SemGloVe: Semantic Co-Occurrences for GloVe From BERT

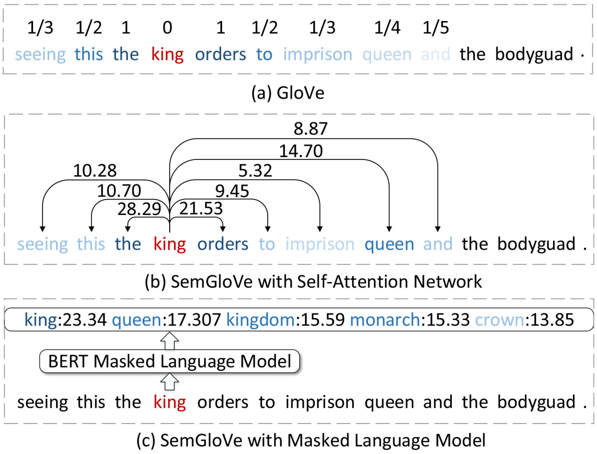
Leilei Gan, Zhiyang Teng, Yue Zhang, *Member, IEEE*, Linchao Zhu, *Member, IEEE*, Fei Wu, and Yi Yang, *Senior Member, IEEE*

***Abstract*—GloVe learns word embeddings by leveraging statistical information from word co-occurrence matrices. However, word pairs in the matrices are extracted from a predefined local context window, which might lead to limited word pairs and potentially semantic irrelevant word pairs. In this paper, we propose *SemGloVe*, which distills *semantic co-occurrences* from BERT into static GloVe word embeddings. Particularly, we propose two models to extract co-occurrence statistics based on either the masked language model or the multi-head attention weights of BERT. Our methods can extract word pairs limited by the local window assumption, and can define the co-occurrence weights by directly considering the semantic distance between word pairs. Experiments on several word similarity datasets and external tasks show that SemGloVe can outperform GloVe.**

***Index Terms*—Word representations, contextual word embeddings, self-attention network, pre-trained language models.**

I. Introduction

Word embeddings [1], [2], [3] represent words with low-dimensional real-valued vectors. They can be useful for lexical semantics tasks, such as word similarity and word analogy, and downstream natural language processing (NLP) tasks [4], [5], [6], [7]. Most existing methods use local window-based methods [2], [3], [8] or matrix factorization of global statistics [9] to learn syntax and semantic information from large-scale corpus. In this paper, we investigate GloVe in details.


1/3 1/2 1 0 1 1/2 1/3 1/4 1/5
seeing this the king orders to imprison queen and the bodyguad .
8.87
14.70
10.28
5.32
10.70
9.45
28.29
21.53
seeing this the king orders to imprison queen and the bodyguad .
king:23.34 queen:17.307 kingdom:15.59 monarch:15.33 crown:13.85
BERT Masked Language Model
seeing this the king orders to imprison queen and the bodyguad .

(a) GloVe

(b) SemGloVe with Self-Attention Network

(c) SemGloVe with Masked Language Model

Fig. 1. Glove (a) and Our two proposed models (b) and (c). The target word and the context word are in red and blue colors, respectively. In (a), the numbers are relevance score, which are position-based distance. In (b) and (c), the relevance scores are the self-attention weights, and the logits of the language model, respectively.

GloVe [9] combines matrix factorization methods with local window context, generating word embeddings by leveraging statistical information from a global word-word co-occurrence matrix. Word-word pairs in the matrix are extracted from a predefined local context window, and the relevance is measured by a position-based distance function. For example, as shown in Fig. 1(a), when targeting at the word "king" with the five-words context window, the word pair "king-queen" can be extracted with a co-occurrence score 1/4, since "queen" is four words away from "king". However, this co-occurrence matrix generation procedure can suffer from two potential problems. First, the counting word pairs are limited by the local context window. Second, the heuristic weighting function does not measure the relevance score directly regarding to the semantic similarities between word pairs, leading to inaccurate co-occurrence counts. For example, "king" and "orders" are weakly correlated in the sentence "he orders a copy of the king of fighters".

One potential solution to the above problems is BERT [10], which is a pre-trained language model based on deep bidirectional Transformers [11]. Previous work has shown that the contextualized representations produced by BERT capture morphological [12], [13], lexical [12], syntactic [14], [15] and semantic knowledge [16], [17]. These knowledge can be disentangled using knowledge distillation models [18], [19] or variational

Manuscript received 23 December 2021; revised 3 May 2022 and 21 June 2022; accepted 2 August 2022. Date of publication 8 August 2022; date of current version 22 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62037001, in part by the Key R&D Projects of the Ministry of Science and Technology under Grant 2020 YFC 0832500, in part by the Zhejiang Province Key Project under Grant 2022SDXHDX0003, in part by the Zhejiang Province Science and Technology Project under Grant 2022C01044, in part by the Starry Night Science Fund of Zhejiang University Shanghai Institute for Advanced Study under Grant SN-ZJU-SIAS-0010, and in part by CAAI-Huawei MindSpore Open Fund under Grants CAAIXSJLJJ-2021-015A and CAAIXSJLJJ-2021-046A. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zoraida Callejas. *(L. Gan, Z. Teng and Y. Zhang contributed equally to this work.) (Corresponding authors: Fei Wu; Yi Yang).*

Leilei Gan and Yi Yang are with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310007, China (e-mail: 11921071@zju.edu.cn; yangyics@zju.edu.cn).

Zhiyang Teng and Yue Zhang are with the School of Engineering, Westlake University, Hangzhou 310024, China, and also with the Institute of Advanced Technology, Westlake Institute for Advanced Study, Hangzhou 310024, China (e-mail: tengzhiyang@westlake.edu.cn; yue.zhang@wias.org.cn).

Linchao Zhu is with the Centre for Artificial Intelligence, University of Technology Sydney, Sydney 2007, Australia (e-mail: linchao.zhu@uts.edu.au).

Fei Wu is with the Shanghai Institute for Advanced Study of Zhejiang University, Shanghai AI Laboratory, Hangzhou , China (e-mail: wufei@zju.edu.cn).

Digital Object Identifier 10.1109/TASLP.2022.3197316

inference [14]. Inspired by these research ideas, we hypothesize that the word-word co-occurrence matrix can be distilled from BERT.

