Random Positive-Only Projections:
PPMI-Enabled Incremental Semantic Space Construction

Behrang QasemiZadeh
DFG SFB 991
Heinrich-Heine-Universität Düsseldorf
Düsseldorf, Germany
zadeh@phil.hhu.de

Laura Kallmeyer
DFG SFB 991
Heinrich-Heine-Universität Düsseldorf
Düsseldorf, Germany
kallmeyer@phil.hhu.de

Abstract

We introduce *positive-only projection* (PoP), a new algorithm for constructing semantic spaces and word embeddings. The PoP method employs random projections. Hence, it is highly scalable and computationally efficient. In contrast to previous methods that use random projection matrices $\mathbf{R}$ with the expected value of 0 (i.e., $E(\mathbf{R}) = 0$), the proposed method uses $\mathbf{R}$ with $E(\mathbf{R}) > 0$. We use *Kendall*'s τ_b correlation to compute vector similarities in the resulting non-Gaussian spaces. Most importantly, since $E(\mathbf{R}) > 0$, weighting methods such as positive pointwise mutual information (PPMI) can be applied to PoP-constructed spaces after their construction for efficiently transferring PoP embeddings onto spaces that are discriminative for semantic similarity assessments. Our PoP-constructed models, combined with PPMI, achieve an average score of 0.75 in the *MEN relatedness test*, which is comparable to results obtained by state-of-the-art algorithms.

1 Introduction

The development of data-driven methods of natural language processing starts with an educated guess, a distributional hypothesis: We assume that some properties of linguistic entities can be modelled by 'some statistical' observations in language data. In the second step, this statistical information (which is determined by the hypothesis) is collected and represented in a mathematical framework. In the third step, tools provided by the chosen mathematical framework are used to implement a similarity-based logic to identify linguistic structures, and/or to verify the proposed hypothesis. Harris's distributional hypothesis (Harris, 1954) is a well-known example of step one that states that meanings of words correlate with the environment in which the words appear. Vector space models and η-normed-based similarity measures are notable examples of steps two and three, respectively (i.e., word space models or word embeddings).

However, as pointed out for instance by Baroni et al. (2014), the *count-based models* resulting from the steps two and three are not discriminative enough to achieve satisfactory results; instead, *predictive models* are required. To this end, an additional transformation step is often added. Turney and Pantel (2010) describe this extra step as a combination of *weighting* and *dimensionality reduction*.[1] This transformation from count-based to predictive models can be implemented simply via a collection of rules of thumb (such as frequency threshold to filter out highly frequent and/or rare context elements), and/or it can involve more sophisticated mathematical transformations, such as converting raw counts to probabilities and using matrix factorization techniques. Likewise, by exploiting the large amounts of computational power available nowadays, this transformation can be achieved via neural word embedding tech-

---

[1] Similar to topics of feature weighting, selection, and engineering in statistical machine learning.

niques (Mikolov et al., 2013; Levy and Goldberg, 2014).

To a large extent, the need for such transformations arises from the *heavy-tailed* distributions that we often find in statistical natural language models (such as the Zipfian distribution of words in contexts when building word spaces). Consequently, count-based models are sparse and high-dimensional and therefore both computationally expensive to manipulate (due of the high dimensionality of models) and nondiscriminatory (due to the combination of the high-dimensionality of the models and the sparseness of observations—see Minsky and Papert (1969, chap. 12)).[2]

On the one hand, although neural networks are often the top performers for addressing this problem, their usage is costly: they need to be trained, which is often very time-consuming,[3] and their performance can vary from one task to another depending on their *objective function*.[4] On the other hand, although methods based on random projections efficiently address the problem of reducing the dimensionality of vectors—such as random indexing (RI) (Kanerva et al., 2000), reflective random indexing (RRI), (Cohen et al., 2010), ISA (Baroni et al., 2007) and random Manhattan indexing (RMI) (Zadeh and Handschuh, 2014)—in effect they retain distances between entities in the original space.[5] Moreover, since these methods use asymptotic Gaussian or Cauchy random projection matrices $\mathbf{R}$ with $\mathrm{E}(\mathbf{R}) = 0$, their resulting vectors cannot be adjusted and transformed using weighting techniques such as PPMI. Consequently, these methods often do not outperform neural embeddings and combinations of PPMI weighting of count-based models followed by matrix factorization—such as the truncation of weighted vectors using singular value decomposition (SVD).

