entropy

MDPI

Article

Word2vec Skip-Gram Dimensionality Selection via Sequential Normalized Maximum Likelihood

Pham Thuc Hung * and Kenji Yamanishi

Graduate School of Information Science and Technology, The University of Tokyo, Hongo, Bunkyo-ku, Tokyo 113-8654, Japan; yamanishi@mist.i.u-tokyo.ac.jp
* Correspondence: hungpt169@gmail.com

Abstract: In this paper, we propose a novel information criteria-based approach to select the dimensionality of the word2vec Skip-gram (SG). From the perspective of the probability theory, SG is considered as an implicit probability distribution estimation under the assumption that there exists a true contextual distribution among words. Therefore, we apply information criteria with the aim of selecting the best dimensionality so that the corresponding model can be as close as possible to the true distribution. We examine the following information criteria for the dimensionality selection problem: the Akaike's Information Criterion (AIC), Bayesian Information Criterion (BIC), and Sequential Normalized Maximum Likelihood (SNML) criterion. SNML is the total codelength required for the sequential encoding of a data sequence on the basis of the minimum description length. The proposed approach is applied to both the original SG model and the SG Negative Sampling model to clarify the idea of using information criteria. Additionally, as the original SNML suffers from computational disadvantages, we introduce novel heuristics for its efficient computation. Moreover, we empirically demonstrate that SNML outperforms both BIC and AIC. In comparison with other evaluation methods for word embedding, the dimensionality selected by SNML is significantly closer to the optimal dimensionality obtained by word analogy or word similarity tasks.

Keywords: model selection; information criteria; minimum description length; sequentially normalized maximum likelihood; word embedding; word2vec

Citation: Hung, P.T.; Yamanishi, K. Word2vec Skip-Gram Dimensionality Selection via Sequential Normalized Maximum Likelihood. *Entropy* **2021**, *23*, 997. https://doi.org/10.3390/e23080997

Academic Editor: Kevin R. Moon

Received: 14 June 2021
Accepted: 27 July 2021
Published: 31 July 2021

1. Introduction

In recent years, word2vec has been widely applied to many aspects of Natural Language Processing (NLP) and information retrieval such as machine translation [1,2], text classification [3], text summarization [4], and named entity recognition [5]. Furthermore, word2vec is used in various fields such as materials science [6], healthcare [7], and recommendation engines [8–10].

The selection of the dimensionality for word2vec is important with regard to two aspects: model accuracy and computing resources. It is crucial to have a model of dimensionality high enough to learn the regularity of the data, but too high a dimensionality tends to cause overfitting. For instance, the experiment results in [11] demonstrated that the performance of the model in tasks such as Google word analogy, Wordsim353, MTurk771 decreased significantly when the dimensionality increased far from the optimal dimensionality. In addition, a large model is accompanied by a massive number of parameters for storage in the machine during training [12], leading to wasted memory resources. Thus, it is crucial to devise a method that can decide upon a dimensionality that satisfies the ability to capture necessary information from training data as well as makes efficient use of the computational resources.

However, few studies have focused on the dimensionality selection problem. Most research evaluating the effectiveness of word embedding focuses on word analogy and word similarity tasks [13]. These evaluation methods require handcrafted datasets for implementation, but such datasets are currently not available to evaluate model training

on non-English verbal and non-verbal data. To the best of our knowledge, only Yin and Shen [11] accomplished the dimensionality selection of a word embedding model without the use of evaluation datasets. However, two aspects of this method need further consideration: the assumption that the noise signal obeys the zero mean-Gaussian distribution has not been verified in real data, and the selected dimensionality is quite different from those obtained by the other evaluation methods based on handcrafted datasets.

Our contributions are two-fold. First, we introduce an effective dimensionality selection method for word2vec based on information criteria. Moreover, our proposed approach does not require handcrafted evaluation datasets, and therefore, can be applied to any type of data not limited to English or verbal-data. This is important and necessary to be able to choose a reasonable dimensionality of the word2vec model when applying it to various fields in information retrieval. Second, from the perspective of information theory, we propose the Sequential Normalized Maximum Likelihood (SNML) criterion in a novel combination with some heuristics for the dimensionality selection problem. The application of our proposed criterion enjoys valuable theoretical guarantees from information theory as well as experimentally ensures that the selected dimensionality is able to capture regularity from the data as well as meet the preferences of models with relatively low but sufficient dimensionality. To the best of our knowledge, this study presents the first application of information criteria in the field of embedding methods as well as the first heuristic comparison of the SNML codelength. These positive results not only encourage wide use of the proposed method in other embedding methods but also suggest a promising solution to evaluate hyper-parameters of a machine learning model.

2. Materials and Methods

2.1. Related Work

2.1.1. Word Embedding

Representations of words in a vector space have been studied exhaustively in the NLP literature. Beginning with a one-hot vector (the very first representation of words), other word representation methods such as latent semantic analysis [14] and latent Dirichlet allocation [15] have been proposed to improve NLP task performance over time. Various methods that represent words as dense vectors (referred to as "word embedding"), including GloVe [16], word2vec (SG and continuous bag of words) [17], are considered as the state-of-the-art in this field. In this paper, we focus on SG, but the proposed approach can be applied to any other word embedding model.

As the SG model often uses the negative sampling technique, in this study, we work with both original SG (oSG) and Skip-gram with Negative Sampling (SGNS) to clarify the idea behind our approach. In order to apply information criteria on the SG, we summarize it and introduce our notations for both oSG and SGNS.

SG normally takes text data as input to the whole training process. This text data is then processed into pairs of word and context (w, c) in order to feed into a neural network [17]. The preprocess procedure can be applied the same way in the case of other data types. Assume a corpus of words and their contexts are obtained after preprocessing: $\mathcal{D} = (\mathbf{w}, \mathbf{c}) = (w_1, c_1), (w_2, c_2), \ldots, (w_n, c_n)$; $w_i \in V_W, c_i \in V_C$, which are one-hot vectors, where V_W and V_C are the word and context vocabularies of sizes S_W and S_C, respectively. The training process of oSG attempts to learn the contextual distribution for each word by maximizing the likelihood function seen below.

$$\begin{aligned} P_{oSG}(\mathbf{c}|\mathbf{w}; E, F) &= \prod_{i=1}^{n} P_{oSG}(c_i|w_i; E, F) \\ &= \prod_{i=1}^{n} \frac{\exp(w_i^T EF)c_i}{\sum_{c' \in \mathbb{C}} \exp(w_i^T EF)c'}, \end{aligned} \tag{1}$$

where E and F are the parameter matrices of the shapes $(S_W \times d)$ and $(d \times S_C)$, respectively. d is the dimensionality of the embedding vector space.

