter $_{61}$, cooters $_{97}$, booters $_{75}$, looters $_{21}$, truckers $_{16}$, revellers $_{44}$, stoners $_{46}$, greasers $_{63}$, tuckers $_{72}$, stokers $_{40}$
	FT	rooter $_{61}$, cooters $_{97}$, hooters $_{18}$, booters $_{75}$, looters $_{21}$, footers $_{25}$, rootes $_{24}$, crosscutters $_{63}$, roaders $_{91}$, freebooters $_{74}$
	GloVe	specifc $_{84}$, unserious $_{99}$, cusumano $_{87}$, whch $_{65}$, uninviting $_{91}$, overexcited $_{99}$, imnsho $_{89}$, preffered $_{91}$, alread $_{94}$, untypical $_{94}$
	SVD	ebbets $_{29}$, comiskey $_{18}$, ballplayers $_{31}$, batboy $_{62}$, crawfords $_{37}$, chisox $_{83}$, bosox $_{81}$, gothams $_{59}$, semipro $_{60}$, krichell $_{79}$
monocultures $_{76}$	SLV-0	monoculture $_{35}$, monocular $_{38}$, protoculture $_{61}$, uncultured $_{51}$, ethnocultural $_{59}$, monocarpic $_{97}$, monocoupe $_{73}$, cultureel $_{99}$, culturing $_{34}$, polyculture $_{88}$
	SLV-20	monoculture $_{35}$, polyculture $_{88}$, cultivations $_{64}$, crops $_{2}$, intercropping $_{90}$, agroforestry $_{34}$, cultivation $_{3}$, silviculture $_{45}$, seedlings $_{11}$, clearcutting $_{47}$
	FT	monoculture $_{35}$, polyculture $_{88}$, ethnocultural $_{59}$, silviculture $_{45}$, cultivations $_{64}$, crops $_{2}$, ecotypes $_{58}$, intercropping $_{90}$, cultures $_{1}$, overgrazing $_{26}$
	GloVe	hereabouts $_{74}$, upend $_{97}$, controvertial $_{87}$, enlivening $_{79}$, pluralisation $_{92}$, selfsame $_{98}$, herrod $_{85}$, overspend $_{83}$, unserious $_{99}$, liquify $_{89}$
	SVD	monoculture $_{35}$, cropland $_{27}$, windbreaks $_{68}$, replanted $_{27}$, orchards $_{6}$, silviculture $_{45}$, plantations $_{3}$, intercropping $_{90}$, cultivations $_{64}$, seedlings $_{11}$
flighted $_{83}$	SLV-0	alighted $_{52}$, flighty $_{43}$, lighted $_{9}$, slighted $_{25}$, flightplan $_{93}$, unlighted $_{83}$, benighted $_{57}$, flightaware $_{86}$, flightdeck $_{80}$, blighted $_{20}$
	SLV-20	flightdeck $_{80}$, flights $_{1}$, flight $_{0}$, flighty $_{43}$, flightless $_{14}$, taxiing $_{19}$, taxied $_{61}$, flightaware $_{86}$, flown $_{2}$, flightline $_{55}$
	FT	flight $_{0}$, flights $_{1}$, flighty $_{43}$, flightdeck $_{80}$, flightless $_{14}$, flightpath $_{67}$, flown $_{2}$, taxiing $_{19}$, alighted $_{52}$, taxied $_{61}$
	GloVe	wikispeak $_{87}$, similary $_{82}$, vandelism $_{97}$, specifc $_{84}$, validations $_{76}$, demagogic $_{77}$, imnsho $_{89}$, incentivise $_{100}$, smidge $_{97}$, geneological $_{98}$
	SVD	flightless $_{14}$, shoebill $_{71}$, tinamous $_{38}$, corvid $_{76}$, curassows $_{78}$, pratincoles $_{45}$, toucans $_{40}$, turacos $_{56}$, hoatzin $_{65}$, anseriformes $_{34}$