We name the co-occurrences distilled from BERT as *semantic co-occurrences*, thus proposing *SemGloVe* to improve GloVe. Our proposed method can distill contextualized semantic information from BERT into static word embeddings. In particular, we present two models. As shown in Fig. 1(b), the first model is based on self-attention networks of BERT (SAN; Section III-A). The idea is to use attention values in SAN for semantic co-occurrence counts. Intuitively, the SAN model can solve the second issue of GloVe by using the self-attention weights as the word pair scoring function. For example, the relevance score of the "king-queen" pair in Fig. 1(b) is 14.7, which is much larger than the 9.45 score of the "king-to" pair even though "to" is closer to "king" than "queen" is.

As shown in Fig. 1(c), the second is based on the masked language model (MLM; Section III-B) of BERT. The idea is to distill word probabilities from a masked language model for co-occurrence counts. First, it generates word pairs by masking the target word to predict context words from the whole vocabulary, which can avoid the local context window restriction of GloVe. The output context words and the target word can be regarded as co-occurring word pairs. For example, in Fig. 1(c), after masking "king," BERT outputs context words such as "queen" and "crown". Both "king-queen" and "king-crown" are regarded as valid co-occurrences even though "crown" does not appear in this sentence. Compared with the local window context, we hypothesize that the distributional hypothesis [20] works better for the masked language model, because words that occur in the same output contexts tend to have similar meanings. Second, the MLM model uses the logits of output words from BERT as the word pair scoring function, which can solve the first problem of GloVe.

Experiments on word similarity datasets show that SemGloVe can outperform GloVe. We also evaluate SemGloVe on external Chunking, Part-Of-Speech (POS) tagging and named entity recognition (NER) tasks, again showing the effectiveness of SemGloVe. Specifically, our analysis shows that SemGloVe has two advantages: 1) it can find semantic relevant word pairs which cannot be captured by GloVe; 2) SemGloVe can generate more accurate global word-word co-occurrence counts compared to GloVe.

SemGloVe gives better averaged results compared with existing state-of-the-art (non-contextualized) embedding methods on both intrinsic and extrinsic evaluation tasks. In addition to its theoretical interest, this leads to three contributions to the research community. First, it enriches the toolbox for computational linguistics research involving word representations that do not vary by the sentence, such as lexical semantics tasks. Second, it adds to the set of input embeddings that are orders of magnitudes faster compared with contextualized embeddings as it only requires training one time and then can be efficiently used for many downstream tasks. Third, SemGloVe provides better static word vectors, which have been shown to be supplementary to contextual word embeddings for downstream tasks, such as text classification task [21] and named entity recognition task [22]. Our code and SemGloVe embeddings are available at https://github.com/leileigan/SemGloVe.

II. Background

In this section, we briefly review GloVe and BERT, which our models are based on.

A. Glove

Given a training corpus, GloVe first obtains the global word-word co-occurrence counts matrix $\mathbf{X}$, whose entries X_{ij} represents the total number of times word $w_j \in V$ occurring in the context of word $w_i \in V$, where V is the word vocabulary of the training corpus. Formally, let $\mathbb{C}(w_i)$ be the set of the local window context of each w_i in the training corpus. GloVe defines X_{ij} with regards to the **position-based distance** between w_i and w_j as:

$$X_{ij} = \sum_{C_k(w_i) \in \mathbb{C}(w_i)} \sum_{w_j \in C_k(w_i)} \text{dis}(w_i, w_j) \tag{1}$$

$$= \sum_{C_k(w_i) \in \mathbb{C}(w_i)} \sum_{w_j \in C_k(w_i)} \frac{1}{|p_j - p_i|}, \tag{2}$$

where $C_k(w_i)$ is a local window context of w_i, p_j and p_i are the positions of w_j and w_i in the context, respectively. Intuitively, words closer to w_i receive larger weights.

Denote the embedding of a target word w_i and the context embedding of a context word w_j as $\boldsymbol{e}_i$ and $\boldsymbol{e}'_j$, respectively. GloVe learns the word vectors by optimizing the following loss function:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij}) \left(\boldsymbol{e}_i^T \boldsymbol{e}'_j + b_i + b'_j - \log X_{ij}\right)^2, \tag{3}$$

where b_i and b_j represent biases for w_i and w_j respectively, and $f(\cdot)$ is a weighting function. $f(\cdot)$ assigns lower weights to less frequent co-occurrences:

$$f(X_{ij}) = \begin{cases} (X_{ij}/x_{\max})^{\alpha} & \text{if } X_{ij} < x_{\max}, \\ 1 & \text{otherwise}, \end{cases} \tag{4}$$

where $x_{\max}$ and α are hyper-parameters.

After training with optimization methods, $\boldsymbol{e}_i + \boldsymbol{e}'_i$ is taken as the final word embeddings for w_i.

B. Bert

BERT is trained from large scale raw texts using masked language modeling task (MLM) with a deep bidirectional Transformer, which consists of multiple self-attention encoder (SAN) layers. Specifically, MLM masks a certain token as a special symbol $\langle\text{MASK}\rangle$ (or a random token) randomly, and predicts the masked token using the contextualized output of the topmost layer.

Formally, given a token sequence $W = \{w_1, w_2, \ldots, w_n\}$, a certain token w_i ($i \in [1 \ldots n]$) is masked. The input layer $\mathbf{H}^0$ and each intermediate layer representation $\mathbf{H}^j$ are defined as:

$$\begin{aligned}\mathbf{H}^0 &= [\boldsymbol{e}_1; \ldots; \boldsymbol{e}_{i-1}; \boldsymbol{e}_i; \boldsymbol{e}_{i+1}; \ldots; \boldsymbol{e}_n] + \mathbf{W}_p \\ \mathbf{H}^j &= \mathbf{SAN\_Encoder}(\mathbf{H}^{j-1}) \;\; j \in [1 \ldots K],\end{aligned} \tag{5}$$

where $\boldsymbol{e}_i$ is the embedding of w_i, $\mathbf{W}_p$ is the position embedding matrix, $\mathbf{SAN\_Encoder}$ is SAN based encoder and K denotes the number of SAN layers. More details about SAN_ENCODER can be found in [11]. For MLM, BERT predicts w_i using

$$\text{logits}(\mathbf{h}_i^K) = \mathbf{W}\mathbf{h}_i^K \tag{6}$$

$$\mathbf{p}[w_i] = \text{softmax}(\text{logits}(\mathbf{h}_i^K)) \tag{7}$$

where $\mathbf{W}$ is a model parameter and $\mathbf{p}[w_i]$ denotes $P(w_i | w_1, \ldots, w_{i-1}, \langle \text{MASK} \rangle, \mathrm{w_{i+1}}, \ldots, \mathrm{w_n})$.

