Postprint: Word Embedding Method Based on Centered Similarity Matrix

Authors: Xu Fan, Wang Peiyan, CAI Dongfeng

Date: 2018-05-20T00:00:00+00:00

Abstract

Word embeddings utilize low-dimensional dense vectors to represent words, enabling vector operations to reflect inter-word relationships, and have been widely applied in natural language processing tasks. This work investigates matrix factorization-based word embedding methods, discovers a linear correlation between the quality of the similarity matrix prior to dimensionality reduction and the quality of the resulting word embeddings, and proposes a method based on centered similarity matrices. This method relatively enhances (weakens) the degree of similarity between similar (dissimilar or weakly similar) words. The effectiveness of the proposed method is verified through word similarity experiments on the WS-353 and RW datasets, achieving maximum improvements in word embedding quality of 0.2896 and 0.1801, respectively. Centering can improve the quality of the similarity matrix before dimensionality reduction, thereby enhancing word embedding quality.

Full Text

Preamble

Title: Method of Word Vector Based on Centring Similarity Matrix

Authors: Xu Fan, Wang Peiyan, Cai Dongfeng
Affiliation: Human-Computer Intelligence Research Center, Shenyang Aerospace University, Shenyang 110136, China

Abstract: Word vectors represent words as low-dimensional dense vectors, capturing lexical relationships through vector operations, and have become widely applied in natural language processing tasks. This paper investigates matrix factorization-based word vector methods and discovers a linear correlation between the quality of the pre-dimensionality-reduction similarity matrix and the quality of the resulting word vectors. Building on this finding, we propose a method based on centring the similarity matrix, which relatively enhances the

similarity between similar words while diminishing similarity between dissimilar or weakly similar words. Experiments on word similarity tasks using the WS-353 and RW datasets validate the effectiveness of our approach, achieving maximum improvements in word vector quality of 0.2896 and 0.1801 on the two datasets respectively. Centring improves the quality of the similarity matrix before dimensionality reduction, thereby enhancing word vector quality.

Keywords: word vector; centralization; similarity matrix

0 Introduction

Word vectors extract semantic and syntactic information from large unlabeled corpora, representing each word as a low-dimensional real-valued vector that maps discrete words to features in a real-valued space. The closer the vector representations, the greater the semantic similarity between words. Word vectors can be used to compute inter-word similarity and serve as features directly applied to natural language processing tasks such as word sense disambiguation, text classification, part-of-speech tagging, and sentiment analysis.

Word vector methods fall into two categories: matrix factorization-based approaches and prediction-based approaches. Matrix factorization methods originate from the distributional hypothesis—that words with similar contexts share similar meanings—tracing back to Latent Semantic Analysis (LSA), which obtains word vectors by factorizing word-document matrices. Current practice predominantly employs word-context co-occurrence matrices. Prediction-based methods stem from neural network models, with Mikolov et al. proposing CBOW and Skip-gram, which have attracted widespread attention due to their semantic properties. Levy et al. conducted a detailed analysis comparing matrix factorization-based methods with Skip-gram on word similarity tasks, experimenting with various parameters and finding that both approaches achieve comparable results under most configurations.

Following Skip-gram' s widespread adoption, researchers began investigating its relationship with matrix factorization-based methods, focusing on theoretical interpretability. Levy et al. demonstrated that the Skip-gram with Negative Sampling (SGNS) training method can be viewed as weighted matrix factorization, equivalent to implicit factorization of the Shifted Positive PMI (SPPMI) matrix. Li et al. similarly found SGNS equivalent to word-context co-occurrence matrix factorization, leveraging this equivalence to incorporate supervision and achieve 9% performance improvement on word analogy tasks with only 10% training data. Pennington et al. proposed the GloVe model by exploiting Skip-gram' s ability to capture linear relationships and its equivalence to matrix factorization, showing strong performance on word analogy tasks.

Since both Skip-gram and matrix factorization-based methods can be regarded as investigations of the word-context co-occurrence matrix, and the resulting

word vector qualities are comparable, research interest in matrix factorization-based methods has been renewed. A representative approach is Hellinger PCA (HPCA), which obtains word vectors by first constructing a word-context co-occurrence matrix using conditional probabilities, then computing pairwise similarities between matrix rows using Hellinger distance to obtain a similarity matrix, and finally reducing its dimensionality. This method demonstrates strong performance on both word similarity and word analogy tasks.

