DAWE: A Double Attention-Based Word Embedding Model with Sememe Structure Information

Shengwen Li¹,²*, Renyao Chen¹, Bo Wan¹,²*, Junfang Gong¹, Lin Yang¹,² and Hong Yao³*

¹ School of Geography and Information Engineering, China University of Geosciences, Wuhan 430074, China; swli@cug.edu.cn (S.L.); cryao@cug.edu.cn (R.C.); wanbo@cug.edu.cn (B.W.); jfgong@cug.edu.cn (J.G.); yanglin@cug.edu.cn (L.Y.)
² National Engineering Research Center for Geographic Information System, University of Geosciences, Wuhan 430074, China
³ School of Computer Science, China University of Geosciences, Wuhan 430074, China
* Correspondence: yaohong@cug.edu.cn; Tel.: +86-27-67883716

Received: 15 July 2020; Accepted: 19 August 2020; Published: 21 August 2020

Abstract: Word embedding is an important reference for natural language processing tasks, which can generate distribution presentations of words based on many text data. Recent evidence demonstrates that introducing sememe knowledge is a promising strategy to improve the performance of word embedding. However, previous works ignored the structure information of sememe knowledges. To fill the gap, this study implicitly synthesized the structural feature of sememes into word embedding models based on an attention mechanism. Specifically, we propose a novel double attention word-based embedding (DAWE) model that encodes the characteristics of sememes into words by a “double attention” strategy. DAWE is integrated with two specific word training models through context-aware semantic matching techniques. The experimental results show that, in word similarity task and word analogy reasoning task, the performance of word embedding can be effectively improved by synthesizing the structural information of sememe knowledge. The case study also verifies the power of DAWE model in word sense disambiguation task. Furthermore, the DAWE model is a general framework for encoding sememes into words, which can be integrated into other existing word embedding models to provide more options for various natural language processing downstream tasks.

Keywords: natural language processing; word representation learning; word2vec; sememes; attention mechanism; structural information

1. Introduction

The basis of applying deep learning to solve natural language processing (NLP) tasks is to obtain high-quality representations of words from large amounts of text data [1]. Traditionally, words are represented in a sparse high-dimensional space using count-based vectors in which each word in a vocabulary is represented by a single dimension [2]. In contrast, word embedding aims to map words into continuous low-dimensional semantic space; in this way, each word is represented by a real-valued vector, namely word vector, often composed tens or hundreds of dimensions [3]. Word embedding assumes that words used in similar ways should have similar representations, thereby naturally capturing their meaning. Word vectors obtained from word embedding have been widely used in many applications: text summarization, sentiment analysis, reading comprehension, machine translation, etc.

In a great deal of word embedding-related works that have emerged in recent years, the Word2Vec [3] model strikes a good balance between efficiency and quality. In the training
process of Word2vec, words are mapped to the same vector space. For words that share a similar context in the corpus, their corresponding vectors should be close to each other in the vector space [4]. During the training process, the vector assignments of the words are repeatedly adjusted until the values are close enough to each other, if they are adjacent in the text corpus. As a result, the low-frequency words in corpus cannot be accurately represented in Word2Vec model because the training process of such words is not sufficient.

Recently, studies have shown that taking external knowledge as the complement of text corpus can effectively improve the quality of word embeddings [5–10]. Among them, the “word–sense–sememe” (Figure 1) knowledge is an intuitive form of organizing words and their senses that are easily organized and understood [11]. By synthesizing the “word–sense–sememe” knowledge, Niu et al. [7] proposed the sememe-encoded word representation learning (SE-WRL) model that made significant performance in word embedding.

![Figure 1. An example of the word–sense–sememe structure, where sememes are defined as minimum units of word meanings. There exists a limited close set of sememes to compose the open set of word meanings (i.e., word sense). For instance, the word (the first layer) “Apple” contains three senses (the second layer): “Apple Brand” (a famous computer brand), “Apple” (a sort of fruit) and “Apple Tree”. The third layer is those sememes explaining each sense. The sememes of sense “Apple Brand” are “computer”, “PatternValue”, “able”, “bring” and “SpeBrand (specific brand)”. The sememe of sense “Apple” is “fruit”. The sememes of sense “Apple Tree” are “fruit”, “reproduce” and “tree”.