Given a set of unlabelled text, $D = \{W^i\}|_{i=1}^{N}$, BERT is trained by maximizing the following objective function:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{|W_i|} \mathbf{P}[w_i^j]. \tag{8}$$

III. Semantic GloVe

We replace the hard counts of co-occurrences into real values from BERT, according to self-attention scores and MLM probabilities, respectively.

A. Semantic Co-Occurrences From Multi-Head Self-Attention

In this section, we leverage the multi-head self-attention weights of BERT to measure the semantic relationships of tokens instead of the heuristic position-based distance function of GloVe.

Specifically, given a sentence $W = \{w_1, \ldots, w_K\}$ from the training corpus, a window size S and a pre-trained BERT model, we wish to define the word-to-word semantic distance (i.e., the *dis* function in Equation (2)) using the self-attention weights of BERT. Since BERT uses word pieces or byte-pair encodings (BPE; [23]) to segment words into BPE tokens, we firstly convert the original BPE-to-BPE attention weights to word-to-word attention weights.

Let the corresponding BPE token sequence is $T = \{t_1, \ldots, t_L\}$, N and M be the number of layers and heads of the BERT model, and $\mathbf{AT}_{ij} \in \mathbb{R}^{L \times L}$ be the BPE token attention weights matrix of the j-th head in the i-th layer. We sum all heads and layers attention weights into one BPE-to-BPE attention weight matrix as follows:

$$\mathbf{AT} = \sum_{i=1}^{N} \sum_{j=1}^{M} \mathbf{AT}_{ij}. \tag{9}$$

where $\mathbf{AT} \in \mathbb{R}^{L \times L}$ is the averaged token-to-token attention weight matrix.

Then, we generate the word-to-word attention weight matrix $\mathbf{AW} \in \mathbb{R}^{K \times K}$ by averaging the BPE-to-BPE attention weights. For w_j within the local window context of word w_i, $j \in [i - S, i + S] \cap j \neq i$, we denote the attention weight (a real value) from word w_i to w_j as AW_{ij} following:

$$AW_{ij} = \frac{1}{m \times n} \sum_{k=s_1}^{s_m} \sum_{l=t_1}^{t_n} \mathbf{AT}(k, l), \tag{10}$$

where $\mathbf{AT}(k, l)$ is the attention weight from BPE token t_k to t_l, m and n are the number of subwords of w_i and w_j, respectively. To remove semantic irrelevant words, we sort AW_{ij} in descending order, and select the top-S words as w_i's context words $C(w_i)$.

For now, values in $\mathbf{AW}$ are still raw attention weights extracted from a pre-trained model, which are unnormalized and are not appropriate for training GloVe. To normalize the distance between the target word w_i and the context word w_j within 0 and 1, the following *Division* distance function is introduced:

$$\text{dis}_{sa}(w_i, w_j) = AW_{ij} / AW_{i1}. \tag{11}$$

Finally, for the whole corpus, elements in the global word-to-word co-occurrence count matrix $\mathbf{X}$ is accumulated as:

$$X_{ij} = \sum_{C_k(w_i) \in \mathbb{C}(w_i)} \sum_{w_j \in C_k(w_i)} \text{dis}_{sa}(w_i, w_j). \tag{12}$$

where $C_k(w_i)$ is the context words of w_i defined by the multi-head self-attention weights, and $\mathbb{C}(w_i)$ is the set of each w_i's context in the training corpus.

We name the GloVe model training on this semantic word-to-word co-occurrence counts as SemGloVe$_{sd}$.

B. Semantic Co-Occurrences From Masked Language Model

BERT pre-trained using deep bidirectional Transformers with the MLM task provides a dynamic way to define context for target word, which can avoid semantic irrelevant word pairs from a local window context. In this section, we introduce another method to distill semantic word-word co-occurrence counts by leveraging the MLM task of BERT.

Formally, given a sentence $W = \{w_1, \ldots, w_K\}$ from the training corpus, the corresponding BPE token sequence $T = \{t_1, \ldots, t_L\}$, a window size S and the topmost layer output $\mathbf{h}_i^K$ of the $\mathbf{SAN\_Encoder}$, we define the context tokens $\mathcal{C}(w_i)$ as the output tokens of the MLM of BERT, and calculate the word-to-word semantic distance (i.e., the *dis* in Equation (2)) using the logits in Equation (6).

Specifically, we first generate the BPE-to-BPE co-occurrence matrix $\mathbf{M} \in \mathbb{R}^{L \times L}$ to deal with the same BPE problem in Section III-A. To be more specific, we feed the original BPE token sequence T into the MLM model and sort the obtained output tokens in descending order with respect to logits$(\mathbf{h}_i^K)$, and then select the top $2S$ tokens to constitute t_i's context tokens, denoted as $C(t_i) = \{t_i^{(1)}, t_i^{(2)}, \ldots, t_i^{(2S)}\}$. The corresponding logits are denoted as $G(t_i) = \{g_i^{(1)}, g_i^{(2)}, \ldots, g_i^{(2S)}\}$. It is worth noting that token t_i actually is not masked. In preliminary experiments, we did try to mask t_i in the input sentence, however, this strategy did not give a better result. We hypothesize that the information carried by t_i is useful to predict the most similar co-occurring words. If masking t_i, more noisy word pairs will be introduced.

Similar in Section III-A, the distance between the target token t_i and the context token $t_i^{(j)}$ is calculated as:

$$\text{dis}_{mlm}(t_i, t_i^{(j)}) = g_i^{(j)}/g_i^{(1)}. \tag{13}$$

Then, for the whole corpus, elements in the global BPE-to-BPE co-occurrence matrix $\mathbf{M}$ is calculated as:

$$M_{ij} = \sum_{C_k(t_i)\in\mathbb{C}(t_i)} \sum_{t_j\in C_k(t_i)} \text{dis}_{mlm}(t_i, t_j). \tag{14}$$

where $C_k(t_i)$ is the context of t_i defined by the masked language model, and $\mathbb{C}(t_i)$ is the set of each t_i's context in the training corpus.