To overcome these problems, we propose a new method called *positive-only projection (PoP)*. PoP is an incremental semantic space construction method which employs random projections. Hence, building models using PoP does not require training but simply generating random vectors. However, in contrast to RI (and previous methods), the PoP-constructed spaces can undergo weighting transformations such as PPMI, after their construction and at a reduced dimensionality. This is due to the fact that PoP uses random vectors that contain only positive integer values. Because the PoP method employs random projections, models can be built incrementally and efficiently. Since the vectors in PoP-constructed models are small (i.e., with a dimensionality of a few hundred), applying weighting methods such as PPMI to these models is incredibly faster than applying them to classical count-based models. Combined with a suitable weighting method such as PPMI, the PoP algorithm yields competitive results concerning accuracy in semantic similarity assessments, compared for instance to neural net-based approaches and combinations of count-based models with weighting and matrix factorization. These results, however, are achieved without the need for heavy computations. Thus, instead of hours, models can be built in a matter of a few seconds or minutes. Note that even without weighting transformation, PoP-constructed models display a better performance than RI on tasks of semantic similarity assessments.

We describe the PoP method in § 2. In order to evaluate our models, in § 3, we report the performance of PoP in the MEN relatedness test. Finally, § 4 concludes with a discussion.

2 Method

2.1 Construction of PoP Models

A transformation of a count-based model to a predictive one can be expressed using a matrix notation such as:

$$\mathbf{C}_{p \times n} \times \mathbf{T}_{n \times m} = \mathbf{P}_{p \times m}. \qquad (1)$$

---

[2]That is, the well known *curse of dimensionality* problem.

[3]Baroni et al. (2014) state that it took *Ronan Collobert* two months to train a set of embeddings from a Wikipedia dump. Even using GPU-accelerated computing, the required computation and training time for inducing neural word embeddings is high.

[4]Ibid, see results reported in supplemental materials.

[5]For η-normed space that they are designed for, i.e., $\eta = 2$ for RI, RRI, and ISA and $\eta = 1$ for RMI.

In Equation 1, $\mathbf{C}$ denotes the count-based model consisting of p vectors and n context elements (i.e., n dimensions). $\mathbf{T}$ is the transformation matrix that maps the p n-dimensional vectors in $\mathbf{C}$ to an m-dimensional space (often, but not necessarily, $m \neq n$ and $m \ll n$). Finally, $\mathbf{P}$ is the resulting m-dimensional predictive model. Note that $\mathbf{T}$ can be a composition of several transformations, e.g., a weighting transformation $\mathbf{W}$ followed by a projection onto a space of lower dimensionality $\mathbf{R}$, i.e., $\mathbf{T}_{n \times m} = \mathbf{W}_{n \times n} \times \mathbf{R}_{n \times m}$.

In the proposed PoP technique, the transformation $\mathbf{T}_{n \times m}$ (for $m \ll n$, e.g., $100 \leq m \leq 7000$) is simply a randomly generated matrix. The elements t_{ij} of $\mathbf{T}_{n \times m}$ have the following distribution:

$$t_{ij} = \begin{cases} 0 & \text{with probability } 1 - s \\ \lfloor \frac{1}{U^\alpha} \rfloor & \text{with probability } s \end{cases}, \quad (2)$$

in which U is an independent uniform random variable in $(0, 1]$, and s is an extremely small number (e.g., $s = 0.01$) such that each row vector of $\mathbf{T}$ has at least one element that is not 0 (i.e., $\sum_{i=1}^{m} t_{ji} \neq 0$ for each row vector $t_j \in \mathbf{T}$). For α, we choose $\alpha = 0.5$. Given Equations 1 and 2 and using the distributive property of multiplication over addition in matrices,[6] the desired semantic space (i.e., $\mathbf{P}$ in Equation 1) can be constructed using the two-step procedure of incremental word space construction (such as used in RI, RRI, and RMI):

Step 1. Each context element is mapped to one m-dimensional *index vector* $\vec{r}$. $\vec{r}$ is randomly generated such that most elements in $\vec{r}$ are 0 and only a few are positive integers (i.e., the elements of $\vec{r}$ have the distribution given in Equation 2).

Step 2. Each target entity that is being analysed in the model is represented by a *context vector* $\vec{v}$ in which all the elements are initially set to 0. For each encountered occurrence of this target entity together with a context element (e.g., through a sequential scan of a corpus), we update $\vec{v}$ by adding the index vector $\vec{r}$ of the context element to it.

This process results in a model built directly at the reduced dimensionality m (i.e., $\mathbf{P}$ in Equation 1). The first step corresponds to the construction of the randomly generated transformation matrix $\mathbf{T}$: Each index vector is a row of the transformation matrix $\mathbf{T}$. The second step is an implementation of the matrix multiplication in Equation 1 which is distributed over addition: Each context vector is a row of $\mathbf{P}$, which is computed in an iterative process.