SE-WRL model believes that contributions of each sememe under a sense are equivalent. However, the nature of sememes determines that different sememes under a sense may be different, which means the contributions of each sememe to the sense should be varied depending on the particular case. The inequality may be caused by two main reasons: (1) Different senses correspond to the different hierarchical structures of sememes. The sememes are organized into a hierarchical structure, such as the sememes in the sense “Apple brand” in Figure 1. Because of the hierarchical structure, there is fusion among sememes, which means that sememes at different branches of different levels are usually not equivalent. For example, the sememe “computer” in the sense “Apple brand” can be presented by its under-layer sememes (“PatternValue”, “able”, “bring” and “SpeBrand (specific brand)”). Furthermore, sememes at the same level are not equivalent in most cases. (2) The context of sememes is varied. The meaning of a word needs to be reflected in a special context, so the sememes are also affected by the context of the word. As shown in Figure 2, when the word “Apple” appears in the context “I am going to the ~ store now.”, the meaning of “Apple” should be close to the sense “Apple brand”. At this point, the sememe “SpeBrand” should have a higher weight than other sememes of the sense “Apple brand”. Therefore, the weights of sememes in a sense should change dynamically with different contexts.
Figure 2. Interpretation of weight changes of sense and sememe. Sense means the meaning that exists within the word and does not change with the context, but its semantic contribution to the word is different in different contexts. For example, the word “Apple” has three different “senses”: “Apple Brand”, “Apple (Fruit)” and “Apple Tree”. In the context “I am going to the ~ store now.”, the semantic of the word “apple” tends to the sense “Apple Brand”. The sememe’s contribution to the sense should be different.

To fill this gap, a “double attention” mechanism is proposed to capture the inequality of sememes, thereby the meanings of words can be more accurately represented. Specifically, we derive a double attention-based word embedding (DAWE) model. This model uses senses as a bridge in the process of encoding sememes into words, in which the word can be represented as a fusion of their different senses and a sense can be represented as the weighted sum of the sememes of the sense.

The original contributions of this work can be summarized as follows: (1) The proposed “double attention” mechanism captures the weight changes of the different sememes of a sense with context, as well as the word changes of the different senses within the word with context so that the obtained word vectors can be represented completely and accurately by sememes. (2) Two specific word training models are derived by combining the DAWE word encoding model with context-aware semantic matching. The experimental results of both word similarity task and word analogy reasoning task on the standard datasets show that the proposed models outperform previous models. (3) The proposed DAWE model is a general framework of encoding sememes into words and can be integrated with other existing word embedding models to provide more methods for word embedding.

2. Notation and Definition

The symbolic conventions that are used below are given here: $W$, $S$ and $X$ represent word set, sense set and sememe set, respectively. For each word $w \in W$, there are multiple senses $s_i^{(w)} \in S^{(w)}$, and $S^{(w)}$ represents the sense set corresponding to the word $w$; for each sense $s_i^{(w)}$, corresponding to several different sememes $x_i^{(s_i)} \in X_i^{(w)}$, $X_i^{(w)}$ represents the sememe set of the $i$th sense corresponding to the word $w$ and $C(w)$ represents the context word set corresponding to the word $w$. We use the bold form $w/s/x \in \mathbb{R}^D$ corresponding to $w/s/x$ to represent the vectors of word/sense/sememe, where $D$ is the dimension of those vectors.

**Definition 1.** Word Embedding.
As shown in Figure 3, for the text corpus C, word embedding maps each word \( w \in W \) to a continuous low-dimensional space \( \mathbb{R}^D \), while ensuring that the final embeddings (vectors) can represent the semantic relevance between words in the original text corpus C.

Definition 2. Encoding Words with Sememes.