Finally, we generate the global word-to-word co-occurrence matrix by averaging the BPE-to-BPE matrix as:

$$X_{ij} = \frac{1}{m\times n}\sum_{k=s_1}^{s_m}\sum_{l=t_1}^{t_n} M_{kl}, \tag{15}$$

where m and n are the number of subwords of w_i and w_j, respectively. Similar ideas can be found in [24], [25] that leverages MLM to construct synonym dictionary.

The GloVe model trained on this kind semantic co-occurrences is named as SemGloVe$_{md}$.

IV. Experiments

We compare SemGloVe with GloVe on a range of intrinsic and extrinsic evaluation tasks, discussing the role the semantic co-occurrence plays in the training process.

A. Settings

To give fair comparisons with previous methods, we follow [26] to use the Wikipedia[1] dump corpus as our training dataset, which is processed to only keep words appearing more than five times. The dataset consists of 57 million sentences and 1.1 billion tokens.

For all our experiments, we set $x_{\max} = 10$, $\alpha = 0.75$, the vectors dimension to 300, the window size S to 5, and the number of iteration to 100. AdaGrad [27] is used as the optimizer with initial learning rate $lr = 0.05$. We dump the weights using the uncased BERT-large model with whole word masking, which has 24 layers and 12 attention heads[2]. The code is implemented by PyTorch and MindSpore.

B. Baselines and Evaluation Methods

In addition to GloVe, we also compare SemGloVe with other state-of-the-art embedding methods, including Word2Vec [3], Deps [28], Fasttext [8], SynGCN and SemGCN [26], on several intrinsic and extrinsic semantic evaluation tasks. In addition to the GloVe baseline under the same settings, we also download GloVe$_{\text{6B}}$ from the official website as one baseline, which is trained on 6 billion tokens[3] and contains five times more tokens than the Wikipedia dump we used.

We evaluate the intrinsic task on word similarity datasets, including WordSim-353 [29], and SimLex-999 [30]. Spearman correlation is taken as the main metric.

For extrinsic evaluation, we build a sentence-state LSTM (S-LSTM) [31] based sequence labeling model which takes the concatenation of ELMo [32] and GloVe or SemGloVe as inputs. S-LSTM is a variant of LSTM, which parallelly updates the states of words by exchanging information locally and globally. The number of parameters of S-LSTM is 29.8 M. We compare this method with BERT$_{\text{base}}$ which has 110 M parameters on a range of tasks, including Chunking, POS tagging and Named Entity Recognition (NER). The datasets used for evaluation are CoNLL-2000, the Wall Street Journal (WSJ) portion of the Penn Tree Bank (PTB) and CoNLL-2003, respectively.

C. Development Experiments

Development experiments are conducted on the WS353S dataset to compare the performance of GloVe and SemGloVe with different hyper-parameters settings. To be specific, we investigate the influence of vector dimension, corpus size and $x_{\max}$ value on the performance.

1) Effect of Vector Dimension: We evaluate GloVe and SemGloVe with different vector dimensions, ranging from 50 to 400. For dimensions smaller than 300, the iteration number is set to 50, otherwise, it is set to 100. As shown in Fig. 2(a), SemGloVe outperforms GloVe under different vector dimension settings. When increasing the vector size from 50 to 300, both GloVe and SemGloVe can improve the performance on the WS353S dataset. However, when further increasing the vector size to 400, GloVe has a slight decrease in performance while SemGloVe$_{md}$ can keep the results stable. For a fair comparison, we choose 300 as the final vector size.

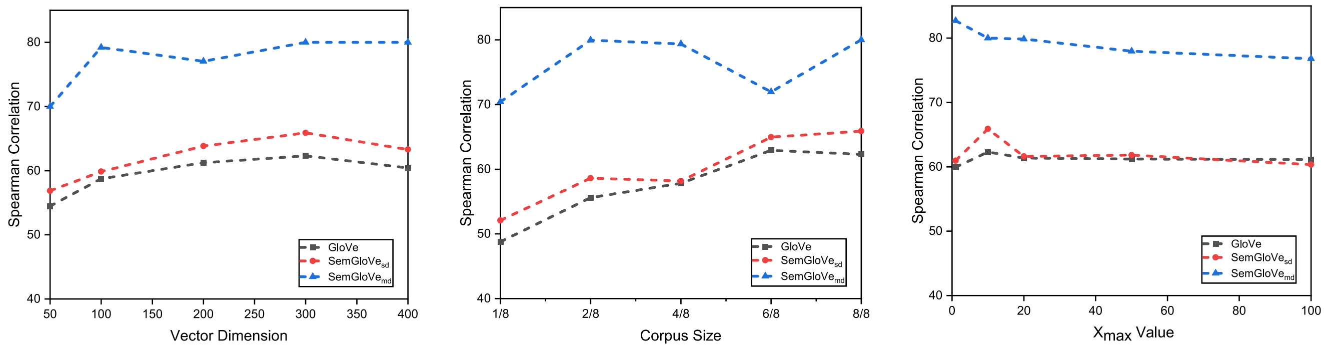
2) Effect of Corpus Size: To investigate the influence of corpus size, we divide the whole training corpus into 8 parts. GloVe and SemGloVe are trained on 1/8, 2/8, 4/8, 6/8 and 8/8 corpus parts, respectively. According to the corpus size, $x_{\max}$ is set to 1.25, 2.5, 5, 7.5 and 10, respectively. As shown in Fig. 2(b), SemGloVe outperforms GloVe in all corpus size settings. As the corpus size increases, GloVe and SemGloVe$_{sd}$ obtain better performance. In the meantime, SemGloVe$_{md}$ can achieve reasonable results even with a small corpus size. We suppose that SemGloVe$_{md}$ can capture sufficient word-word co-occurrences counts under small corpus size, which we will analyze in Section IV-E.