Building upon the algorithmic process described in prior work, this paper investigates matrix factorization-based word vector methods and finds that the pre-dimensionality-reduction similarity matrix directly influences word vector quality. Using Pearson correlation coefficient, we verify a strong linear relationship between them. Consequently, we propose a word vector method based on centring the similarity matrix. By centring the similarity matrix, similarity between similar words is relatively enhanced while similarity between dissimilar or weakly similar words is diminished. We validate the effectiveness of this approach on word similarity tasks, showing that word vectors obtained from centred similarity matrices significantly outperform those from non-centred matrices.

---

1 Matrix Factorization-Based Word Vector Methods

1.1 Construction of Word-Context Co-occurrence Matrix

Each element in the word-context co-occurrence matrix C represents the co-occurrence weight $C(t_i, c_j)$ between target word t_i and context word c_j, where V is the vocabulary size of target words and D is the vocabulary size of context words. Thus, C is a $V \times D$ matrix where each row $C_{i,\cdot}$ represents word t_i based on context words c_j.

Common weight calculation methods include Term Frequency (TF), Pointwise Mutual Information (PMI), and Conditional Probability (CP). Levy et al. proposed using Positive PMI (PPMI) and SPPMI to achieve equivalence with the Skip-gram model. Lebret and Collobert, as well as the GloVe model, employ conditional probability. This paper adopts the specific calculation method proposed by Caron et al., as shown in , where $\#(t_i, c_j)$ denotes co-occurrence frequency, $\#(t_i)$ and $\#(c_j)$ represent individual occurrence frequencies in the corpus, and N is the total number of words. When t_i and c_j never co-occur, $C(t_i, c_j) = 0$, and we define $PMI(t_i, c_j) = 0$ in such cases.

1.2 Construction of Similarity Matrix

The similarity matrix A is constructed by computing similarity between each pair of row vectors $C_{i,\cdot}$ and $C_{j,\cdot}$ from the co-occurrence matrix, yielding a symmetric matrix where A_{ij} represents the similarity between words w_i and w_j.

Similarity measures include cosine similarity, Euclidean distance, and Hellinger distance. This paper employs both cosine similarity and Hellinger distance.

Cosine similarity is a common similarity measure calculated as:

$$\text{sim}_{\cos}(w_i, w_j) = \frac{C_{i,\cdot} \cdot C_{j,\cdot}}{\|C_{i,\cdot}\| \times \|C_{j,\cdot}\|}$$

Hellinger distance measures similarity between two discrete probability distributions, requiring normalization of word vectors. The normalized vector is:

$$P_i = \left[\frac{C(t_i, c_1)}{\sum_{k=1}^{D} C(t_i, c_k)}, \frac{C(t_i, c_2)}{\sum_{k=1}^{D} C(t_i, c_k)}, \ldots, \frac{C(t_i, c_D)}{\sum_{k=1}^{D} C(t_i, c_k)} \right]$$

The Hellinger distance is then calculated as:

$$\text{sim}_{\text{DH}}(w_i, w_j) = \frac{1}{\sqrt{2}} \sqrt{\sum_{k=1}^{D} (\sqrt{p_{i,k}} - \sqrt{p_{j,k}})^2}$$

1.3 Matrix Factorization

Since the similarity matrix A is symmetric, we apply eigenvalue decomposition for dimensionality reduction to obtain the word vector matrix E. Eigenvalue decomposition retains only the top d eigenvectors corresponding to the largest eigenvalues, as larger eigenvalues capture more information. This process can be viewed as mapping A to a low-dimensional space to obtain its projection matrix $\hat{A}$ such that the difference between A and $\hat{A}$ is minimized.

The decomposition is expressed as:

$$A = Q \Sigma Q^T$$

where Q is the eigenvector matrix and Σ is the diagonal eigenvalue matrix. By sorting eigenvalues λ_i and selecting the top d eigenvectors, we can approximate the matrix using its principal directions, achieving dimensionality reduction.