It is a process of using sememes as a semantic supplement to encode words. In the process, word embedding can be simplified as the encoding of words from sememes to words, which means word vectors can be obtained by encoding corresponding sememes of words:

\[
 w = f_{X \rightarrow w}(X(w), \theta_{X \rightarrow w}),
\]

where \( \theta_{X \rightarrow w} \) represents the parameters when encoding sememe set \( X(w) \) to its corresponding word \( w \). \( f_{X \rightarrow w}(X(w), \theta_{X \rightarrow w}) \) can be a simple encoding function, such as sum operation \( f_{X \rightarrow w}(X(w), \theta_{X \rightarrow w}) = \sum_{i=1}^{[X(w)]} x_i(w) \) or average operation \( f_{X \rightarrow w}(X(w), \theta_{X \rightarrow w}) = \frac{1}{[X(w)]} \sum_{i=1}^{[X(w)]} x_i(w) \), or \( f_{X \rightarrow w}(X(w), \theta_{X \rightarrow w}) \) be neural networks, such as \( f_{X \rightarrow w}(X(w), \theta_{X \rightarrow w}) = \sigma(W \cdot X(w) + b) \), where \( \sigma \) denotes the activation function, \( W \) is the weight matrix and \( b \) is the bias.

Definition 3. Encoding Words with Sememes through Senses.

As shown in Figure 1, a word may consist of many different senses, each of which is described by several sememes. Therefore, this “word–sense–sememe” structure allows us to achieve the encoding process from sememes to words using senses as a semantic bridge, that is, the encoding process of the word \( w \) is represented as a mapping function of all its corresponding senses \( S(w) \). The formalization is as follows:

\[
 w = f_{S \rightarrow w}(S(w), \theta_{S \rightarrow w}),
\]
for each sense $s_{i}^{(w)} \in S^{(w)}$, it is encoded by all its corresponding sememes $X_{i}^{(w)}$:

\[ s_{i}^{(w)} = f_{X \rightarrow s} \left( X_{i}^{(w)}, \theta_{X \rightarrow s} \right), \]  

(3)

where $\theta_{S \rightarrow w}$ and $\theta_{X \rightarrow s}$ denote the trainable parameters.

The objective of this study is to find the $f_{S \rightarrow w}$ function and the $f_{X \rightarrow s}$ function in Equation (2) and Equation (3), while taking full advantage of the structure of the sememes.

3. Related Works

In this section, we mainly introduce the works related to this study, including classical word embedding models and the word embedding models that introduce internal semantic information of words and external semantic information (image, knowledgebase, etc.). These works are illustrated in Figure 4.

![Figure 4. Several typical works of word embeddings.](image)

3.1. Classical Word Embeddings

Word embedding aims to embed words into continuous low-dimensional, high-density semantic space. Early models usually use an NLM (neural language model) to generate word vectors (word-level word embedding vectors). The typical representative of them is the Word2Vec, which includes a CBOW model (continuous bag-of-words model) and a Skip-gram model (continuous skip-gram model), as shown in Figure 4. The key idea of Word2Vec is that the words with similar text contexts (or those words appearing in the same window that slides through the text with a size of k) should be close to each other in the semantic space, that is, their word vectors should be similar. As shown in Figure 3, the words “first”, “second” and “third” are close to each other in the semantic space because they have the same context “This is the ~ sentence”.

\[ L(H) = \sum_{u=3}^{U} \sum_{i=3}^{I} \log P(w_u, \ldots, w_i|w_u) \]  

(4)
Skip-gram is a model which predicts the context words (surrounding words) given a target word (the center word). It wants to maximize the likelihood function as follows:

$$L(H) = \sum_{t=k}^{n-k} \log P(w_{t-k}, \ldots, w_{t+k} | w_t) \approx \sum_{t=k}^{n-k} \log \prod_{w_c \in C(w_t)} P(w_c | w_t).$$  \hspace{1cm} (4)

where $n$ is the size of the text corpus, that is, the number of words contained in the corpus. $P(w_{t-k}, \ldots, w_{t+k} | w_t)$ denotes the probability of the context $[w_{t-k}, \ldots, w_{t+k}]$ being predicted by the target word $w_t$. $[w_{t-k}, \ldots, w_{t+k}]$ is the set of the first and last $k$ words of the current word $w_t$ in the text sequence and $k$ is the size of the context window. For example, for the text sequence, “I twisted an apple off the tree,” when the target word is “apple” and $k = 2$, then $[w_{t-k}, \ldots, w_{t+k}] = [\text{twisted}, \text{an}, \text{off}, \text{tree}].$ Based on the assumption of context independence, the probability of predicting context $[w_{t-k}, \ldots, w_{t+k}]$ by the target word $w_t$ can be converted to the product of the probability of predicting each word $w_c$ in the context (the co-occurrence probability of the target word and the context word):