Unlike oSG, SGNS learns the probability that a particular context occurred around a word or not: $P(x_{i0} = 1|w_i, c_i; E, F)$. Furthermore, SGNS introduces negative sampling by sample S_z context words: $z_i = \{z_{i1}, z_{i2}, \ldots, z_{iS_z}\} \in V_C^{(S_z)}$ for each particular word w_i: $P(x_{ij} = 0|w_i, z_{ij}; E, F); j = \{1, 2, \ldots, S_z\}$. The training process of SGNS attempts to maximize the following likelihood function:

$$\begin{aligned} P_{SGNS}(\mathbf{x}|\mathbf{w}, \mathbf{c}, \mathbf{z}; E, F) &= \prod_{i=1}^{n} P_{SGNS}(x_i|w_i, c_i, z_i; E, F) \\ &= \prod_{i=1}^{n} \sigma({w_i}^T EFc_i) \prod_{j=1}^{S_Z} \sigma(-{w_i}^T EFz_{ij}), \end{aligned} \tag{2}$$

where σ denotes sigma function [18]. In the remainder of this paper, we denote $P(\mathcal{D}; \theta)$ for both $P_{oSG}(\mathbf{c}|\mathbf{w}; E, F)$ and $P_{SGNS}(\mathbf{x}|\mathbf{w}, \mathbf{c}, \mathbf{z}; E, F)$.

2.1.2. Dimensionality of SG

Unlike our approach, Yin and Shen [11] considered word embedding to be an implicit matrix factorization problem [19] and approached the issue by deciding the rank of the component matrix. Their work was conducted by introducing Pairwise Inner Product (PIP) loss, a measure that evaluates the goodness of the rank of matrix factorization. The best rank is chosen to minimize a given upper bound of the PIP loss.

However, the selected number of dimensions does not agree with the optimal dimensionality performance based on the other evaluation tasks. For example, the best dimensionality of SG chosen by PIP loss is **129**, and the best 5% dimensionalities range from 67 to 218, while the best dimensionalities in the WordSim353 (WS), MTurk771 (MTurk), and Google word analogy (WA) datasets are **56**, **102**, and **220**, respectively [11]. Moreover, the matrix factorization operation conducted during the PIP loss calculation suffers from computational disadvantages and exceeds the calculation limit for huge amounts of data (e.g., Wikipedia dataset in our experiments).

2.1.3. Information Criteria

Word2vec is classified as a self-supervised machine learning model. Therefore, the number of dimensions can be selected by comparing the value of the loss function on the validation dataset. An alternative approach to dimensionality selection involves using information criteria such as the Akaike Information Criterion (AIC) [20], Bayesian Information Criterion (BIC) [21], and Minimum Description Length (MDL) [22]. Compared to the cross-validation method, these information criteria do not require a hold-out validation dataset, which prevents wastage of our precious data.

Since AIC, BIC, and MDL have different backgrounds with regard to the estimation of expected log-likelihood and approximation of the log marginal likelihood, we need to carefully choose the criteria to be used in specific cases. In fact, AIC and BIC rely heavily on the asymptotic theory, which states that as the data size grows to infinity, the estimated parameters converge in probability to the true values of the parameters. However, the asymptotic theory does not apply to word2vec, i.e., as the number of data increases to infinity, we can obtain different optimal parameters set (E, F). Therefore, AIC and BIC are not guaranteed to work theoretically. Nonetheless, several empirical studies have applied them successfully.

Unlike AIC and BIC, MDL with Normalized Maximum Likelihood (NML) codelength is an accurate model selection criterion for real-world data analysis based on limited samples. NML is also known as the best codelength in the context of the minimax optimality property [23].

However, choosing the best method for dimensionality selection is still an experimental task in word2vec. In the next section, we describe in detail the application of MDL to

the dimension selection problem and the reason for choosing this method. We then provide empirical comparisons between the methods listed in this section.

In order to apply these information criteria to the dimensionality selection problem, we introduce our notations for the AIC and BIC first.

$$AIC = 2(S_W \times d + d \times S_C) - 2\ln\left(P(\mathcal{D}; \hat{\theta}(\mathcal{D}))\right), \tag{3}$$

$$BIC = \ln(n)(S_W \times d + d \times S_C) - 2\ln\left(P(\mathcal{D}; \hat{\theta}(\mathcal{D}))\right), \tag{4}$$

where, $\hat{\theta}(\mathcal{D}) = (\hat{E}(\mathcal{D}), \hat{F}(\mathcal{D}))$ is the maximum likelihood estimation of the parameters on data $\mathcal{D}$.

2.2. Dimensionality Selection via the MDL Principle

2.2.1. Applying the MDL Principle, NML and SNML Codelengths

Word2vec was derived based on the distributional hypothesis of Harris [24], which states that words in similar contexts have similar meanings. Therefore, assuming the existence of the true context distribution for given words $P^*(\cdot|w)$, it is reasonable to choose the dimensionality that has the ability to learn the context distribution most similar to the true distribution. The MDL principle [22] is a powerful solution for model selection, and is considered for the dimensionality selection as per our interest.

The MDL principle states that the best hypothesis (i.e., a model and its parameters) for a given set of data is the one that leads to the best compression of the data, namely the minimum codelength [22]. Specifically, we consider each dimensionality corresponding to a probability model class $\mathcal{M}_d$.

$$\mathcal{M}_d = \{P(\mathcal{D}; \theta) : \theta = (E \in R^{(S_W \times d)}, F \in R^{(d \times S_C)})\}, \tag{5}$$

Assuming that we are able to encode a data series $\mathcal{D}$ by a series of only 0 and 1, the length of this binary series is called the codelength of data series $\mathcal{D}$. We take the expression $\mathcal{L}(\mathcal{D}; \mathcal{M}_d)$ as the codelength of data $\mathcal{D}$ that can be obtained when encoding with the given information about the model class $\mathcal{M}_d$. The MDL principle states that the closer the model class $\mathcal{M}_d$ is to the true distribution generated data $P^*(\cdot|w)$, the shorter the codelength $\mathcal{L}(\mathcal{D}; \mathcal{M}_d)$ that can be obtained.

Given a model class, there are many methods to estimate the shortest codelength of a given dataset such as: two-part codelength, Bayesian codelength [25], NML or SNML codelength. Therein, the NML codelength is the best-known codelength in the MDL literature to achieve the minimax regret [23]. The formula for the NML codelength is given below.

$$\mathcal{L}_{NML}(\mathcal{D}; \mathcal{M}_d) = -\log P(\mathcal{D}; \hat{\theta}(\mathcal{D})) + \log C(\mathcal{M}_d), \tag{6}$$

where $\log C(\mathcal{M}_d) = \log \sum_{\mathcal{D} \in \mathbb{D}^{(n)}} P(\mathcal{D}; \hat{\theta}(\mathcal{D}))$ is known as Parametric Complexity (PC); $\mathbb{D}^{(n)}$ denotes all possible data series with the length of n.