2.2 Measuring Similarity

Once $\mathbf{P}$ is constructed, if desirable, similarities between entities can be computed by their *Kendall*'s τ_b ($-1 \leq \tau_b \leq 1$) correlation (Kendall, 1938). To compute τ_b, we adopt an implementation of the algorithm proposed by Knight (1966), which has a computational complexity of $\mathrm{O}(n \log n)$.[7]

In order to compute τ_b, we need to define a number of values. Given vectors $\vec{x}$ and $\vec{y}$ of the same dimension, we call a pair of observations (x_j, y_j) and (x_{j+1}, y_{j+1}) in $\vec{x}$ and $\vec{y}$ *concordant* if $(x_j < x_{j+1} \wedge y_j < y_{j+1}) \vee (x_j > x_{j+1} \wedge y_j > y_{j+1})$. The pair is called *discordant* if $(x_j < x_{j+1} \wedge y_j > y_{j+1}) \vee (x_j > x_{j+1} \wedge y_j < y_{j+1})$. Finally, the pair is called *tied* if $x_j = x_{j+1} \vee y_j = y_{j+1}$. Note that a tied pair is neither concordant nor discordant. We define n_1 and n_2 as the number of pairs with tied values in $\vec{x}$ and $\vec{y}$, respectively. We use n_c and n_d to denote the number of concordant and discordant pairs, respectively. If m is the dimension of the two vectors, then n_0 is defined as the total number of observation pairs: $n_0 = \frac{m(m-1)}{2}$. Given these definitions, Kendall's τ_b is given by

$$\tau_b = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}}.$$

The choice of τ_b can be motivated by generalising the role that cosine plays for comput-

---

[6]That is $(\mathbf{A} + \mathbf{B}) \times \mathbf{C} = \mathbf{A} \times \mathbf{C} + \mathbf{B} \times \mathbf{C}$.

[7]In our evaluation, we use the implementation of Knight's algorithm in the *Apache Commons Mathematics Library*.

ing similarities between vectors that are derived from a standard Gaussian random projection. In random projections with $\mathbf{R}$ of (asymptotic) $\mathcal{N}(0,1)$ distribution, despite the common interpretation of the cosine similarity as the angle between two vectors, cosine can be seen as a measure of the product-moment correlation coefficient between the two vectors. Since $\mathbf{R}$ and thus the obtained projected spaces have zero expectation, Pearson's correlation and the cosine measure have the same definition in these spaces (see also Jones and Furnas (1987) for a similar claim and on the relationships between correlation and the inner product and cosine). Subsequently, one can propose that in Gaussian random projections, Pearson's correlation is used to compute similarities between vectors.

However, the use of projections proposed in this paper (i.e., $\mathbf{T}$ with a distribution set in Equation 2) will result in vectors that have a non-Gaussian distribution. In this case, τ_b becomes a reasonable candidate for measuring similarities (i.e., correlations between vectors) since it is a nonparametric correlation coefficient measure that does not assume a Gaussian distribution (see Chen and Popovich (2002)) of projected spaces. However, we do not exclude the use of other similarity measures and may employ them in future work. In particular, we envisage additional transformations of PoP-constructed spaces to induce vectors with Gaussian distributions (see for instance the log-based PPMI transformation used in the next section). If a transformation to a Gaussian-like distribution is performed, then it is expected that the use of Pearson's correlation, which works under the assumption of Gaussian distribution, yields better results than Kendall's correlation (as confirmed by our experiments).

2.3 Some Delineation of the PoP Method

The PoP method is a *randomized algorithm*. In this class of algorithms, at the expense of a tolerable loss in accuracy of the outcome of the computations (of course, with a certain acceptable amount of probability) and by

the help of *random decisions*, the computational complexity of algorithms for solving a problem is reduced (see, e.g., Karp (1991), for an introduction to randomized algorithms).[8] For instance, using Gaussian-based sparse random projections in RI, the computation of eigenvectors (often of the complexity of $\mathrm{O}(n^2 \log m)$) is replaced by a much simpler process of random matrix construction (of an estimated complexity of $\mathrm{O}(n)$)—see Bingham and Mannila (2001). In return, randomized algorithms such as the PoP and RI methods give different results even for the same input.

Assume the difference between the optimum result and the result from a randomized algorithm is given by δ (i.e., the error caused by replacing deterministic decisions with random ones). Much research in theoretical computer science and applied statistics focuses on specifying bounds for δ, which is often expressed as a function of the probability ϵ of encountered errors. For instance, δ and ϵ in Gaussian random projections are often derived from the lemma proposed by Johnson and Lindenstrauss (1984) and its variations. Similar studies for random projections in ℓ_1-normed spaces and deep neural networks are Indyk (2000) and Arora et al. (2014), respectively.