$$P(w_{t-k}, \ldots, w_{t+k} | w_t) \equiv \prod_{w_c \in C(w_t)} P(w_c | w_t),$$ where $C(w_t) = [w_{t-k}, \ldots, w_{t+k}]$.

By introducing the negative sampling [12] method, the co-occurrence probability of each context word and the target word can be formalized by the following:

$$P(w_c | w_t) \equiv \sigma(w_c^T w_t) \prod_{w_{c'} \in \text{NEG}(w_t)} (1 - \sigma(w_{c'}^T w_t)), \hspace{1cm} (5)$$

where $\sigma(\cdot)$ denotes the sigmoid function and $\text{NEG}(w_t)$ is the negative word set for the target word $w_t$. The objective of negative sampling is to make the context word $w_c$ as close as possible to the target word $w_t$ in the semantic space and as far away as possible from the negative sample $w_{c'}$. It aims to make the co-occurrence probability of $w_c$ and $w_t$ ($\sigma(w_c^T w_t)$) greater than $w_c$ and $w_{c'}$ ($\sigma(w_{c'}^T w_t)$). Although Word2Vec strikes a good balance between efficiency and quality, the representation of low-frequency words remains a challenge due to the lack of adequate training for sparse words.

3.2. Word Embeddings with Internal Semantic Information

In addition to the word co-occurrence, the internal features of words have also been shown to contribute to word embedding. Related works can be roughly divided into three categories: models based on morphological information, models based on character information and models based on subword information. Examples of “morphological information”, “character information” and “subword information” are illustrated in Figure 4, where morphological information mainly refers to the features from components (i.e., prefix, root and suffix) of the word. Bian, Gao and Liu [1] utilized morphological (prefix, root and suffix), syntactic and semantic knowledge to achieve high-quality word embeddings. Chen et al. [13] and Sun et al. [14] performed character-level embedding and word embeddings obtained by fusing character features and word features. Xu et al. [15] also used a character-level embedding, and the weight information of different characters was taken into account in the fusion process. Cao and Lu [16] combined both the morphological information of the word and the information of the character-level and captured the structure information of the context by adding the subword information (character n-gram, root/affix and inflections). To better discover the laws of language for word embedding, Li et al. [17] tried to discover the relationship between morphology and semantics in language expression and summarized 68 implicit morphological relationships and 28 display semantics relationships.

Actually, in Chinese, characters are not the smallest granularity units, but strokes. On top of this, there are structures such as radicals and components. Shi et al. [18] and Yin et al. [19] added the features of radicals of the characters inside target words to CBOW model. Yu et al.’s [20] method, regarded as a more refined version of those of Shi, Zhai, Yang, Xie and Liu [18] and Yin, Wang, Li, Li and Wang [19], captured not only the radical information but also other components inside the
character. To better exploit the structural information inside the character, Cao et al. [21] proposed to use the set of Stroke n-gram information of characters to supplement the semantics of characters.

The methods mentioned above only use the semantic information of the word itself, such as from word-level embeddings to character-level embeddings or other more fine-grained embeddings. However, the semantic information obtained from the word is limited. Besides, the models are influenced by the formation of language, the characteristics of language, etc., thus it is difficult to generalize to other languages.

3.3. Word Embeddings with External Semantic Information

A lot of semantic information related to words is now emerging, such as images with text labels, as well as some semantic knowledge bases including WordNet [22], BabelNet [23], ConceptNet [24] and HowNet [11]. These semantic data should help us improve the accuracy of word vectors.