However, the PC term involves extensive computations and is not realistic to implement. Instead, we apply the SNML codelength [26] in this study to reduce the computation cost using the formula seen below:

$$\mathcal{L}_{SNML}(\mathcal{D}; \mathcal{M}_d) = \sum_{i=1}^{n} \mathcal{L}_{SNML}(\mathcal{D}_i | \mathcal{D}^{i-1}; \mathcal{M}_d), \tag{7}$$

where $\mathcal{D}^i$ denotes data series $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_i$ and $\mathcal{D} = \mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n$. The SNML codelength is calculated as the total codelength where the codelength for each datum is sequentially calculated such as the NML codelength every time it is input. It is known that the SNML codelength is a good approximation to the NML codelength [27]. Since the SNML codelength is sequentially calculated, its computational cost at each step is much lower than that of the NML codelength. Based on the assumption of independence between the data records, the training process of the word2vec model comes after data records have

been shuffled. Under this independence assumption, the independent process of SNML does not depend on the order of data.

In addition, the SNML codelength function $\mathcal{L}_{SNML}(\mathcal{D}_i|\mathcal{D}^{i-1};\mathcal{M}_d)$ can be applied to oGS and SGNS in the forms seen below.

$$\begin{aligned}\mathcal{L}_{SNML}(\mathcal{D}_i|\mathcal{D}^{i-1};\mathcal{M}_d^{oSG}) = &-\log P_{oSG}(c_i|w^i,c^{i-1};\hat{\theta}(w^i,c^i)) \\ &+\log\sum_{c\in V_C} P_{oSG}(c|w^i,c^{i-1};\hat{\theta}(w^i,c^{i-1},c)),\end{aligned} \tag{8}$$

$$\begin{aligned}\mathcal{L}_{SNML}(\mathcal{D}_i|\mathcal{D}^{i-1};\mathcal{M}_d^{SGNS}) = &-\log P_{SGNS}(x_i|w^i,c^i,z^i,x^{i-1};\hat{\theta}(w^i,c^i,z^i,x^i))+ \\ &\log\sum_{x\in\mathcal{O}^{(S_z)}} P_{SGNS}(x|w^i,c^i,z^i,x^{i-1};\hat{\theta}(w^i,c^i,z^i,x^{i-1},x)),\end{aligned} \tag{9}$$

where $\mathcal{O}^{(S_z)}$ is a set of all possible one-hot vectors of S_z dimensions.

2.2.2. Some Heuristics Associated with SNML Codelength Calculation

The computation of the SNML codelength still costs nS_C times to execute the maximum likelihood estimation for each data record $\mathcal{D}_i$, which is also not realistic. We introduce two techniques for saving the computational costs for SNML: heuristic comparison and importance sampling on the SNML codelength.

Heuristic comparison

A simple observation reveals that if the codelength of data obtained with model class $\mathcal{M}_d$ is the shortest, then only some part of the data can also be achieved with the shortest codelength compressed with the same model class. Therefore, instead of computing the codelength for all n records of data, we can use the codelength of a small set of data. In fact, the results of our experiments show that focusing on the last several thousand records of data is sufficient to compare model classes.

Figure 1 demonstrates the differences in SNML codelengths of different dimensionalities compared with the dimensionality that achieves the shortest codelength on the data. The vertical axis shows the difference of data codelengths obtained by two different dimensionalities shown in the legends (e.g., $d1$ *vs*. $d2$ *dim*); specifically, it is calculated by $\mathcal{L}(\mathcal{D}';d1) - \mathcal{L}(\mathcal{D}';d2)$ where $\mathcal{L}$ is the codelength function; $\mathcal{D}'$ is data; $d1$ and $d2$ are dimensionality; while the value of horizontal axis shows the number of records of $\mathcal{D}'$.

To facilitate comparisons among dimensionalities that are markedly different from one another (such as 30 dimensions versus 65 dimensions in Figure 1(1), or 200 dimensions versus 130 dimensions in Figure 1(2)), it is sufficient to use only 6000 data records to provide information about the best dimensionality to be chosen. Therefore, adding data thereafter simply increases the SNML codelength but does not change our answer substantially. However, for similar dimensionalities, such as 60, 65, and 70 dimensions in Figure 1(3), the first one million data records cannot help us identify the optimal dimensionality. This phenomenon leads to confusion when the codelengths between two model classes are not too different. Furthermore, the number of data records required to determine the best dimensionality comes from the nature of the dataset and the tasks themselves. For example, in the case of word2vec, when SGNS randomizes only a few samples for the negative label from a large context set, the codelength of each data record will vary more than the codelength in oSG, which does not randomly sample negative samples.Therefore, SGNS needs more data records to determine the difference between candidate dimensionality.

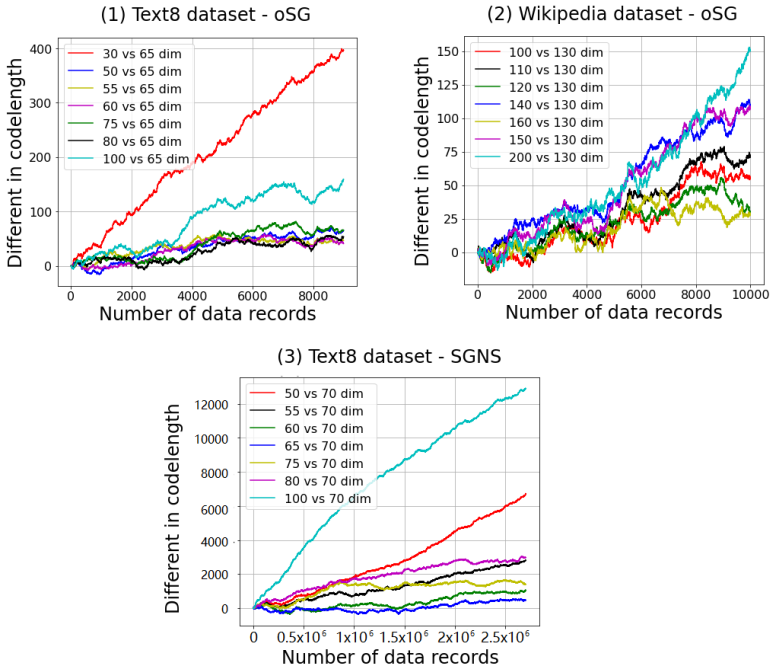

Figure 1. Cumulative SNML codelengths of different dimensionalities compared to the dimensionality result with the shortest codelength.

To ensure that the correct model is chosen, we need to increase the number of records to estimate the SNML codelength so as to allow a better comparison of these two dimensionalities. However, a small dimension error in the dimensionality selection of word2vec does not affect the final performance considerably. Therefore, the trade-off between the computing time and model selection accuracy is determined by the number of records beyond those required to estimate the SNML codelength with sufficient finality.