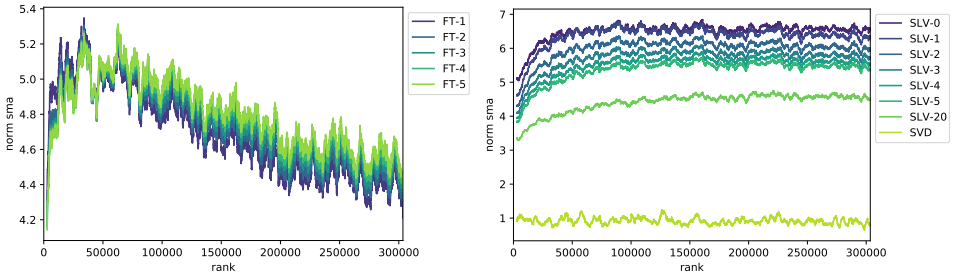
Table C5. Same as table 5, but using **subword information**, positional contexts and symmetric context window of fixed size 2.

model	SimLex	RW	GSem	GSem$^{\text{-N}}$	GSem$^{\text{-P}}$	GSem$^{\text{-N-P}}$
SLV-0	38.8	47.7	40.2	39.7	0.4	0.7
SLV-5	37.3	49.8	77.1	75.9	9.7	9.6
SLV-20	35.1	48.3	**79.5**	**78.8**	**40.2**	**35.7**
FT-1	38.7	51.2	70.4	70.1	18.8	18.7
FT-2	39.4	51.7	74.3	74.1	21.3	21.4
FT-5	**40.3**	**52.1**	78.8	78.5	22.1	21.2
GloVe	35.2	36.1	74.8	73.0	3.0	3.5
SVD	31.6	44.0	48.5	41.1	30.9	26.8

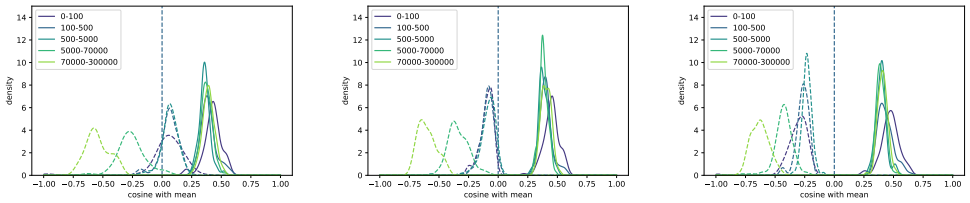
model	GSyn	GSyn$^{\text{-N}}$	GSyn$^{\text{-P}}$	GSyn$^{\text{-N-P}}$	MSR	MSR$^{\text{-N}}$	MSR$^{\text{-P}}$	MSR$^{\text{-N-P}}$
SLV-0	70.1	68.4	9.8	10.7	60.9	59.5	5.5	5.9
SLV-5	71.7	70.6	16.4	17.8	59.9	58.7	9.1	9.3
SLV-20	66.3	65.6	**18.8**	**19.8**	55.5	54.7	11.7	11.6
FT-1	**74.5**	**74.0**	17.8	19.3	61.2	60.2	**12.6**	**12.9**
FT-2	74.0	73.6	17.6	19.1	61.6	60.6	11.5	12.1
FT-5	74.2	73.6	17.4	19.0	**62.2**	**61.3**	11.2	11.7
GloVe	59.2	58.2	9.7	9.7	47.5	45.7	2.8	3.2
SVD	49.3	46.5	17.1	15.8	37.3	31.5	9.9	8.5



(a) FT-1 norms (b) FT-2 norms (c) FT-5 norms



(d) fastText norms SMA (e) Subword LexVec norms SMA



(f) FT-1 cosines with mean (g) FT-2 cosines with mean (h) FT-5 cosines with mean

Figure C6. Same as fig. 6, but using **subword information**, positional contexts and symmetric context window of fixed size 2.