Levy et al. propose that the optimal value for d should be $p \to 0.5$. Following prior work, we sort eigenvalues λ_i from the decomposition and use the product of the top d eigenvectors and their eigenvalues raised to the power of $p = 0.5$:

$$E = Q_d \Sigma_d^{0.5}$$

1.4 Construction of Matrix Factorization-Based Word Vector Methods

By combining the three weight calculation methods and two similarity measures, we construct five matrix factorization-based word vector methods, as shown in . All use the eigenvalue decomposition described above for dimensionality reduction. Since PMI can produce negative values, it is not paired with Hellinger distance.

shows the method names, weight calculation methods, and similarity calculation methods.

To illustrate the effect of centring, we examine five word pairs with "lobster" as the target word: "seafood," "eye," "glass," "boy," and "shore." compares similarity values before and after centring. Before centring, similarities cluster between 0.27 and 0.35, showing little differentiation. After centring, distinct differences emerge. For instance, "lobster-eye" and "lobster-seafood" have pre-centring similarities of 0.3238 and 0.3272 respectively—nearly identical. Post-centring, these become -0.0017 and 0.2868, transforming "lobster-eye" from similar to dissimilar while clearly distinguishing the two relationships. This demonstrates that centring relatively enhances similarity between similar words while diminishing it between dissimilar or weakly similar words, yielding a more discriminative similarity matrix.

[Figure 1: see original paper] and [Figure 2: see original paper] visualize the 2D vector distributions of these words before and after centring. Before centring, all words cluster in the left region of the origin, with most at similar distances from "lobster," making discrimination difficult. After centring, words distribute around the origin, clearly showing "lobster" is much closer to "seafood" than to "eye." Thus, centring enables similar words to cluster more closely while separating dissimilar words, producing more reasonable vector representations.

---

2 Centring-Based Similarity Matrix Optimization

This section introduces our centring-based similarity matrix optimization method. Both Hellinger distance and cosine similarity used in this paper can be viewed as inner product operations and, according to kernel function definitions, can be considered kernel functions. A kernel function represents the inner product of two sample points in feature space, determining data distribution in that space, as described in:

$$k(x_i, x_j) = \langle \varphi(x_i), \varphi(x_j) \rangle, \quad \varphi : X \to H$$

where X is the input space and H is the feature space.

Literature on kernel methods proposes centring kernel functions. Let k_α represent the centred kernel function:

$$k_\alpha(x_i, x_j) = \langle \varphi(x_i) - \frac{1}{n}\sum_{i=1}^{n}\varphi(x_i), \varphi(x_j) - \frac{1}{n}\sum_{i=1}^{n}\varphi(x_i)\rangle$$

$$= k(x_i, x_j) - \frac{1}{n}\sum_{i=1}^{n}k(x_i, x_j) - \frac{1}{n}\sum_{j=1}^{n}k(x_i, x_j) + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}k(x_i, x_j)$$

When data in feature space is far from the origin, kernel matrix elements become nearly identical, resulting in an ill-conditioned kernel matrix. Centring addresses this issue by ensuring data distributes around the origin. From this perspective, matrix factorization-based word vector methods construct feature spaces through similarity functions. Centring makes words distribute around the origin in feature space, increasing variation among similarity matrix elements and effectively distinguishing degrees of word similarity.

The centring method for similarity matrix A involves transforming each element by subtracting its row and column means and adding the overall matrix mean. The centred similarity matrix A^c is calculated as:

$$A^c = A - V_{r,\cdot} - V_{\cdot,c} + V$$

where $V_{r,\cdot}$ contains row means, $V_{\cdot,c}$ contains column means, and V contains the overall mean.

In matrix form:

$$A^c = A - \frac{1}{n}AJ - \frac{1}{n}JA + \frac{1}{n^2}JAJ$$

where I is the identity matrix and J is an $n \times n$ matrix of all ones, i.e., $J = [1]_{ij}$.

---

3 Experiments

3.1 Experimental Setup

We use the 2015 English Wikipedia corpus as our training set, containing 3,991,454 articles. During preprocessing, all words are converted to lowercase and punctuation is removed. Due to the large vocabulary, we filter out high-frequency and low-frequency words, retaining words with frequencies between 10^{-6} and 10^{-5}, resulting in a vocabulary of 30,946 words.

Word vector quality is evaluated through word similarity tasks by computing similarity between word pairs. We use both cosine similarity and dot product

similarity. Dot product similarity, derived from equation (4), effectively reveals the relationship between similarity matrices and word vector quality. Evaluation employs Spearman correlation coefficient to compare computed similarities with human-annotated datasets. We use two publicly available datasets: WS-353, comprising 353 common word pairs annotated for similarity across nouns, verbs, and adjectives; and RW, containing rare or morphologically complex word pairs from Stanford.