3) Effect of x_{max} Value: To evaluate the effect of $x_{\max}$, we train our models with different $x_{\max}$ values. As shown in Fig. 2(c), SemGloVe outperforms GloVe under different $x_{\max}$ values. GloVe and SemGlove$_{sd}$ achieve the best results when $x_{\max}$ is set to 10, while SemGloVe$_{md}$ maintains strong and stable results with different $x_{\max}$ values. The reasons we suppose can be explained as follows. As indicated by Equation 3 and 4, co-occurrences with frequencies smaller than $x_{\max}$ will be assigned lower weights for training, thus for window based contexts which are not able to capture high-quality co-occurrences, the choice of $x_{\max}$ is important for the final performance. A large $x_{\max}$ value will lead to the neglect of useful word-word

[1] https://dumps.wikimedia.org/enwiki/20180301

[2] https://github.com/google-research/bert

[3] https://nlp.stanford.edu/projects/glove/



(a) Vector Dimension (b) Vector Dimension (c) $X_{\max}$ Value

Fig. 2. Performance on WS353S dataset as function of vector dimension, corpus size and $x_{\max}$ value.(a) Vector Dimension (b) Vector Dimension (c) $X_{\max}$ Value.

TABLE I
INTRINSIC EVALUATION: COMPARISON ON WORDSIM-353 (WS353) AND SIMLEX-999 (SIMLEX999) DATASETS. WS353S AND WS353R ARE SUBSETS OF WORDSIM-353, WHICH ARE USED TO MEASURE THE RELATEDNESS AND SIMILARITY OF WORDS, RESPECTIVELY

Models	WS353	WS353S	WS353R	SimLex999	Average
Word2Vec	61.6	69.2	54.6	35.1	52.6
Deps	60.6	65.7	36.2	39.6	50.5
Fasttext	68.3	74.4	65.4	34.9	60.8
SemGCN	60.9	65.9	60.3	**48.8**	59.0
SynGCN	60.9	73.2	45.7	45.5	56.3
GloVe	52.6	62.3	55.5	28.2	49.7
GloVe$_{6B}$	56.5	64.6	50.0	31.3	50.6
SemGloVe$_{sd}$	56.0	65.9	56.3	31.9	52.5
SemGloVe$_{md}$	**69.7**	**80.0**	**68.9**	46.5	**66.3**

The bold results are the best results.

co-occurrences, while a small value will bring more noises. However, for SemGloVe$_{md}$ which captures context from deep bidirectional Transformers, there exists less noisy co-occurrences. As a result, the performance can maintain stable with different $x_{\max}$ values. We set $x_{\max}$ to 10 in the remaining experiments.

D. Final Results

1) Intrinsic Evaluation Results: The final intrinsic evaluation results of SemGloVe and the baselines are listed in Table I. First, we find that SemGloVe$_{sd}$ and SemGloVe$_{md}$ outperform GloVe on the four word similarity evaluation datasets. Specifically, SemGloVe$_{sd}$ obtains 5.6% absolute increase in performance on average, which demonstrates that self-attention weights of BERT are better to measure word similarities than the original position-based distance method. In addition, compared with GloVe, SemGloVe$_{md}$ gives 33.4% absolute increase in performance on average, which demonstrates that masked language model of BERT can model the context of words better than the predefined local window context, and produce more semantic relevant word pairs. Moreover, We also find that the averaged results of SemGloVe$_{md}$ outperform all the best methods in the literature.

TABLE II
EXTRINSIC EVALUATION: COMPARISONS ON CHUNKING, POS TAGGING AND NER TASKS ON CONLL-2000, THE WALL STREET JOURNAL (WSJ) PORTION OF THE PENN TREE BANK (PTB) AND CONLL-2003. THE METRICS FOR THE THREE TASKS ARE F1-VALUE, ACCURACY AND F1-VALUE, RESPECTIVELY

Models	Chunking	POS	NER	Average
BERT$_{base}$	96.60	**97.77**	92.40	95.59
GloVe	96.72	97.65	92.27	95.21
SemGloVe	**96.75**	97.70	**92.60**	**95.68**

The bold results are the best results.

Since SemGloVe$_{md}$ performs better than SemGloVe$_{sd}$ on intrinsic tasks, we take them as the final SemGloVe.

2) Extrinsic Evaluation Results: The final extrinsic evaluation results are shown in Table II. We find that SemGloVe outperforms GloVe on three tasks, which demonstrates that SemGloVe contains more semantic information for downstream tasks. Specifically, SemGloVe gives 0.47% absolute increase in performance on average, respectively. Furthermore, compared with contextual word representations BERT$_{\text{base}}$, SemGloVe also shows competitive performance.

The comparisons on extrinsic tasks is to show that SemGloVe (static word vectors) is still useful for downstream tasks, instead

TABLE III
Some Specific Word-Word Co-Occurrence Counts of GloVe and SemGloVe

Methods	GloVe	SemGloVe$_{sd}$	SemGloVe$_{md}$
⟨ cat, cats ⟩	0.33×10^2	0.86×10^2	3.48×10^4
⟨ paris, french ⟩	8.23×10^2	1.41×10^3	5.21×10^4
⟨ man, woman ⟩	1.48×10^3	2.57×10^3	1.25×10^5
⟨ himself, oneself ⟩	0	0	4.42×10^4
⟨ meanwhile, meantime ⟩	0	0	2.72×10^4
⟨ additionally, moreover ⟩	0	0	2.90×10^4
⟨ biophysical, biochemical ⟩	12.53	12.93	1.16×10^2
⟨ egyptologist, archaeologist ⟩	15.00	17.30	0.39×10^2
⟨ egyptologist, egyptian ⟩	11.32	8.89	8.58×10^3

of showing the weakness of contextualized embeddings as input representations. Another advantage of GloVe and SemGloVe, as compared with BERT, is that they are light weight and much faster for training and testing for downstream tasks.

E. Analysis

We give analysis to the generated word-word co-occurrence counts of GloVe and SemGloVe, and try to answer the following research questions: (1) Can SAN weights of BERT be used to measure semantic similarity between word pairs, and lead to more accurate co-occurrence counts? (2) Can masked language model of BERT generate more semantic relevant word pairs? (3) Can the proposed *Division* distance function be better than the original position-based distance function?