Importance sampling

Since the size of the context set S_C is large (approximately 30,000–100,000 or above, according to the training dataset), the computation of PC for SNML in oSG is still very expensive. We apply the importance sampling method to approximately estimate the SNML description length for each data record. In detail, if a distribution Q on the context set satisfies $Q(c) \neq 0 \; \forall \; c \in V_C$, the following formula can be applied.

Let $f(c) = P\left(c \middle| w^i, c^{i-1}; \hat{\theta}(w^i, c^{i-1}, c)\right)$, then

$$\sum_{j=1}^{S_C} f(c_j) = \mathbb{E}_Q\left(\frac{f(c)}{Q(c)}\right) \approx \frac{1}{m} \sum_{c \in \mathcal{S}} \frac{f(c)}{Q(c)}, \tag{10}$$

where $\mathcal{S} = \{c_1, c_2, .., c_m\} \sim Q(c)$: set of samples draw from distribution Q.

This estimation asymptotes to the true value as m (the number of samples) increases, and distribution Q is similar to function $f(c)$. In our experiment, the uniform distribution is the best choice for distribution Q, and the sampling size is chosen to be 1/10 the size of the context set to balance the computation time and sampling error.

3. Results

3.1. Experimental Settings

3.1.1. Data

We compared the above-mentioned model selection criteria using SG trained on three datasets: synthetic data, text8, and Wikipedia.

Synthetic data

Synthetic data were generated based on several random questions from the WA dataset. Assuming a numeric context set, we generated categorical distributions on this set for all words for which the parameter vectors of the corresponding distributions satisfy the constraints in the questions. For example, corresponding to question: *Tokyo*, *Japan*, *Paris*, *France*, the process involves the generation of four random contextual distributions, $\widetilde{P}(\cdot|Tokyo)$, $\widetilde{P}(\cdot|Japan)$, $\widetilde{P}(\cdot|Paris)$, $\widetilde{P}(\cdot|France)$, such that:

$$cosine(\widetilde{P}(\cdot|Tokyo), \widetilde{P}(\cdot|Japan)) = cosine(\widetilde{P}(\cdot|Paris), \widetilde{P}(\cdot|France)), \tag{11}$$

The implementation for generation of such categorical distributions is also available on GitHub (https://github.com/truythu169/snml-skip-gram).

We then sampled words using a uniform distribution and contexts using $\widetilde{P}$ adding normal distribution noises. Using these pairs of word and context, oSG and SGNS can be trained to achieve a 100% score on the questions used to create data with the appropriate dimensionality. Furthermore, good dimensionality should result in contextual distributions similar to $\widetilde{P}$. To evaluate this similarity, we used a dissimilar function for the oSG model and a similar function for the SGNS model as follows:

$$dissimilar(\mathcal{M}_d^{(oSG)}, \widetilde{P}) = \frac{1}{S_W} \sum_{w \in V_W} D_{KL}(P_{oSG}(\cdot|w; \hat{\theta})||\widetilde{P}(\cdot|w)), \tag{12}$$

$$similar(\mathcal{M}_d^{(SGNS)}, \widetilde{P}) = \frac{1}{S_W} \sum_{w \in V_W} \rho(f_{P_{SGNS}}(\cdot|w; \hat{\theta}), f_{\widetilde{P}}(\cdot|w)), \tag{13}$$

where, D_{KL} denotes for Kullback–Leibler divergence, ρ denotes Spearman's rank correlation coefficient, $f_{P_{SGNS}}(\cdot|w; \hat{\theta})$ and $f_{\widetilde{P}}(\cdot|w)$ are vectors that take $P_{SGNS}(x = 1|w, c; \hat{\theta})$ and $\widetilde{P}(c|w)$ ($c \in V_C$) as elements, respectively. The choice of D_{KL} for oSG comes from the fact that oSG outputs a categorical distribution, which can be compared with the true distribution using D_{KL}; while SGNS results in a list of probability values that are expected to have a strong positive correlation with values of $\widetilde{P}$. We used $dissimilar(\mathcal{M}_d^{(oSG)}, \widetilde{P})$ and $similar(\mathcal{M}_d^{(SGNS)}, \widetilde{P})$ as the oracle criterion to evaluate the optimal dimensionality for synthetic data. A good dimensionality selection method is expected to select a dimensionality that is nearby the one chosen by the oracle criterion.

Text datasets

The text8 and Wikipedia datasets were preprocessed using a window size of 5, removing words that occur less than 73 times and applying subsampling with a threshold of 10^{-5}. In addition, we only used the first 20,000 articles of the English Wikipedia dump for the training process.

3.1.2. Training Process

Optimization settings

In order to speed up the training process, we implemented a momentum optimizer and mini-batch with a batch size of 1000 for oSG training and stochastic gradient descent for SGNS, as in [28]. A learning rate α for oSG was set to 1.0, and momentum was set to 0.9. For SGNS, α was chosen to be 0.1, and the number of negative samples, 15. The number of epochs was chosen so that the negative log-likelihood value is not significantly reduced. For instance, in the case of oSG, 200 and 90 epochs, respectively, were selected for text8 and Wikipedia, while for SGNS, 15 epochs were selected for text8. Practically, these optimization settings achieve the best performance in our experiment. For example, the best word analogy (WA) scores for text8 are 32.6% (SGNS) and 38.6% (oGS), the corresponding value for Wikipedia is 50.5% (oNS).

Because of the limitations posed by the computational resources, we experimented on a finite number of dimensionalities, which we think is sufficient to clarify the idea behind

this research. The evaluated dimensionalities are shown in the figures corresponding to each dataset.

Importance sampling

In our experiment, the uniform distribution is the best choice for the distribution Q to approximate the SNML codelength. In Table 1, we show the average error of the codelength per record of data according to the sampling size using importance sampling. The implementation is tested on the text8 dataset. Finally, we chose the sampling size to be 1/10 as the size of the context set (the context set comprises about 30,000 words) to balance the computation time and sampling error.

Table 1. Average error of importance sampling.

Sampling Size	6000	3000	1500	600	300
Average Error	0.0022	0.0045	0.009	0.02	0.042

Estimation of SNML codelength

The estimation of SNML codelength required us to repeat the parameter estimation $\hat{\theta}(w^i, c^i)$ $s \times m$ times, where s is the number of records beyond those required to estimate the SNML codelength, and m is the sampling size. However, repeatedly estimating parameters from scratch is very time consuming. We can alternatively estimate $\hat{\theta}(w^i, c^i)$ from $\hat{\theta}(w^{i-1}, c^{i-1})$ by taking the gradient descent of (w_i, c_i).