Comparison methods include the five matrix factorization-based approaches from , plus Skip-gram and GloVe. All models train 100-dimensional vectors with a window size of 5.

3.2.1 Word Similarity Experiments

Results across different datasets and similarity measures are presented in through .

Key findings:

a) **Centring consistently improves performance**: Centred similarity matrices yield better word vectors than non-centred ones. In WS-353, centred vectors outperform non-centred vectors for both dot product and cosine similarity (Tables 4 and 6). RW dataset shows similar improvements (Tables 5 and 7).

b) **Greater improvement with dot product similarity**: In WS-353, centring improves performance by 0.2896 with dot product similarity versus 0.2174 with cosine similarity. In RW, improvements are 0.1801 and 0.0528 respectively.

c) **Dataset-dependent optimal similarity measure**: In WS-353, dot product similarity achieves better results (best score: 0.6401) than cosine similarity (0.6257) across all models. Conversely, in RW, cosine similarity performs better (0.3351) than dot product similarity (0.3098).

d) **Optimal models**: In WS-353, the GCE model with centring (GCE-C) performs best, achieving 0.6401 and 0.6257. In RW, the TCE model with centring (TCE-C) excels, reaching 0.3098 and 0.3351.

compares GCE-C and TCE-C against Skip-gram and GloVe. In WS-353, both centred models surpass Skip-gram and GloVe, with GCE-C achieving the highest scores. In RW, both centred models outperform GloVe, with TCE-C performing comparably to Skip-gram.

3.2.2 Relationship Between Pre-Reduction Similarity Matrix and Word Vector Quality

To investigate the relationship between pre-dimensionality-reduction similarity matrices and word vector quality, we analyze RW dataset results further. We compute Pearson correlation coefficients between the pre-reduction similarity

matrix and human-annotated similarities as a quality metric for the similarity matrix.

[Figure 3: see original paper] through [Figure 6: see original paper] plot these Pearson values against Spearman correlation coefficients for word vector quality (using dot product and cosine similarity). The results show a linear relationship: higher correlation between the pre-reduction similarity matrix and human annotations corresponds to better word vector quality. Comparing pre- and post-centring plots reveals stronger linear correlation after centring, indicating that centred similarity matrices better align with human judgments and are more reasonable, thereby improving word vector quality.

presents Pearson correlation coefficients between similarity matrix quality (Pearson values) and word vector quality (Spearman values). Post-centring values exceed pre-centring values for both similarity measures, confirming that centred similarity matrices better match human-annotated similarities and enhance the linear relationship with word vector quality.

---

4 Conclusion

This paper proposes a word vector method based on centring the similarity matrix. By centring the similarity matrix derived from word-context co-occurrence before dimensionality reduction, we obtain improved word vectors. Experiments on WS-353 and RW datasets validate the effectiveness of this approach.

Key conclusions:

a) Pre-dimensionality-reduction similarity matrix quality linearly correlates with word vector quality: better alignment with human-annotated similarities yields better word vectors.

b) Centring improves similarity matrix quality, consequently enhancing word vector quality.

Based on these findings, similarity matrix quality is the critical determinant of word vector quality in matrix factorization-based methods. Therefore, constructing high-quality similarity matrices should be the primary focus. A promising future direction involves semi-supervised approaches to guide similarity matrix values toward external knowledge, further improving word vector quality.

References

[1] Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning [C]// Proc of the 25th International Conference on Machine Learning. New York: ACM Press, 2008: 160-167.

[2] Bengio Y, Schwenk H, Senécal J, et al. Neural probabilistic language models [J]. Journal of Machine Learning Research, 2003, 3 (6): 1137-1155.

[3] Lai S, Liu K, He S, et al. How to Generate a Good Word Embedding [J]. IEEE Intelligent Systems, 2016, 31 (6): 5-14.

[4] 于东, 荀恩东. 基于 Word Embedding 语义相似度的字母缩略术语消歧 [J]. 中文信息学报, 2014, 28 (5): 51-59.