Co-occurrence Counts Analysis: To answer above questions, we analyze the difference of co-occurrence counts between GloVe and SemGloVe from two aspects, some specific word pairs and the general statistical information. First, we investigate some representative co-occurrence counts. As shown in Table III, word pairs of the first three rows, which are close in semantic, such as <paris, french>, can be found in all five methods. However, SemGlove$_{sd}$ can generate larger co-occurrence counts compared with GloVe, which proves that self-attention weights of BERT can assign larger counts to more semantic similar word pairs within the context. In fact, SemGloVe$_{md}$ can reach an average of 50 times counts of GloVe or SemGloVe$_{sd}$. We suppose that the words in a window context can be diverse and noisy, and thus distract the total co-occurrence counts. This helps to explain why SemGloVe$_{md}$ trained on small size corpus can still achieve strong performance. Furthermore, SemGloVe$_{md}$ can generate semantic relevant word pairs in the second three rows, which cannot be found in GloVe or SemGloVe$_{sd}$. This is because the two methods generate word pairs from a local window, while SemGloVe$_{md}$ can output most similar words over the whole vocabulary based on a deep bidirectional context.

As shown in Equation 14 and Equation 15, SemGloVe $_{md}$ firstly calculate the BPE-to-BPE co-occurrence count and then generate the word-to-word co-occurrence matrix by averaging the counts of the subword pairs. We hypothesize this design can lead to rich semantic word pairs of SemGloVe$_{md}$ because word pairs with common subwords will be connected and obtain a large co-occurrence count. We give some examples in the third three rows of Table III. For instance, word "egyptologist" consists of two subwords " Egypt" and "##ologist," which lead to rich semantic word pairs, < egyptologist, egyptian > and < egyptologist, archaeologist>, respectively.

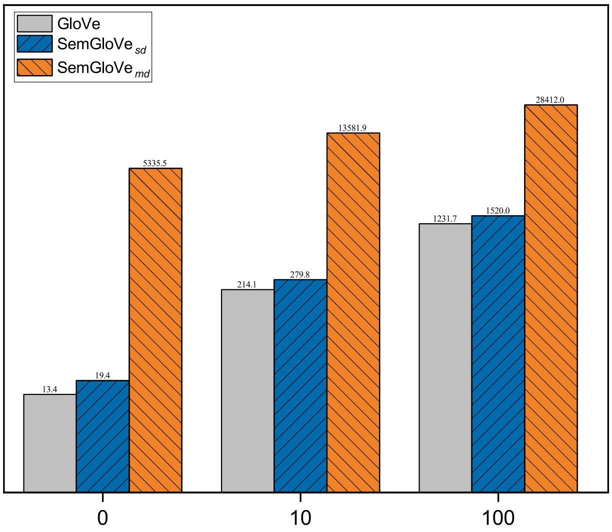


Fig. 3. Average word-word co-occurrence counts of GloVe and two variants of SemGloVe.

Second, we explore the average word-to-word co-occurrence counts of GloVe and SemGloVe by dividing the counts into three groups. The minimum counts of the first, second and third groups are 0, 10, 100, respectively. The average count of each group is equal to the total counts divided by the number of word pairs. As shown in Fig. 3, the average counts of SemGloVe$_{sd}$ are larger than GloVe in all groups. In addition, the average counts of SemGloVe$_{md}$ surpass the number of GloVe and GloVe$_{sd}$ by a large amount in all three groups. This again shows that the SAN weights of BERT can help to determine the semantic distance between words, while MLM can generate semantic relevant word pairs without limiting by the local window context.

Effect of Distance Functions: Different from the position-based distance function of GloVe, our *Division* distance function is based on either the weights of SAN or the output logits of MLM. To investigate whether our distance function captures

TABLE IV
COMPARISONS OF DIFFERENT DISTANCE FUNCTION

Models	WS353	WS353S	WS353R	SimLex999	Average
GloVe	52.6	62.3	55.5	28.2	49.7
SemGloVe$_{sr}$	54.0	62.8	54.9	29.3	50.3
SemGloVe$_{sd}$	56.0	65.9	56.3	31.9	52.5
SemGloVe$_{mr}$	67.8	79.2	66.7	45.1	64.7
SemGloVe$_{md}$	**69.7**	**80.0**	**68.9**	**46.5**	**66.3**

The bold results are the best results.

TABLE V
ANALYSIS OF WINDOW SIZE. THE NUMBERS IN THE PARENTHESES ARE THE TOTAL WORD PAIRS

Models	WS353	WS353S	WS353R	SimLex999	Average
GloVe(0.3B)	52.6	62.3	55.5	28.2	49.7
SemGloVe$_{sd}$(0.2B)	56.0	65.9	56.3	31.9	52.6
SemGloVe$_{sd10}$(0.3B)	55.3	63.2	56.4	28.6	50.9
SemGloVe$_{sdR}$(0.2B)	52.7	60.5	54.6	27.7	48.9
SemGloVe$_{md5}$	66.1	78.6	67.0	45.3	64.3
SemGloVe$_{md}$	**69.7**	**80.0**	**68.9**	**46.5**	**66.3**

The bold results are the best results.

more semantic knowledge, we first sort the context words according to their SAN weights or MLM logits in descending order, and then directly replace the *Division* function of SemGloVe with position-based distance function, and name the two corresponding method as SemGloVe$_{sr}$ and SemGloVe$_{mr}$. The comparisons are listed in Table IV. We can observe that the *Division* distance function performs better than the original distance function of GloVe in mapping BERT's weights into co-occurrence counts, which can be because that the $Division$ function leverage information of weights better than the original function.

Effect of Window Sizes: Although SemGloVe with SAN and GloVe have the same window size, the context words in GloVe are twice as many as those in SemGloVe. This is because the context of GloVe is symmetric. In SemGloVe with SAN, in fact, we sort the scores of all the 10 words in descending order, and then select the top 5 words as context words. To analyze the effect of window size, we take ablation studies on the word similarity dataset as shown in Table V. "SemGloVe$_{sd10}$" is a setting, where we select the top 10 words as context words, while for "SemGloVe$_{sdR}$," we replace the co-occurrence counts of "SemGloVe$_{sd}$" with the corresponding counts of GloVe. We can find that the performance of "SemGloVe$_{sd10}$" is slightly worse than "SemGloVe$_{sd}$," but still better than GloVe. However, "SemGloVe$_{sdR}$" reaches almost the same results as GloVe. We conclude that reducing the window size in SemGloVe$_{sd}$ can reserve valuable word pairs.