3.2. Experimental Results

3.2.1. Synthetic Data

We compared five criteria: AIC, BIC, SNML, accuracy on the WA task, and loss value on the validation dataset (CV) with the oracle criterion. The experimental results are shown in Figures 2 and 3. Due to the differences between the criteria values, we scaled all the values to range from 0 to 1 for visual purposes. Moreover, while the dissimilar oracle and other criteria take the dimensionality that minimizes the value, WA takes the maximum. Therefore, in the figure, we draw the line showing the negative value plus one for the dissimilar oracle, AIC, BIC, SNML, and CV so that the higher value states better indicate the dimensionality to be chosen. This scale procedure was also applied to Figures 4–9. The horizontal axis in these figures shows the number of dimensions.

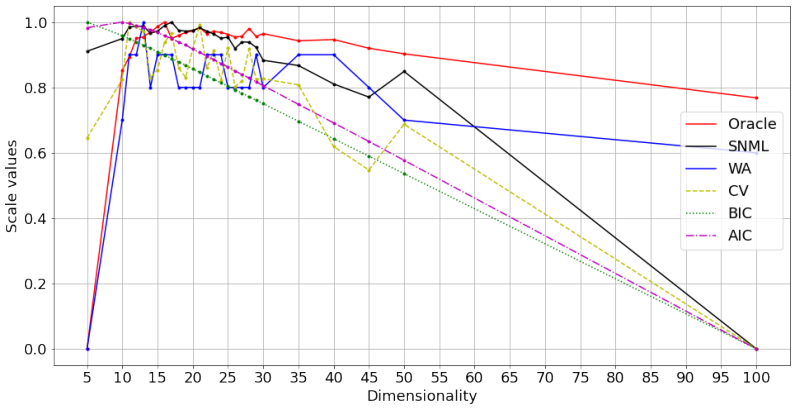

Figure 2. Normalized values of criteria compared with the oracle on artificial data: training with oSG.

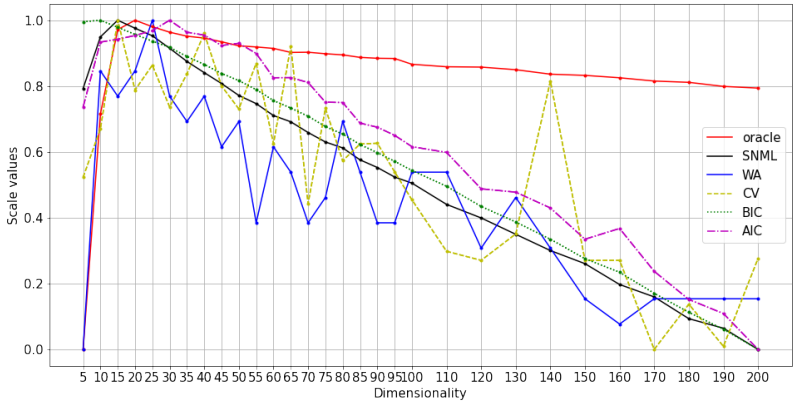

Figure 3. Normalized values of criteria compared with the oracle on artificial data: training with SGNS.

The results for oSG show that the BIC exhibits a monotonous decrease, while the optimal dimensionality chosen by the oracle and SNML is 16 and 17, respectively. On the one hand, AIC and CV choose a more distinct dimensionality: 10 and 13, respectively. For SGNS, the oracle chooses 20 dimensions, SNML and CV choose 15, and WA achieves the highest score at 25 dimensions. On the other hand, the BIC chooses 10 dimensions, while the AIC chooses 30.

In both oSG and SGNS, SNML chooses the dimensionality closest to the oracle criterion. Thus, SNML outperforms both AIC and BIC. Note that the synthetic data are designed to achieve a 100% WA score using contextual distribution; however, the scores achieved by using embedded vectors are sensitive to noises and change significantly according to the dimensionality.

3.2.2. Text Data

We compared the SNML criterion with the NLP word analogy task (using WA) and word similarity tasks (using WS, MTurk, and MEN-3k test collection (MEN)). As knowledge regarding the underlying true distribution of the data is lacking, it is difficult to determine the best dimensionality selection method. However, assuming the existence of the true contextual distribution, NLP tasks will roughly prioritize models closest to the true distribution. Therefore, the dimensionality selected by a good method is expected to be close to the optimal dimensionalities for NLP tasks. Note that the evaluation method using scores of NLP tasks is available only in the case of English text data; therefore, it is reasonable to apply a method that achieves the same results as the NLP tasks-based method to any other type of data.

We experimented with at least three runs for each dataset, and the average results are shown in Figures 4–6. The comparison of SNML with the information criteria, CV, and PIP is depicted in Figures 7–9.

The main results of the study are shown in Table 2, and the actual values of experiments are summarized in the Appendix A. The optimal dimensionalities chosen by the proposed method were compared with the optimal dimensionality in word analogy and word similarity tasks in NLP [13]. Accordingly, for oSG, SNML and CV chose the same dimensionality, which is closer to the optimal dimensionality in NLP tasks than AIC, BIC and PIP. For SGNS, SNML chose the dimensionality closer to the optimal dimensionality for three (WS, WA, MEN) in four tasks (WA, WS, MEN and MTurk) implemented when compared to CV; and four in four tasks implemented when compared to BIC and PIP. We conclude that SNML is better than CV, AIC, BIC and PIP in almost all implemented NLP tasks. Note that we are unable to implement PIP on Wikipedia because the computational complexity was beyond the capabilities of our servers. We are also unable to find the minimum values of BIC and AIC (for text8 train with oSG) for dimensions over a long range.

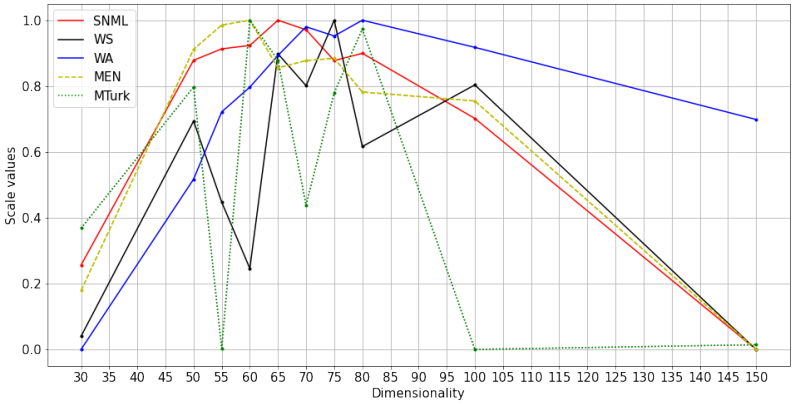

Figure 4. Normalized values and scores on NLP tasks with SNML: text8 training with oSG.