At this moment, unfortunately, we are not able to provide a detailed mathematical account for specifying δ and ϵ for the results obtained by the PoP method (nor are we able to pinpoint a theoretical discussion about PoP's underlying random projection). Instead, we rely on the outcome of our simulations and the performance of the method in an NLP task. Note that this is not an unusual situation. For instance, Kanerva et al. (2000) proposed RI with no mathematical justification. In fact, it was only a few years later that Li et al. (2006) proposed mathematical lemmas for justifying very sparse Gaussian random projections such as RI (QasemiZadeh, 2015). At any rate, projections onto manifolds is a vibrant research both in theoretical computer

---

[8] Such as many classic search algorithms that are proposed for solving NP-complete problems in artificial intelligence.

science and in mathematical statistics. Our research will benefit from this in the near future. If δ refers to the amount of distortion in pairwise ℓ_2 norm correlation measures in a PoP space,[9] it can be shown that δ and its variance σ_δ^2 are functions of the dimension m of the projected space, that is: $\sigma_\delta^2 \approx \frac{1}{m}$, based on similar mathematical principles proposed by Kaski (1998) (and of Hecht-Nielsen (1994)) for the *random mapping*.

Our empirical research and observations on language data show that projections using the PoP method exhibit similar behavioural patterns as other sparse random projections in α-normed spaces. The dimension m of random index vectors can be seen as the capacity of the method to memorize and distinguish entities. For m up to a certain number ($100 \leq m \leq 6000$) in our experiments, as was expected, a PoP-constructed model for a large m shows a better performance and smaller δ than a model for a small m. Since observations in semantic spaces have a *very*-long-tailed distribution, choosing different values of non-zero elements for index vectors does not effect the performance (as mentioned, in most cases 1, 2 or 3 non-zero elements are sufficient). Furthermore, changes in the adopted distribution of t_{ij} only slightly affect the performance of the method.

In the next section, using empirical investigations we show the advantages of the PoP model and support the claims from this section.

3 Evaluation & Empirical Investigations

3.1 Comparing PoP and RI

For evaluation purposes, we use the MEN relatedness test set (Bruni et al., 2014) and the UKWaC corpus (Baroni et al., 2009). The dataset consists of 3000 pairs of words (from 751 distinct tagged lemmas). Similar to other 'relatedness tests', Spearman's rank correlation ρ score from the comparison of human-based ranking and system-induced rankings is the figure of merit. We use these resources

for evaluation since they are in public domain, both the dataset and corpus are large, and they have been used for evaluating several word space models—for example, see Levy et al. (2015), Tsvetkov et al. (2015), Baroni et al. (2014), Kiela and Clark (2014). In this section, unless otherwise stated, we use cosine for similarity measurements.

Figure 1 shows the performance of the simple count-based word space model for lemmatized-context-windows that extend symmetrically around lemmas from MEN.[10] As expected, up to a certain context-window size, the performance using count-based methods increases with an extension of the window.[11] For context-windows larger than 25+25 the performance gradually declines. More importantly, in all cases, we have $\rho < 0.50$.

We performed the same experiments using the RI technique. For each context window size, we performed 10 runs of the RI model construction. Figure 1 reports for each context-window size the average of the observed performances for the 10 RI models. In this experiment, we used index vectors of dimensionality 1000 containing 4 non-zero elements. As shown in Figure 1, the average performance of the RI is almost identical to the performance of the count-based model. This is an expected result since RI's objective is to retain Euclidean distances between vectors (thus cosine) but in spaces of lowered dimensionality. In this sense, RI is successful and achieves its goal of lowering the dimensionality while keeping Euclidean distances between vectors. However, using RI+cosine does not yield any improvements in the similarity assessment task.

We then performed similar experiments using PoP-constructed models, with the same context window sizes and the same dimen-

---

[9] As opposed to pairwise correlations in the original high-dimensional space.

[10] We use the tokenized preprocessed UKWaC. However, except for using part-of-speech tags for locating lemmas listed in MEN, we do not use any additional information or processes (i.e., no frequency cut-off for context selection, no syntactic information, etc.).

[11] After all, in models for relatedness tests, relationships of topical nature play a more important role than other relationships such as synonymy.
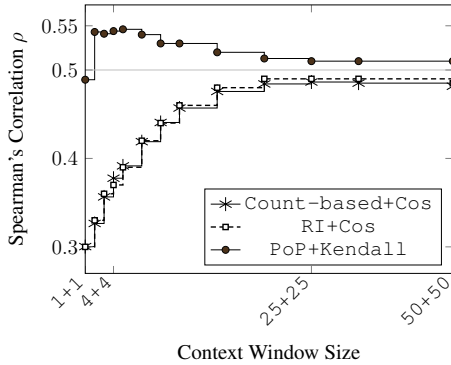
Figure 1: Performance of the classic count-based a-word-per-dimension model vs. RI vs. Pop in the MEN relatedness test. Note that count-based and RI models show almost an identical performance in this task.
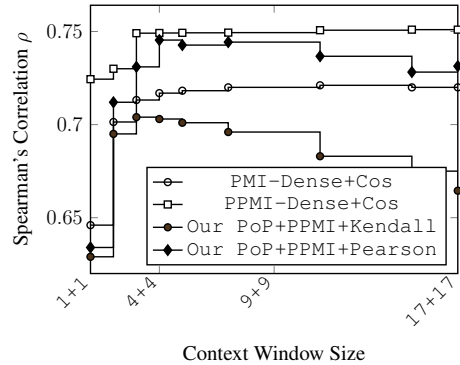