We also reduce the window size from 10 to 5 in SemGloVe with MLM, and name this setting as SemGloVe$_{md5}$. As shown in Table V, we can find that reducing the window size in MLM model will decrease the performance.

Effect of Pre-trained Models: Our current SemGloVe is based on BERT-large. To ablate the effectiveness of the proposed method on different pre-trained language models, we conduct experiments based on ALBERT [33]. The results are listed in Table VI. As been shown, SemGloVe based on ALBERT also gives better results compared with GloVe. We also have tried to train SemGloVe based on RoBERTa [34]. However, RoBERTa is a case-sensitive pre-trained language model and its tokenization is also sensitive to whitespace token. These two characteristics increase the difficulty in calculating the word-to-word counts. As a result, in our experiments, SemGloVe based on RoBERTa did not give competitive results compared with those based on BERT and ALBERT.

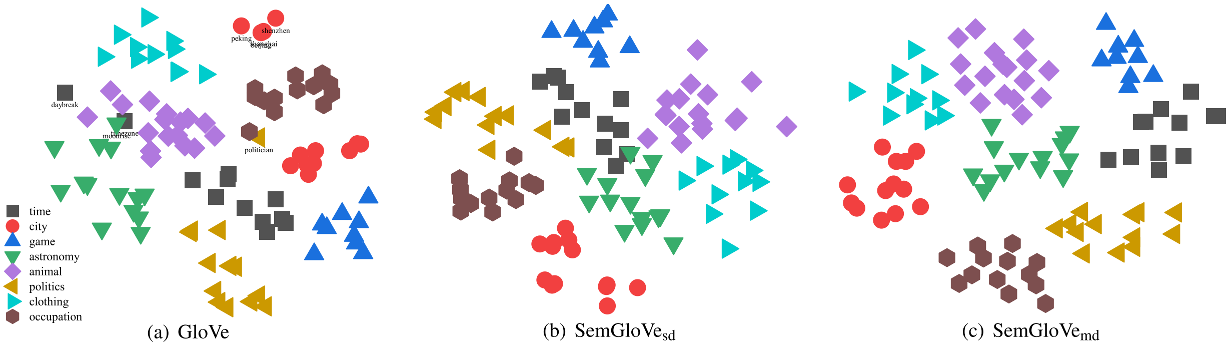
Visualization: We use t-SNE [35] to visualize GloVe, SemGloVe $_{sd}$ and SemGloVe$_{md}$ embeddings. As shown in Fig. 4, GloVe has several outliers for classes of city, time, astronomy and politics. One reason is that for outlier words, such as "daybreak," there are few useful related word pairs in the co-occurrence matrix of GloVe. SemGloVe$_{sd}$ performs better than GloVe, and contains fewer outlier words. However, some classes are not separated clearly. In contrast, SemGloVe$_{md}$ gives the best visualization result among all models. Words from the same class are clustered together, and different classes have clear boundaries. We attribute this to the rich semantic word pairs of SemGloVe$_{md}$. For the same word "daybreak," the co-occurrence matrix of SemGloVe$_{md}$ instead contains semantic relevant word pairs, such as $<$ daybreak, today, $1.54 \times 10^3 >$ and <daybreak, sunday, 3.20×10^3>.

V. RELATED WORK

1) Static Word Representations: Skip-Gram (SG) and Continuous-Bag-of-Words (CBOW) [2], [3] are both local window context based static word vectors, which have been the key components of representation learning and deep learning [36],

TABLE VI
COMPARISONS OF DIFFERENT PRE-TRAINED LANGUAGE MODELS

Models	WS353	WS353S	WS353R	SimLex999	Average
GloVe	52.6	62.3	55.5	28.2	49.7
GloVe$_{6B}$	56.5	64.6	50.0	31.3	50.6
+BERT-large					
SemGloVe$_{md}$	69.7	80.0	68.9	46.5	66.3
+ALBERT-large					
SemGloVe$_{md}$	61.2	72.3	61.4	43.0	59.5



(a) GloVe (b) SemGloVe$_{sd}$ (c) SemGloVe$_{md}$

Fig. 4. Visualization of three methods on randomly selected eight classes of words, including time, city, game, astronomy, animal, politics, clothing and creator. Each class contains ten words on average. (a) GloVe (b) SemGloVe$_{sd}$ (c) SemGloVe$_{md}$.

[37], [38], [39]. The former predicts the context words using the center word, while the latter predicts the center word using its neighbors. [28] improve the word embeddings by injecting syntactic information from the dependency parse trees. Fasttext [8] enriches word embeddings with subwords. Previous work also investigates how to incorporate external semantic knowledge into the above word embeddings. [40] and [41] propose to post-process word vectors using word synonymy or antonymy knowledge from semantic lexicons or task-specific ontology. [26] leverage graph neural networks to incorporate syntactic and semantic knowledge into word embeddings. [42] jointly train word vectors using a text corpora and a knowledge base, which contains special semantic relations. All these methods inject semantic information from structured data, which is expensive to construct, while our methods can benefit from large-scale language models which are pre-trained on unlabelled corpora.

g) Distilling Knowledge From BERT: [18] introduce knowledge distillation, which can transfer knowledge from a large (teacher) network to a small (student) network. Following this idea, [19] propose to distill knowledge from the last layer of BERT into a single-layer BiLSTM network. [43] propose Patient-KD, which not only learns from the last layer of BERT, but also learns from multiple intermediate layers by two strategies: PKD-last and PKD-skip. TinyBERT [44] learns from the embedding layer, the hidden states and attention matrices of intermediate layers, and the logits output of the prediction layer of BERT. [45] improve sequence to sequence text generation models by leveraging BERT's bidirectional contextual knowledge.

There is also a line of concurrent work trying to derive static vectors for words, phrases and sentences from contextual word embeddings. [46], [47], [48] use the deep contextual hidden representations to improve word2vec [2], [3]. The simplest way to derive static vectors for phrases or sentences is max-pooling or mean-pooling over contextual word embeddings. However, these methods rely on lexical overlaps. To alleviate this issue, [49] obtain phrase-level vectors (Phrase-BERT) from BERT which is fine-tuned with a contrastive objective. [50] derive static vectors for sentences from BERT which is fine-tuned on natural language inference dataset using siamese network structures.

Different from all the previous works, in this paper, we distill semantic knowledge from a pre-trained BERT into GloVe using the weights of SAN and MLM.