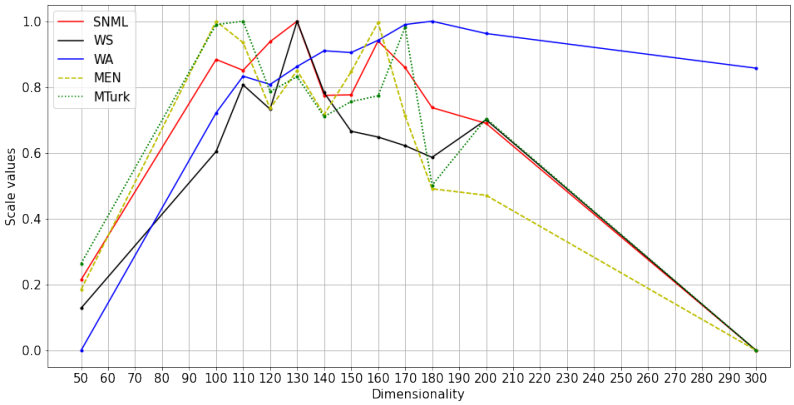

Figure 5. Normalized values and scores on NLP tasks with SNML: Wikipedia training with oSG.

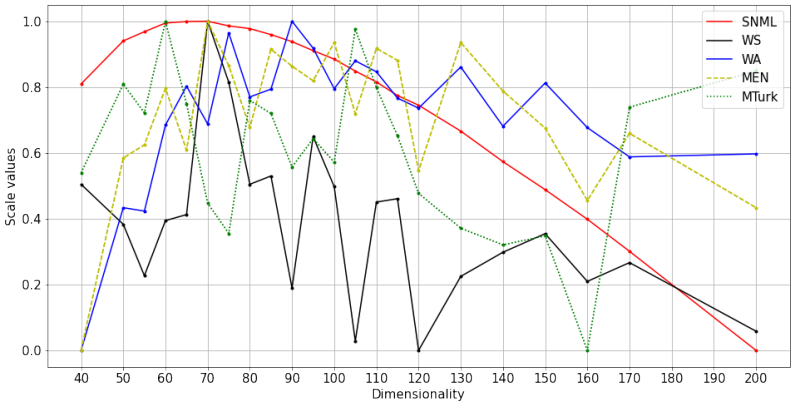

Figure 6. Normalized values and scores on NLP tasks with SNML: text8 training with SGNS.

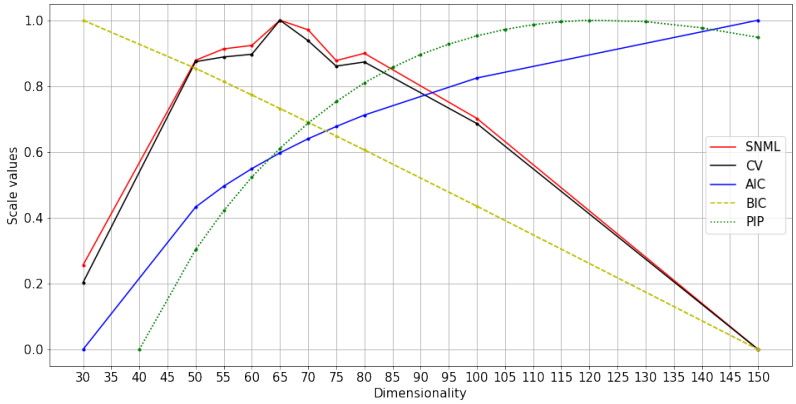

Figure 7. Normalized values of information criteria, CV, and PIP: text8 training with oSG.

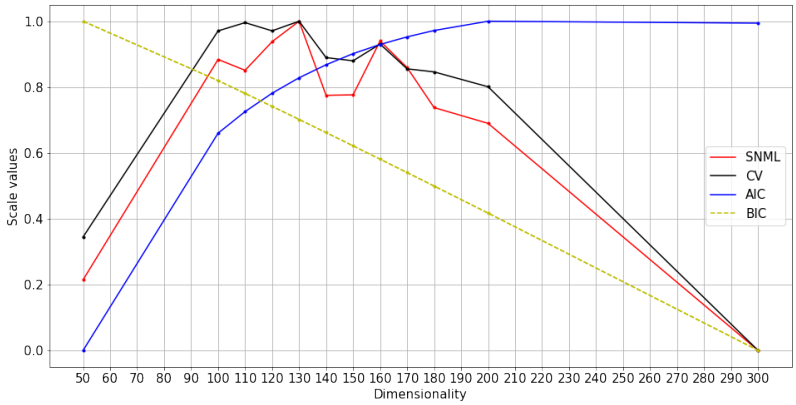

Figure 8. Normalized values of information criteria, CV, and PIP: Wikipedia training with oSG.

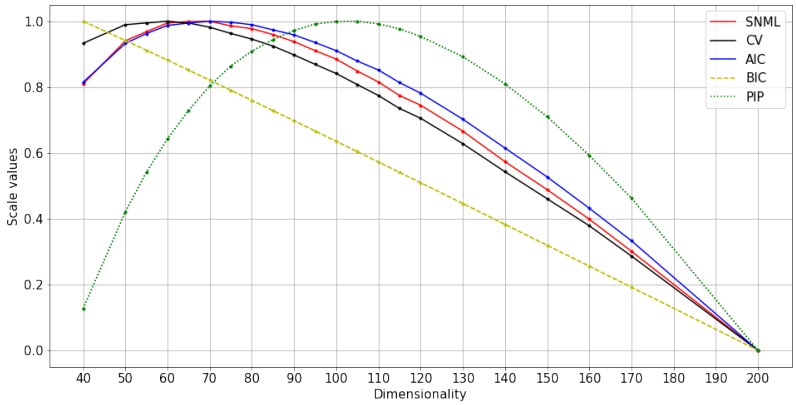

Figure 9. Normalized values of information criteria, CV, and PIP: text8 training with SGNS.

Table 2. Optimal dimensionalities chosen by different criteria (-: unknown).

	SNML	**WS**	**WA**	**MEN**	**MTurk**	**CV**	**AIC**	**PIP**
Text8 (oSG)	65	75	80	60	60	65	-	120
Wikipedia (oSG)	130	130	180	100	110	130	200	-
Text8 (SGNS)	70	70	95	70	60	60	70	105

Furthermore, the SNML criterion tends to favor smaller dimensions, although it is sufficient to ensure good performance on other NLP tasks while heavily penalizing model classes that tend to overfitting. This characteristic of SNML helps us avoid choosing large models, and therefore, the available resources should be fully utilized. On the other hand,

although AIC favors bigger dimensions and the performance of the model is slightly reduced, this approach is computationally advantageous over SNML. This advantage makes AIC useful in some situations.

4. Discussion

To the best of our knowledge, this is the first study that applies information criteria to dimensionality selection for embedding methods.

In order to demonstrate the basic property of our method, we applied it to the very basic model of this field (i.e., Skip-gram model). Our proposed framework can be applied to other embedding methods as well as other neural network-based models once a likelihood function corresponding to any embedding method is defined. For example, in the case of BERT [29], the likelihood function can be defined using a joint probability distribution of a masked token and the next sentence. This likelihood function is then substituted for distribution P in Equations (7)–(9) to obtain SNML codelengths.