A large and growing body of literature has researched on joining external semantic information for word embedding. Liu et al. [25] proposed a character-level embedding model that attempts to capture the common structure between characters from visual features by using morphological images corresponding to characters. Wang, Zhang and Zong [26] proposed a word-level embedding model, which uses images from the real world as a complement to text semantics, rather than directly replacing text semantic information with visual feature information. In terms of considering external semantic knowledge base, Yang and Sun [9] used Tongyici Cilin [27] whose purpose is to make the words with the same semantic classification in the Tongyici Cilin close to each other (Tongyici Cilin is a Chinese semantic knowledge base based on synonym sets, which can classify words according to their semantics). Mancini, Camacho-Collados, Iacobacci and Navigli [6] used BabelNet to annotate the different senses of words and then performed joint learning to get word and sense embeddings. Tissier, Gravier and Habrard [8] introduced the concepts of “strong pairs” and “weak pairs” from dictionary entries, so as to better distinguish the relative intensity of word pairs in the semantic space. Liu et al. [28] proposed a knowledge-enabled language representation model with knowledge graphs (KGs), in which KG triples are injected into the sentences as domain knowledge. Niu, Xie, Liu and Sun [7] proposed the sememe-encoded word representation learning (SE-WRL) model. The SE-WRL model embeds words by encoding sememe in word–sense–sememe knowledge of HowNet. Since the word–sense–sememe is an intuitive form of organizing words, it is easily organized and interpretable [11] and has a wide range of potential uses.

Specifically, three SE-WRL models are mentioned: simple sememe aggregation model (SSA), sememe attention over context model (SAC) and sememe attention over target model (SAT).

1. The SSA model simply represents vector $w$ of each word as the average of all its sememe vectors, as shown in Equation (6).

$$w = \frac{1}{m} \sum_{s^{(w)}_{i} \in S^{(w)}} \sum_{x^{(w)}_{j} \in X^{(w)}} x^{(s_{i})}_{j},$$

where $m$ is the number of sememes of the word $w$.

2. Based on the SSA model, Niu, Xie, Liu and Sun [7] developed a SAC model and a SAT model that can distinguish different word meanings:

$$\text{att}(s^{(wu)}_{i}, w_{v}) = \frac{\exp(w_{v} \cdot s^{(wu)}_{i})}{\sum_{k=1}^{S^{(wu)}} \exp(w_{v} \cdot s^{(wu)}_{k})}.$$  

In the above formula, if $w_{v}$ represents the target word $w_{t}$ and $w_{u}$ represents the context $w_{c}$, then it is SAC model, while, if $w_{v}$ represents context $w_{c}$ and $w_{u}$ represents the target word $w_{t}$, then it is the SAT model.
In this study, we used both sememe and word–sense–sememe structure as external supplements to refine the process of word embedding. Different from SE-WRL, our model captures the weight changes of different sememes under the same sense over different contexts, while using sememe to encode words.

4. Methodology

The proposed double attention-based word embedding (DAWE) model is derived from SE-WRL, where “double attention” refers to sense-level attention and sememe-level attention. The model assumes that the meaning of a word in a sentence is composed of senses with different weights, and each sense is composed of different sememes with different weights. In addition, the study assumes that a better way to disambiguate the senses of words in different contexts is to carefully design the process of constituting senses from sememes.

DAWE model, introduced in Section 4.1, is a general framework for encoding sememes into words. Double attention over context model (DAC) introduced in Section 4.2 and double attention over target model (DAT) introduced in Section 4.3 are two specific word training models that are obtained by integrating the DAWE model through context-aware semantic matching.

4.1. Double Attention-Based Word Embedding Model

To encode the semantics of sememes into words through the “word–sense–sememe” structure, a DAWE model is developed, as shown in Figure 5.

![Figure 5. DAWE model.](image)

In the DAWE model, a “double attention” architecture is adopted: (1) Sense-level attention is to capture senses weight changes with context. A word may have different meanings in different contexts, but those meanings are not isolated. We argue that the meaning of a word in a specific context should be a fusion of different senses. As the context changes, the fusion weight of the senses also changes accordingly. (2) Sememe-level attention is to capture the weight change of sememes with context. In the SE-WRL model, each weight of sememes that constitutes the senses is thought as equivalent. Actually, when a sense presents different meanings, the weights of sememes under a sense should be different.