Figure 2: Performances of (P)PMI-transformed models for various sizes of context-windows. From context size 4+4, the performance remains almost intact (0.72 for PMI and 0.75 for PPMI). We also report the average performance for PoP-constructed models constructed at the dimensionality $m = 1000$ and $s = 0.002$. PoP+PPMI+Pearson exhibits a performance similar as dense PPMI-weighted models, however, much faster and using far less amount of computational resources. Note that reported PoP+PMI performances can be enhanced by using $m > 1000$.

sions as in the RI experiments, averaging again over 10 runs for each context window size. The performance is also reported in Figure 1. For the PoP method, however, instead of using the cosine measure we use Kendall's τ_b for measuring similarities. The PoP-constructed models converge faster than RI and count-based methods and for smaller context-windows they outperform the count-based and RI methods with a large margin. However, as the sizes of the windows grow, performances of these methods become more similar (but PoP still outperforms the others). In any case, the performance of PoP remains above 0.50 (i.e., $\rho > 0.50$). Note that in RI-constructed models, using Kendall's τ_b also yield better performance than using cosine.

3.2 PPMI Transformation of PoP Vectors

Although PoP outperforms RI and count-based models, compared to the state-of-the-art methods, its performance is still not satisfying. Transformations based on association measures such as PPMI have been proposed to improve the discriminatory power of context vectors and thus the performance of models in semantic similarity assessment tasks (see Church and Hanks (1990), Turney (2001), Turney (2008), and Levy et al. (2015)). For a given set of vectors, pointwise mutual information (PMI) is interpreted as a measure of information overlap between vectors. As

put by Bouma (2009), PMI is a mathematical tool for measuring how much the actual probability of a particular co-occurrence (e.g., two words in a word space) deviate from the expected probability of their individual occurrences (e.g., the probability of occurrences of each word in a words space) under the assumption of independence (i.e., the occurrence of one word does not affect the occurrences of other words).

In Figure 2, we show the performance of PMI-transformed spaces. Count-based PMI+Cosine models outperform other techniques including the introduced PoP method. The performance of PMI models can be further enhanced by their normalization, often discarding negative values[12] and using PPMI. Also, SVD truncation of PPMI-weighted spaces can improve the performance slightly (see the above mentioned references) requiring, however, expensive computations of

---

[12]See Bouma (2009) for a mathematical delineation. Jurafsky and Martin (2015) also provide an intuitive description.

eigenvectors.[13] For a $p \times n$ matrix with elements v_{xy}, $1 \le x \le p$ and $1 \le y \le n$, we compute the PPMI weight for a component v_{xy} as follows:

$$ppmi(v_{xy}) = \max(0, \log \frac{v_{xy} \times \sum_{i=1}^{p} \sum_{j=1}^{n} v_{ij}}{\sum_{i=1}^{p} v_{iy} \times \sum_{j=1}^{n} v_{xj}}). \quad (3)$$

The most important benefit of the PoP method is that PoP-constructed models, in contrast to previously suggested random projection-based models, can be still weighted using PPMI (or any other weighting techniques applicable to the original count-based models). In an RI-constructed model, the sum of values of row and column vectors of the model are always 0 (i.e., $\sum_{i=1}^{p} v_{iy}$ and $\sum_{j=1}^{n} v_{xj}$ in Equation 3 are always 0). As mentioned earlier, this is due to the fact that a random projection matrix in RI has an asymptotic standard Gaussian distribution (i.e., transformation matrix $\mathbf{R}$ has $E(\mathbf{R}) = 0$). As a result, PPMI weights for the RI-induced vector elements are undefined. In contrast to RI, the sum of values of vector elements in the PoP-constructed models is always greater than 0 (because the transformation is carried out by a projection matrix $\mathbf{R}$ of $E(\mathbf{R}) > 0$). Also, depending on the structure of data in the underlying count-based model, by choosing a suitably large value of s, it can be guaranteed that the sum of column vectors is always a non-zero value. Hence, vectors in PoP models can undergo the PPMI transformation defined in Equation 3. Moreover, the PPMI transformation in PoP models is much faster, compared to the one performed on count-based models, due to the low dimensionality of vectors in the PoP-constructed model. Therefore, the PoP method makes it possible to benefit both from the high efficiency of randomized techniques as well as from the high accuracy of PPMI transformation in semantic similarity tasks.