The optimal dimensionality selected by SNML is low; however, it is sufficient to ensure good performance in terms of significant closeness of optimal dimensionality in NLP tasks. However, deep learning models (DL) usually benefit from over-parameterization properties, i.e, the performance of models is not significantly reduced due to the increasing number of parameters. Furthermore, there exist other approaches to the overfitting problem, such as regularization, early-stopping, randomly drop-out, etc., or strategies to adopt large DL models to small data, such as transfer learning, semi-supervised learning, etc. In the paper, we introduce an alternative method to the same problem from an information-theoretic view. The optimal dimensionality selected by the proposed framework can benefit from the over-parameterization property of DL by adopting alternatives such as applying other codelength methods or considering other parameterization methods for the likelihood function. Future challenges in this field include determining which modification results in the most improvement for the dimensionality selection strategy.

5. Conclusions

When considering word2vec SG as a probability distribution estimation problem, the optimal dimensionality can provide an estimation of contextual distribution as close as possible to the true distribution-generated data. We tested information criteria (AIC and BIC) and SNML with some heuristics to select such a dimensionality. The experimental results on synthetic data showed that the SNML could choose a dimensionality such that the corresponding probability model is able to learn the contextual distribution closest to the true distribution-generated data. The experiments on text datasets showed that SNML has the ability to choose a desirable dimensionality with regard to two aspects, although low dimensionality is sufficient to ensure good performance in terms of significant closeness of optimal dimensionality in NLP tasks without a hold-out test dataset. Furthermore, SNML typically outperforms AIC, BIC, CV, and PIP in the selection of good dimensionality for NLP tasks in our experiments. Our method therefore holds promise for choosing the most appropriate dimensionality in word2vec when training with data not limited to English or non-verbal.

To the best of our knowledge, this is the first study that applies information criteria to dimensionality selection for word embedding. In fact, the limitations associated with computation or asymptotic estimation of NML or SNML codelength make it difficult to apply such criteria in these areas. By introducing some heuristics in the SNML codelength calculation, we have discovered a new and useful approach, namely MDL-based knowledge embedding. Our proposed approach can be applied to other embedding methods once a likelihood function corresponding to any embedding method is defined. A more detailed evaluation will be left for future study.

Author Contributions: Conceptualization, P.T.H. and K.Y.; methodology, P.T.H. and K.Y.; software, P.T.H.; validation, P.T.H. and K.Y.; investigation, P.T.H and K.Y.; resources, P.T.H and K.Y.; data curation, P.T.H; writing—original draft preparation, P.T.H.; writing—review and editing, K.Y.; visualization, P.T.H.; supervision, K.Y. Both authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by JST KAKENHI 191400000190 and JST-AIP JPMJCR19U4.

Data Availability Statement: (text8) Matt Mahoney, 2011, Large Text Compression Benchmark, http://mattmahoney.net/dc/textdata (accessed on 14 June 2021); (Wikipedia) Wikimedia Foundation, 2019, Wikipedia Database backup dumps, https://dumps.wikimedia.org/ (accessed on 14 June 2021)

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Experiments Result in Actual Values

Pre-scaled results of experiments are provided in the following Tables A1–A4. The results show mean values of different trials.

Table A1. Result of oSG on synthetic data.

Dim	WA	AIC	BIC	CV	SNML	Oracle
5	0	0.983	1	0.645	0.911	0
10	0.7	1	0.959	0.824	0.949	0.852
11	0.9	0.996	0.949	1	0.985	0.893
12	0.9	0.99	0.939	0.986	0.987	0.95
13	1	0.984	0.929	0.97	0.988	0.955
14	0.8	0.977	0.92	0.83	0.966	0.969
15	0.9	0.968	0.909	0.853	0.972	0.986
16	0.9	0.958	0.899	0.938	0.989	1
17	0.9	0.95	0.889	0.966	1	0.949
18	0.8	0.938	0.878	0.86	0.975	0.96
19	0.8	0.93	0.868	0.83	0.972	0.968
20	0.8	0.917	0.857	0.918	0.974	0.973
21	0.8	0.907	0.846	0.991	0.983	0.983
22	0.9	0.896	0.836	0.859	0.972	0.965
23	0.9	0.885	0.825	0.913	0.964	0.971
24	0.9	0.873	0.814	0.823	0.95	0.968
25	0.8	0.863	0.804	0.921	0.954	0.962
26	0.8	0.851	0.793	0.803	0.919	0.954
27	0.8	0.84	0.782	0.82	0.939	0.957
28	0.8	0.829	0.772	0.918	0.939	0.98
29	0.9	0.817	0.761	0.826	0.923	0.956
30	0.8	0.806	0.75	0.827	0.883	0.965
35	0.9	0.748	0.697	0.808	0.867	0.943
40	0.9	0.691	0.643	0.619	0.811	0.947
45	0.8	0.635	0.59	0.548	0.77	0.92
50	0.7	0.577	0.536	0.687	0.849	0.903
100	0.6	0	0	0	0	0.768

Table A2. Result of SGNS on text8 dataset.

Dim	Mean SNML	WS	WA	MEN	MTurk	Mean CV
40	2.2546	67.84	0.26	68.5	54.34	3.2143
50	2.2493	67.51	0.29	70.05	55.46	3.2101
55	2.2481	67.08	0.29	70.15	55.1	3.2097
60	2.247	67.54	0.31	70.6	56.25	3.2093
65	2.2469	67.59	0.31	70.11	55.21	3.2097
70	2.2468	69.18	0.31	71.14	53.96	3.2106
75	2.2474	68.68	0.32	70.79	53.58	3.212
80	2.2477	67.84	0.31	70.29	55.25	3.2133
85	2.2485	67.9	0.31	70.92	55.09	3.2149
90	2.2494	66.98	0.33	70.78	54.41	3.2169
95	2.2505	68.23	0.32	70.67	54.77	3.219
100	2.2515	67.82	0.31	70.97	54.48	3.221
105	2.253	66.54	0.32	70.4	56.15	3.2235
110	2.2544	67.69	0.32	70.92	55.41	3.226
115	2.256	67.72	0.31	70.83	54.81	3.2289
120	2.2572	66.47	0.31	69.94	54.08	3.2311
130	2.2604	67.08	0.32	70.97	53.65	3.2368
140	2.2642	67.28	0.3	70.58	53.43	3.2431
150	2.2677	67.43	0.31	70.29	53.55	3.2492
160	2.2714	67.03	0.3	69.7	52.11	3.2553
170	2.2753	67.19	0.3	70.24	55.17	3.2621
200	2.2876	66.62	0.3	69.65	55.63	3.2833

Table A3. Result of oSG on text8 dataset.