[5] Socher R, Perelygin A, Wu J Y, et al. Recursive deep models for semantic compositionality over a sentiment treebank [C]// Empirical Methods in Natural Language Processing. 2013: 1631-1642.

[6] Kim Y. Convolutional neural networks for sentence classification [C]// Empirical Methods in Natural Language Processing. 2014: 1746-1751.

[7] Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch [J]. Journal of Machine Learning Research, 2011, 12 (1): 2493-2537.

[8] Santos C N D, Gattit M. Deep convolutional neural networks for sentiment analysis of short texts [C]// Proc of International Conference on Computational Linguistics. New York: ACM Press, 2014: 69-78.

[9] 袁书寒, 向阳. 词汇语义表示研究综述 [J]. 中文信息学报, 2016, 30 (5): 1-12.

[10] Pennington J, Socher R, Manning C. Glove: global vectors for word representation [C]// Empirical Methods in Natural Language Processing. 2014: 1532-1543.

[11] Deerwester S, Dumais S T, Furnas G W, et al. Richard Harshman indexing by latent semantic analysis [J]. Journal of the American Society for Information Science, 1990, 41 (6): 391-407.

[12] Caron J. Experiments with LSA scoring: optimal rank and basis [J]. History of Education Quarterly, 2001, 50 (2): 182–203.

[13] Kevin L, Curt B. Producing high-dimensional semantic spaces from lexical co-occurrence [J]. Behavior Research Methods, Instrumentation, and Computers, 1996, 28 (2): 203–208.

[14] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space [C]// Proc of International Conference on Machine Learning. New York: ACM Press, 2013.

[15] Mikolov T, Yih S W, Zweig G. Linguistic regularities in continuous space word representations [C]// North American Chapter of the Association for Computational Linguistics. 2013: 746-751.

[16] Levy O, Goldberg Y, Dagan I. Improving distributional similarity with lessons learned from word embeddings [J]. Bulletin De La Société Botanique De France, 2015, 75 (3): 552-555.

[17] Levy O, Goldberg Y. Neural word embedding as implicit matrix factorization [C]// Advances in Neural Information Processing Systems. 2014: 2177-2185.

[18] Li Y, Xu L, Tian F, et al. Word embedding revisited: a new representation learning and explicit matrix factorization perspective [C]// Proc of International Joint Conference on Artificial Intelligence. [S. l.]: AAAI Press, 2015: 3650-3656.

[19] Lebret R, Collobert R. Word embeddings through Hellinger PCA [C]// Proc of Conference of the European Chapter of the Association for Computational Linguistics. 2014: 482-490.

[20] Lebret R, Collobert R. Rehabilitation of count-based models for word vector representations [C]// Proc of Conference on Intelligent Text Processing and Computational Linguistics. 2015: 417-429.

[21] Turney, Peter D, Pantel, et al. From frequency to meaning: vector space models of semantics [J]. Journal of Artificial Intelligence Research, 2010, 37 (1): 141-188.

[22] Choi S S, Cha S H, Tappert C C. A survey of binary similarity and distance measures [J]. Journal of Systemics Cybernetics & Informatics, 2010, 8 (1): 43-48.

[23] Bullinaria J A, Levy J P. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD [J]. Behavior Research Methods, 2012, 44 (3): 890-907.

[24] Österlund A, Ödling D, Sahlgren M. Factorization of latent variables in distributional semantic models [C]// Empirical Methods in Natural Language Processing. 2015: 227-231.

[25] Marina M A. Data Centering in Feature Space [C]// Proc of the 9th International Workshop on Artificial Intelligence & Statistics. 2003.

[26] 王裴岩, 蔡东风. 基于统计检验的核函数度量方法研究 [J]. 计算机科学, 2015, 42 (4): 199-205.

[27] Sedgwick P. Spearman' s rank correlation coefficient [J]. British Medical Journal, 2014, 349 (nov28 1): g7327.

[28] Rivlin E. Placing search in context: the concept revisited [J]. ACM Trans on Information Systems, 2002, 20 (1): 116–131.

[29] Luong M, Socher R, Manning C D. Better word representations with recursive neural networks for morphology [C]// Computational Natural Language Learning. 2013: 104-113.

[30] 裴楠. 基于计数模型的 Word Embedding 算法研究 [D]. 沈阳: 沈阳航空航天大学, 2017.

Note: Figure translations are in progress. See original paper for figures.