VI. CONCLUSION

We present SemGloVe, which replaces the hard counts of GloVe by distilling semantic co-occurrences from BERT. Compared with GloVe, SemGloVe can extract word pairs without local window constraints, and can count co-occurrences by directly considering the semantic distance between word pairs. Intrinsic and extrinsic experiments show that SemGloVe outperforms GloVe.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for their comments and suggestions.

REFERENCES

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, pp. 1137–1155, 2003.

[2] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[3] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[4] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, 2011.

[5] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAd: 100,000+ questions for machine comprehension of text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 2383–2392.

[6] M. Lewis, K. Lee, and L. Zettlemoyer, "LSTM CCG parsing," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2016, pp. 221–231.

[7] K. Lee, L. He, and L. Zettlemoyer, "Higher-order coreference resolution with coarse-to-fine inference," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2018, pp. 687–692.

[8] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, no. 1, pp. 135–146, 2017.

[9] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[11] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[12] M. E. Peters, M. Neumann, L. Zettlemoyer, and W.-T. Yih, "Dissecting contextual word embeddings: Architecture and representation," 2018, *arXiv:1808.08949*.

[13] J. Hewitt and C. D. Manning, "A structural probe for finding syntax in word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4129–4138.

[14] X. L. Li and J. Eisner, "Specializing word embeddings (for parsing) by information bottleneck," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 2744–2754.

[15] Y. Goldberg, "Assessing BERT's syntactic abilities," 2019, *arXiv:1901.05287*.

[16] J. Da and J. Kasai, "Cracking the contextual commonsense code: Understanding commonsense reasoning aptitude of deep contextual representations," in *Proc. First Workshop Commonsense Inference Natural Lang. Process.*, 2019, pp. 1–12.

[17] K. Ethayarajh, "How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 55–65.

[18] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[19] R. Tang, Y. Lu, L. Liu, L. Mou, O. Vechtomova, and J. Lin, "Distilling task-specific knowledge from BERT into simple neural networks," 2019, *arXiv:1903.12136*.

[20] Z. S. Harris, "Distributional structure," *Word,*vol. 10, no. 2/3, pp. 146–162, 1954.

[21] I. Alghanmi, L. E. Anke, and S. Schockaert, "Combining BERT with static word embeddings for categorizing social media," in *Proc. 6th Workshop Noisy User-Generated Text*, 2020, pp. 28–33.

[22] X. Wang et al., "Automated concatenation of embeddings for structured prediction," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 2643–2660.

[23] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 1715–1725.

[24] L. Gan et al., "Triggerless backdoor attack for NLP tasks with clean labels," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2022, pp. 2942–2952. [Online]. Available: https://doi.org/10.18653/v1/2022.naacl-main.214

[25] L. Li, R. Ma, Q. Guo, X. Xue, and X. Qiu, "BERT-ATTACK: Adversarial attack against BERT using BERT," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 6193–6202.

[26] S. Vashishth, M. Bhandari, P. Yadav, P. Rai, C. Bhattacharyya, and P. Talukdar, "Incorporating syntactic and semantic information in word embeddings using graph convolutional networks," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 3308–3318.

[27] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, pp. 2121–2159, 2011.

[28] O. Levy and Y. Goldberg, "Dependency-based word embeddings," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, 2014, pp. 302–308.

[29] L. Finkelstein et al., "Placing search in context: The concept revisited," *ACM Trans. Inf. Syst.*, vol. 20, no. 1, pp. 116–131, 2002.

[30] D. Kiela, F. Hill, and S. Clark, "Specializing word embeddings for similarity or relatedness," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 2044–2048.

[31] Y. Zhang, Q. Liu, and L. Song, "Sentence-state LSTM for text representation," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 317–327.

[32] M. Peters et al., "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2018, pp. 2227–2237.

[33] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite BERT for self-supervised learning of language representations," in *Proc. Int. Conf. Learn. Representations*, 2019.

[34] Y. Liu et al., "Roberta: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[35] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[36] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[37] Y. Zhuang, M. Cai, X. Li, X. Luo, Q. Yang, and F. Wu, "The next breakthroughs of artificial intelligence: The interdisciplinary nature of AI," *Engineering*, vol. 6, no. 3, pp. 245–247, 2020.

[38] Y. Pan, "Multiple knowledge representation of artificial intelligence," *Engineering*, vol. 6, no. 3, pp. 216–217, 2020.

[39] N. Lei et al., "A geometric understanding of deep learning," *Engineering*, vol. 6, no. 3, pp. 361–374, 2020.

[40] M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, and N. A. Smith, "Retrofitting word vectors to semantic lexicons," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2015, pp. 1606–1615.

[41] N. Mrkšić et al., "Counter-fitting word vectors to linguistic constraints," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2016, pp. 142–148.

[42] M. Alsuhaibani, D. Bollegala, T. Maehara, and K.-I. Kawarabayashi, "Jointly learning word embeddings using a corpus and a knowledge base," *PLoS one*, vol. 13, no. 3, 2018, Art. no. e0193094.

[43] S. Sun, Y. Cheng, Z. Gan, and J. Liu, "Patient knowledge distillation for BERT model compression," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 4314–4323.

[44] X. Jiao et al., "TinyBERT: Distilling BERT for natural language understanding," 2019, *arXiv:1909.10351*.

[45] Y.-C. Chen, Z. Gan, Y. Cheng, J. Liu, and J. Jing Liu, "Distilling knowledge learned in BERT for text generation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7893–7905.

[46] R. Bommasani, K. Davis, and C. Cardie, "Interpreting pretrained contextualized representations via reductions to static embeddings," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4758–4781.

[47] P. Gupta and M. Jaggi, "Obtaining better static word embeddings using contextual embedding models," 2021, *arXiv:2106.04302*.

[48] Y. Wang, L. Cui, and Y. Zhang, "Improving skip-gram embeddings using BERT," *IEEE/ACM Trans. Audio Speech, Lang. Process.*, vol. 29, pp. 1318–1328, 2021.

[49] S. Wang, L. Thompson, and M. Iyyer, "Phrase-BERT: Improved phrase embeddings from BERT with an application to corpus exploration," 2021, *arXiv:2109.06304*.

[50] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 3982–3992.