Dim	SNML	WS	WA	MEN	MTurk	CV
30	78,038.99	65.75	0.32	67.96	53.82	86,695.17
50	77,707.09	67.39	0.35	70.17	54.95	86,323.13
55	77,688.63	66.78	0.36	70.39	52.86	86,315.21
60	77,683.08	66.27	0.37	70.43	55.48	86,310.65
65	77,642.21	67.91	0.37	70	55.16	86,253.29
70	77,657.7	67.66	0.38	70.06	54	86,287.76
75	77,707.66	68.16	0.38	70.08	54.9	86,330.63
80	77,695.85	67.2	0.38	69.78	55.41	86,323.71
100	77,801.35	67.67	0.37	69.69	52.85	86,427.45
150	78,175.82	65.65	0.36	67.42	52.89	86,808.16

Table A4. Result of oSG on Wikipedia dataset.

	SNML	WS	WA	MEN	MTurk	CV
50	90,809.62	62.88	0.45	70.47	59.08	271,750.38
100	90,483.56	62.67	0.49	71.17	60.55	270,826.27
110	90,499.69	63.56	0.49	71.22	60.37	270,789.28
120	90,457.3	62.75	0.49	70.76	59.92	270,826.06
130	90,427.01	64.58	0.5	71.13	60.32	270,782.92
140	90,536.69	63.69	0.49	70.54	60.65	270,946.03
150	90,535.94	63.14	0.5	70.91	60.41	270,960.41
160	90,456.31	62.39	0.5	71.36	59.3	270,885.36
170	90,495.59	62.46	0.5	70.85	61.07	270,996.38
180	90,554.84	62.33	0.5	70.79	59.63	271,010.14
200	90,578.18	63.83	0.5	70.39	60.65	271,076.85
300	90,914.16	62.94	0.49	69.36	57.95	272,259.16

References

1. Luong, T.; Pham, H.; Manning, C.D. Effective Approaches to Attention-based Neural Machine Translation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; Association for Computational Linguistics: Stroudsburg, PA, USA, pp. 1412–1421. doi:10.18653/v1/D15-1166.
2. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.U.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA 2017; pp. 5998–6008.
3. Lilleberg, J.; Zhu, Y.; Zhang, Y. Support vector machines and Word2vec for text classification with semantic features. In *ICCI*CC*; Ge, N., Lu, J., Wang, Y., Howard, N., Chen, P., Tao, X., Zhang, B., Zadeh, L.A., Eds.; IEEE Computer Society: Los Alamitos, CA, USA, 2015; pp. 136–140.
4. Nallapati, R.; Zhou, B.; dos Santos, C.; GuÌ‡lçehre, Ç.; Xiang, B. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, Berlin, Germany, 11–12 August 2016.
5. Sienčnik, S.K. Adapting word2vec to Named Entity Recognition. In Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA 2015, Vilnius, Lithuania, 11–13 May 2015; Linköping University Electronic Press: Linköpings, Sweden, 2015; number 109, pp. 239–243.
6. Tshitoyan, V.; Dagdelen, J.; Weston, L.; Dunn, A.; Rong, Z.; Kononova, O.; Persson, K.A.; Ceder, G.; Jain, A. Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature* **2019**, *571*, 95–98. doi:10.1038/s41586-019-1335-8.
7. Gligorijevic, D.; Stojanovic, J.; Djuric, N.; Radosavljevic, V.; Grbovic, M.; Kulathinal, R.J.; Obradovica, Z. Large-Scale Discovery of Disease-Disease and Disease-Gene Associations. *Sci. Rep.* **2016**, *6*, 32404. doi:10.1038/srep32404.
8. Barkan, O.; Koenigstein, N. Item2Vec: Neural Item Embedding for Collaborative Filtering. *arXiv* **2016**, arXiv:1603.04259.
9. Grbovic, M.; Cheng, H. Real-Time Personalization Using Embeddings for Search Ranking at Airbnb. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018.
10. Wang, J.; Huang, P.; Zhao, H.; Zhang, Z.; Zhao, B.; Lee, D.L. Billion-Scale Commodity Embedding for E-Commerce Recommendation in Alibaba. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018.
11. Yin, Z.; Shen, Y. On the Dimensionality of Word Embedding. In *Advances in Neural Information Processing Systems 31*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2018; pp. 895–906.
12. Shu, R.; Nakayama, H. Compressing Word Embeddings via Deep Compositional Code Learning. In Proceedings of the International Conference on Learning Representations,Vancouver, BC, Canada, 30 April–3 May 2018.
13. Zhai, M.; Tan, J.; Choi, J.D. Intrinsic and Extrinsic Evaluations of Word Embeddings. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16), Phoenix, AZ, USA, 12–17 February 2016.
14. Deerwester, S.C.; Dumais, S.T.; Landauer, T.K.; Furnas, G.W.; Harshman, R.A. Indexing by Latent Semantic Analysis. *JASIS* **1990**, *41*, 391–407.
15. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
16. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the EMNLP 2014: Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, 25–29 October 2014.
17. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
18. Wikipedia contributors. Sigmoid Function. 2021. Available online: https://en.wikipedia.org/wiki/Sigmoid_function (accessed on 1 July 2021).
19. Levy, O.; Goldberg, Y. Neural Word Embedding as Implicit Matrix Factorization. In *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems—Volume 2*; MIT Press: Cambridge, MA, USA, 2014; pp. 2177–2185.
20. Akaike, H., Information Theory and an Extension of the Maximum Likelihood Principle. In *Selected Papers of Hirotugu Akaike*; Springer: New York, NY, 1973; pp. 199–213.
21. Schwarz, G. Estimating the Dimension of a Model. *Ann. Stat.* **1978**, *6*, 461–464. doi:10.1214/aos/1176344136.
22. Rissanen, J. Modeling by Shortest Data Description. *Automatica* **1978**, *14*, 465–471. doi:10.1016/0005-1098(78)90005-5.
23. Shtarkov, Y.M. Universal Sequential Coding of Single Messages. *Probl. Inf. Transm.* **1987**, *23*, 3–17.
24. Harris, Z.S. Distributional structure. *Word* **1954**, *10*, 146–162.
25. Park, I.M.; Pillow, J.W. Bayesian Efficient Coding. *Prepr. Serv. Biol.* **2017**, doi:10.1101/178418.
26. Rissanen, J.; Roos, T.; Myllymäki, P. Model selection by sequentially normalized least squares. *J. Multivar. Anal.* **2010**, *101*, 839–849. doi:10.1016/j.jmva.2009.12.009.
27. Rissanen, J. *Optimal Estimation of Parameters*; Cambridge University Press: Cambridge, UK, 2012.
28. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*; Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q., Eds.; Curran Associates, Inc.: New York, NY, USA, 2013; pp. 3111–3119.

29. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019), Minneapolis, MN, USA, 2–7 June 2019.