If we put aside the information-theoretic interpretation of PPMI weighting (i.e., distilling statistical information that matters),

the logarithmic transformation of probabilities in the PPMI definition plays the role of a *power transformation* process for converting long-tailed distributions in the original high-dimensional count-based models to Gaussian-like distributions in the transformed models. From a statistical perspective, any variation of PMI transformation can be seen as an attempt to stabilize the variance of vector coordinates and therefore to make the observations more similar/fit to Gaussian distribution (a practice with a long history in research, particularly in the biological and psychological sciences).

To exemplify this phenomenon, in Figure 3, we show histograms of the distributions of the assigned weights to the vector that represents the lemmatized form of the verb 'abandon' in various models. As shown, the raw collected frequencies in the original high-dimensional count-based model have a long tail distribution (see Figure 3a). Applying the log transformation to this vector yields a vector of weights with a Gaussian distribution (Figure 3b). Weights in the RI-constructed vector (Figure 3c) have a perfect Gaussian distribution but with an expected value of 0 (i.e., $\mathcal{N}(0, 1)$). The PoP method, however, largely preserves the long tail distribution of coordinates from the original space (Figure 3d), which in turn can be weighted using PPMI and thereby transformed into a Gaussian-like distribution.

Given that models after the PPMI transformation have bell-shaped Gaussian distributions, we expect that a correlation measure such as Pearson's r, which takes advantage of the prior knowledge about the distribution of data, outperforms the non-parametric Kendall's τ_b for computing similarities in PPMI-transformed spaces.[14] This is indeed the case (see Figure 2).

---

[13]In our experiments, applying SVD truncation to models results in negligible improvements between 0.01 and 0.001.

[14]Note that using correlation measures such as Pearson's r and Kendall's τ_b in count-based model may excel measures such as cosine. However, their application is limited due to the high-dimensionality of count-based methods.
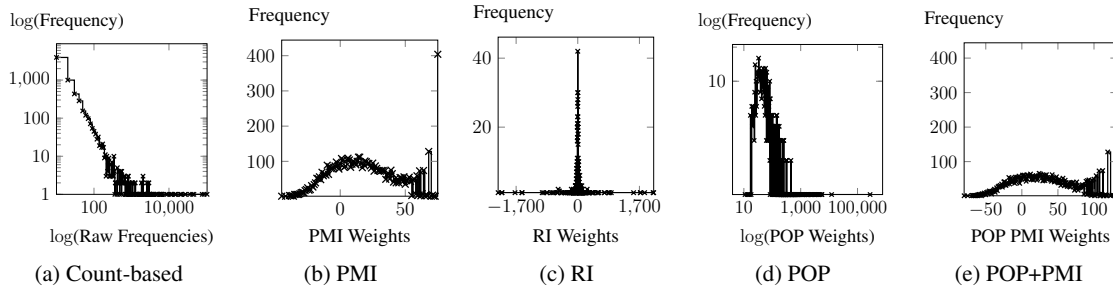
Figure 3: A histogram of the distribution of frequencies of weights (i.e., the value of the coordinates) in various models built from 1+1 context-windows for the lemmatized form of the verb 'abandon' in the UKWaC corpus.

3.3 PoP's Parameters, its Random Behavior and Performance

As discussed in § 2.3, PoP is a randomized algorithm and its performance is influenced by a number of parameters. In this section, we study the PoP method's behavior by reporting its performance in the MEN relatedness test under different parameter settings. To keep evaluations and reports to a manageable size, we focus on models built using context-windows of size 4+4.

Figure 4 shows the method's performance when the dimension m of the projected index vectors increases. In these experiments, index vectors are built using 4 non-zero elements; thus, as m increases, s in Equation 2 decreases. For each m, $100 \leq m \leq 5000$, the models are built 10 times and the average as well as the maximum and the minimum observed performances in these experiments are reported. For PPMI transformed PoP spaces, with increasing dimensions, the performance boosts and, furthermore, the variance in performance (i.e., the shaded areas)[15] gets smaller.