As shown in Figure 5, DAWE is a word embedding model based on the word–sense–sememe structure, as well as a word sense disambiguation (WSD) model. In DAWE, sememes constitute the different senses of a word, and then different senses reconstitute into word meanings that are relevant to the textual context.
The purpose of word embedding is to keep semantic relevance of words while words are embedded into a unified semantic space. However, word embedding has the semantic confusion defect of representing all the meanings of a word in the same vector. To remedy such deficiencies, the different meanings of words need to be modeled separately to overcome the chaos of word embedding. The research suggests that better decomposition of word meanings combined with context leads to better representations of word meanings. WSD is to distinguish the different senses of words in different contexts, which can be roughly divided into unsupervised methods and knowledge-based methods. DAWE uses a knowledge-based approach to disambiguate the different senses of words in context using weighted sememes for the presenting of senses under a word. As a word embedding model based on knowledge, the objective of DAWE is the same as conventional approaches based on knowledge, which is to have words with the same semantics close to each other and words with different semantics away from each other [29].

According to the location of the object of the “attention”, DAWE models can be extended to double attention over context model (DAC) and double attention over target model (DAT). Figures 6 and 7 illustrate the relationships and differences between the two models.

### 4.2. Double Attention over Context Model

As shown in Figure 6, DAC consists of two parts: encoding part and training part, which correspond to the DAWE encoding framework and Skip-gram training framework respectively.

For each context word \( w_c \in C(w) \) \((C(w) = [w_{t-k}, \ldots, w_{t-1}, w_{t+1}, \ldots, w_{t+k}]\), where \( k \) is the size of the context window), we have:

\[
w_c = \sum_{i=1}^{\left| S(w_c) \right|} \text{att}\left(s_i^{(w_c)}, w_t\right) s_i^{(w_c)},
\]

where \( \text{att}\left(s_i^{(w_c)}, w_t\right) \) denotes that the target word \( w_t \) is used as attention to calculating the weight of the \( i \)th sense of the context word \( w_c \), as follows:

\[
\text{att}\left(s_i^{(w_c)}, w_t\right) = \frac{\exp\left(s_i^{(w_c)} \cdot w_t\right)}{\sum_{j=1}^{\left| S(w_c) \right|} \exp\left(s_j^{(w_c)} \cdot w_t\right)},
\]

where \( s_i^{(w_c)} \) denotes the value used in the calculation of weight, which is obtained by the sememe embeddings of the sememe set \( X_i^{(w_c)} \) corresponding to the sense \( s_i^{(w_c)} \). It can be formalized by the following:

\[
s_i^{(w_c)} = \sum_{j=1}^{\left| X_i^{(w_c)} \right|} \text{att}\left(x_j^{(s_i)}, w_t\right) x_j^{(s_i)}.
\]

Similar to \( \text{att}\left(s_i^{(w_c)}, w_t\right) \), \( \text{att}\left(x_j^{(s_i)}, w_t\right) \) indicates that the target word \( w_t \) is used as attention to calculating the weight of the \( j \)th sememe in the \( i \)th sense of the context word \( w_c \), as follows:

\[
\text{att}\left(x_j^{(s_i)}, w_t\right) = \frac{\exp\left(x_j^{(s_i)} \cdot w_t\right)}{\sum_{k=1}^{\left| X_i^{(w_c)} \right|} \exp\left(x_k^{(s_i)} \cdot w_t\right)}.
\]

DAWE is a two-layer encoding framework. The first layer is sense encoding, which corresponds to Equation (10) and Equation (11). In the first layer, the sememe embeddings are used as input, and then the sense embeddings are obtained through sememe-level attention. The second layer is word encoding, corresponding to Equation (8) and Equation (9). In the second layer, the sense embeddings
obtained by Equation (10) are used as input, and then the word embeddings are obtained through sense-level attention. In DAC, the target word $w_t$ is used to guide the generation of word vectors of context words. Under this attention mechanism, if the sememe vectors and sense vectors of the context word are more relevant to the target word vectors, the corresponding sememes and senses will get higher weight. This is similar to the idea in Word2Vec that the more similar words are closer in the semantic space. In this way, the different senses of context words can be disambiguated too.

$$\text{Figure 6. Double attention over context model (DAC).}$$

$$
\text{Figure 7. Double attention over target model (DAT).}
$$
4.3. Double Attention over Target Model

DAT is a variant of DAC, and word vectors are also encoded by the DAWE model and trained by the Skip-gram model. In contrast to DAC, DAT takes context embedding as attention to guide the generation of the word vector of the target word. The model structure of DAT is shown in Figure 7.