However, for the count-based PoP method without PPMI transformation (shown by the dash-dotted lines) and with the number of non-zero elements fixed to 4, increasing m over 2000 decreases the performance. This is unexpected since an increase in dimensionality is usually assumed to entail an increase in performance. This behavior, however, can
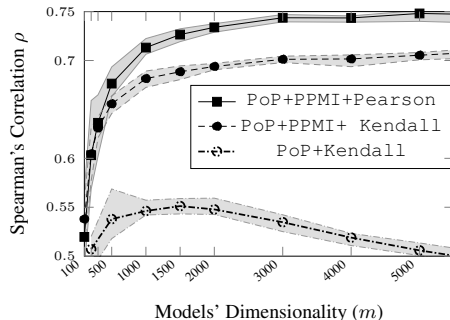

Figure 4: Changes in PoP's performance when the dimensionality of models increases. The average performance in each set-up is shown by the marked lines. The margins around these lines show the minimum and maximum performance observed in 10 independent executions.

be the result of using a very small s; simply put, the number of non-zero elements are not sufficient to build projected spaces with adequate distribution. To investigate this matter, we study the performance of the method with the dimension m fixed to 3000 but with index vectors built using different numbers of non-zero elements, i.e., different values of s.

Figure 5 shows the observed performances. For PPMI-weighted spaces, increasing the number of non-zero elements clearly deteriorates the performance. For unweighted PoP models, an increase in s up to the limit that does not result in non-orthogonal index vectors enhances performances. As shown in Figure 6, when the dimensionality of the index vectors is fixed and s increases, the chances of
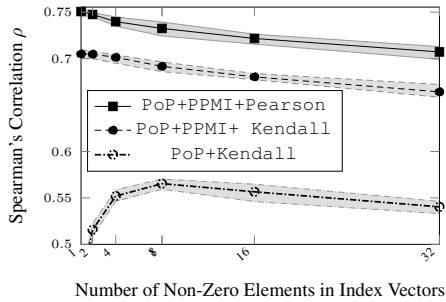
---

[15]Evidently, the probability of worst and best performances can be inferred from the reported average results.

Figure 5: Changes in PoP's performances when the dimensionality of models are fixed to $m = 3000$ and the number of non-zero elements in index vectors (i.e., s) increases. The average performances in each set-up are shown by marked lines. The margins around these lines show the minimum and maximum performance observed in 10 independent executions.

having non-orthogonal vectors in index vectors are boosted. Hence, the chance of distortions in similarities increases. These distortions can enhance the result if they are controlled (e.g., using a training procedure such as the one used in neural net embedding). However, when left to chance, they can often lower the performance. Evidently, this is an oversimplified justification: in fact, s plays the role of a switch that controls the resemblance between the distribution of data in the original space and the projected/transformed spaces. It seems that the sparsity of vectors in the original matrix plays a role in finding the optimal value for s. If PoP-constructed models are used directly (together with τ_b) for computing similarities, then we propose $0.002 < s$. If PoP-constructed models are subject to an additional weighting process for stabilizing vector distributions into Gaussian-like distributions such as PPMI, we propose using only 1 or 2 non-zero elements.

Last but not least, we confirm that by carefully selecting context elements (i.e., removing stop words and using lower and upper bound frequency cut-offs for context selection) and fine tuning PoP+PPMI+Pearson (i.e., increasing the dimension of models and scaling PMI weights as in Levy et al. (2015))
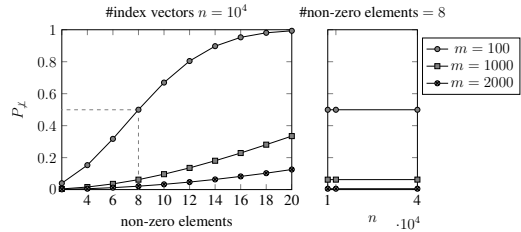

Figure 6: The proportion of non-orthogonal pairs of index vectors ($P_{\not\perp}$) obtained in a simulation for various dimensionality and number of non-zero elements. The left figure shows the changes of $P_{\not\perp}$ for a fixed number of index vectors $n = 10^4$ when the number of non-zero elements increases. The right figure shows $P_{\not\perp}$ when the number of non-zero elements is fixed to 8 but the number of index vectors n increases. As shown, $P_{\not\perp}$ is determined by the number of non-zero elements and the dimensionality of index vectors and independently of n.

we achieve an even higher score in the MEN test (i.e., an average of 0.78 with the max of 0.787). Moreover, although improvements from applying SVD truncation are negligible, we can employ it for reducing the dimensionality of PoP vectors (e.g., from 6000 to 200).

4 Conclusion

We introduced a new technique called PoP for the incremental construction of semantic spaces. PoP can be seen as a dimensionality reduction method, which is based on a newly devised random projection matrix that contains only positive integer values. The major benefit of PoP is that it transfers vectors onto spaces of lower dimensionality without changing their distribution to a Gaussian shape with zero expectation. The obtained transformed spaces using PoP can, therefore, be manipulated similarly to the original high-dimensional spaces, only much faster and consequently requiring a considerably lower amount of computational resources.

PPMI weighting can be easily applied to PoP-constructed models. In our experiments, we observe that PoP+PPMI+Pearson can be used to build models that achieve a high per-

formance in semantic relatedness tests. More concretely, for index vector dimensions $m \geq 3000$, PoP+PPMI+Pearson achieves an average score of 0.75 in the MEN relatedness test, which is comparable to many neural embedding techniques (e.g., see scores reported in Chen and de Melo (2015) and Tsvetkov et al. (2015)). However, in contrast to these approaches, PoP+PPMI+Pearson achieves this competitive performance without the need for time-consuming training of neural nets. Moreover, the processes involved are all done on vectors of low dimensionality. Hence, the PoP method can dramatically enhance the performance in tasks involving distributional analysis of natural language.

Acknowledgments

References

Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. 2014. Provable bounds for learning some deep representations. In *Proceedings of the 31st International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 584–592.

Marco Baroni, Alessandro Lenci, and Luca Onnis. 2007. ISA meets Lara: An incremental word space model for cognitively plausible simulations of semantic learning. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 49–56, Prague, Czech Republic, June. Association for Computational Linguistics.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland, June. Association for Computational Linguistics.

Ella Bingham and Heikki Mannila. 2001. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge discovery and data mining*, KDD '01, pages 245–250, New York, NY, USA. ACM.

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the Biennial GSCL Conference*.

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

Jiaqiang Chen and Gerard de Melo. 2015. Semantic information extraction for improved word embeddings. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 168–175, Denver, Colorado, June. Association for Computational Linguistics.

Peter Y. Chen and Paula M. Popovich. 2002. *Correlation: Parametric and Nonparametric Measures*. Quantitative Applications in the Social Sciences. Sage Publications.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16(1):22–29, March.

Trevor Cohen, Roger Schvaneveldt, and Dominic Widdows. 2010. Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. *Journal of Biomedical Informatics*, 43(2):240 – 256.

Zellig Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Robert Hecht-Nielsen. 1994. Context vectors: general purpose approximate meaning representations self-organized from raw data. *Computational Intelligence: Imitating Life*, pages 43–56.

Piotr Indyk. 2000. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *41st Annual Symposium on Foundations of Computer Science*, pages 189–197.

William Johnson and Joram Lindenstrauss. 1984. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, volume 26 of *Contemporary Mathematics*, pages 189–206. American Mathematical Society.

William P. Jones and George W. Furnas. 1987. Pictures of relevance: A geometric analysis of similarity measures. *Journal of the American Society for Information Science*, 38(6):420–442.

Daniel Jurafsky and James H. Martin, 2015. *Speech and Language Processing*, chapter Chapter 19: Vector Semantics. Prentice Hall, 3rd edition. Draft of August 24, 2015.

Pentti Kanerva, Jan Kristoferson, and Anders Holst. 2000. Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, pages 103–6. Erlbaum.

Richard M. Karp. 1991. An introduction to randomized algorithms. *Discrete Applied Mathematics*, 34(13):165 – 201.

Samuel Kaski. 1998. Dimensionality reduction by random mapping: fast similarity computation for clustering. In *The 1998 IEEE International Joint Conference on Neural Networks*, volume 1, pages 413–418.

Maurice G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93.

Douwe Kiela and Stephen Clark. 2014. A systematic study of semantic vector space model parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 21–30, Gothenburg, Sweden, April. Association for Computational Linguistics.

William R. Knight. 1966. A computer method for calculating kendall's tau with ungrouped data. *Journal of the American Statistical Association*, 61(314):436–439.

Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2177–2185. Curran Associates, Inc.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

Ping Li, Trevor J. Hastie, and Kenneth W. Church. 2006. Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '06, pages 287–296, New York, NY, USA. ACM.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Marvin Lee Minsky and Seymour Papert. 1969. *Perceptrons: An Introduction to Computational Geometry*. MIT Press.

Behrang QasemiZadeh. 2015. Random indexing revisited. In *20th International Conference on Applications of Natural Language to Information Systems, NLDB*, pages 437–442. Springer.

The Apache Commons Mathematics Library [Computer Software]. 2016. KendallsCorrelation Class. Retrieved from https://commons.apache.org/proper/commons-math/javadocs/api-3.6.1/index.html.

Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. 2015. Evaluation of word vector representations by subspace alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2049–2054, Lisbon, Portugal, September. Association for Computational Linguistics.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, January.

Peter D. Turney. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of the 12th European Conference on Machine Learning*, EMCL '01, pages 491–502, London, UK, UK. Springer-Verlag.

Peter D. Turney. 2008. The latent relation mapping engine: Algorithm and experiments. *J. Artif. Int. Res.*, 33(1):615–655, December.

Behrang Q. Zadeh and Siegfried Handschuh. 2014. Random Manhattan integer indexing: Incremental L1 normed vector space construction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1713–1723.

A   Supplemental Material

Codes and resulting embeddings from experiments are available from `https://user.phil-fak.uni-duesseldorf.de/~zadeh/material/pop-vectors`.