For each target word \( w_t \in W \), we have:

\[
 w_t = \sum_{i=1}^{||S(w_t)||} \text{att}(s_i, C(w_t)) \cdot s_i, \tag{12}
\]

where \( \text{att}(s_i, w_{\text{context}}) \) denotes that the context word set \( C(w_t) \) is used as attention to calculating the weight of the \( i \)-th sense of \( w_t \), as follows:

\[
 \text{att}(s_i, w_{\text{context}}) = \frac{\exp(s_i \cdot C(w_t))}{\sum_{j=1}^{||S(w_t)||} \exp(s_j \cdot C(w_t))}, \tag{13}
\]

The calculation of \( s_i \) is similar to DAC (Equation (10) and Equation (11)), which is the weighted sum of all sememe embeddings of sememe set \( X_i \) corresponding to sense \( s_i \), where \( C(w_t) \) denotes the context, and its corresponding word vector is obtained by the average of all context word vectors in the context window. It is formalized by the following:

\[
 C(w_t) = \frac{1}{2k} \sum_{j=1-k}^{j=t+k} w_j, \ j \neq t, \tag{14}
\]

where \( k \) is the size of the context window.

DAT uses context as attention and is richer in contextual semantics than DAC, hence it should be more conducive to the choice of sememes and senses.

4.4. Optimization

This section takes DAT as an example to illustrate the training process of the proposed model. As shown in Figure 8, in DAT’s pre-processing phase, each word in the vocab needs to be annotated according to “word–sense–sememe” knowledge (association is established among sememe, sense, and word). Then, in DAWE framework, the target word \( w_t \) is encoded through the “double-attention” mechanism. In DAT, the context (Equation (14)) is used to guide the encoding of the target word (see Section 4.3 for details). The parameters that need to be optimized include not only word embeddings but also sense embeddings and sememe embeddings:

\[
 \begin{align*}
 w_i & := w_t + \alpha \cdot \Delta w_t \\
 w_{t+k} & := w_{t+k} + \alpha \cdot \Delta w_{t+k} \\
 S(w_i) & := S(w_i) + \alpha \cdot \Delta S(w_i), \ i \in \{1, 2, \ldots, k\} \\
 X(w_i) & := X(w_i) + \alpha \cdot \Delta X(w_i)
\end{align*}
\tag{15}
\]

where \( \alpha \) denotes the learning rate; \( k \) is the size of the context window (in Figure 8, \( k = 2 \)); \( S(w_i) \) denotes the vector set of senses corresponding to \( w_i \); and \( X(w_i) \) denotes the sememe vector set corresponding to \( w_i \).
optimization is the same as classical Skip-gram (Equation (4)); however, the parameters that need to be optimized include not only word embeddings but also sense embeddings and sememe embeddings:

\[
\begin{align*}
\dot{w}_i & := w_i + \alpha \cdot \Delta w_i \\
\dot{w}_{i+1} & := w_{i+1} + \alpha \cdot \Delta w_{i+1} \\
S(i) & := S(i) + \alpha \cdot \Delta S(i) \\
X(i) & := X(i) + \alpha \cdot \Delta X(i)
\end{align*}
\]

where \(\alpha\) denotes the learning rate; \(k\) is the size of the context window (in Figure 8, \(k=2\)); \(S(i)\) denotes the vector set of senses corresponding to \(w_i\); and \(X(i)\) denotes the sememe vector set corresponding to \(w_i\).

**Figure 8.** Training with DAT.

The optimization process of DAC is similar to DAT, and is not be elaborated upon in this paper.

### 5. Experiments and Results

Our experiments were conducted on a Chinese word embedding task. Model performances were examined with two tasks: the word similarity task and the word analogy task. In this section, we first introduce the experimental datasets, including the training set and the evaluation set in the two evaluation tasks. Next, we introduce the experimental settings, including the selection of baselines and the setting of parameters. Finally, we present the metrics and results of the two evaluation tasks.

#### 5.1. Datasets

For training, HowNet annotated text corpus Clean-SogouT1 [7] was selected to train our model. Each word in the vocab of Clean-SogouT1 dataset is annotated in this form: \((w, \text{sense num} (s), \text{sememe number of first sense}, \text{sememe set of first sense} X^{(1)}(w), \ldots, \text{sememe number of the \(s\)th sense}, \text{sememe set of \(s\)th sense} X^{(s)}(w))\). The example of “Apple” in Figure 1 can be represented as \((\text{“Apple”}, 3, 5, (\text{“computer”}, \text{“PatternValue”}, \text{“able”}, \text{“bring”}, \text{“SpeBrand (specific brand)”}), 1, (\text{“fruit”}), 3, (\text{“fruit”}, \text{“reproduce”}, \text{“tree”})\)). The basic statistics of the dataset are shown in Table 1. Table 1 shows that more than 60% of the words in the dataset have more than two senses, which suggests that dynamic sense disambiguation is necessary for improving word embedding models. Following Niu, Xie, Liu and Sun [7], this study removed words from the vocab set with word frequency under 50.

**Table 1.** Text corpus statistics. AS/PW, average senses/per word; AS/PS, average sememes/per sense; PWMS, percentage of words that have multiple senses.

| Text Corpus       | Words | Vocab Size | Sememes | AS/PW | AS/PS | PWMS  |
|-------------------|-------|------------|---------|-------|-------|-------|
| Clean-SogouT1     | 1.8B  | 350K       | 1889    | 2.683 | 1.701 | 60.78%|

For evaluation, we chose the Chinese word similarity (CWS) dataset and the Chinese word analogy (CWA) dataset provided by Niu, Xie, Liu and Sun [7] to evaluate the performance of the models in the word similarity task and the word analogy reasoning task. The CWS datasets Wordsim-240 and Wordsim-297 contain 240 similar word pairs and 297 similar word pairs, respectively, and each word pair in the CWS dataset has its corresponding similarity score, e.g., “consumer, customer, 8.4”. Each entry in the CWA dataset is composed of four words \(w_1, w_2, w_3, w_4\). The form of word analogy
is: $w_2 - w_1 \cong w_4 - w_3$, such as the classic example: $w_{king} - w_{man} \cong w_{queen} - w_{woman}$. The bold form $w \subset \mathbb{R}^D$ denotes the embedding of the word $w$. The statistics of the CWA dataset used in this study is shown in Table 2.

Table 2. Chinese word analogy dataset, which contains three analogy types: capitals of countries (Capital), e.g., $w_{London} - w_{England} \cong w_{Beijing} - w_{China}$; cities in states (City), e.g., $w_{Jacksonville} - w_{Florida} \cong w_{San Francisco} - w_{California}$ and family relationships (Relationship), e.g., $w_{Father} - w_{Mother} \cong w_{Son} - w_{Daughter}$.

| Capital | City | Relationship | All |
|---------|------|--------------|-----|
| 677     | 175  | 272          | 1124|

5.2. Experimental Settings

In the experiments, we chose Skip-gram (the basic training framework of our models), CBOW (another model in Word2Vec, for comparison with Skip-gram) and GloVe [30] (different from the calculation method of Skip-gram in the local context window, Glove obtains the word embeddings by global matrix decomposition) as the comparison models. We also chose the SSA model (encoding words with sememes without attention mechanism), SAC (for comparison with DAC) and SAT (for comparison with DAT) proposed in Niu, Xie, Liu and Sun [7] as our baselines.

Following Niu, Xie, Liu and Sun [7], the vector dimensions of word embeddings, sense embeddings and sememe embeddings were set to 200; the size of the context window was set to 8; the initial learning rate was 0.025; and the number of negative samples was set to 25 in the negative sampling method. For the SAT and DAT, we set the context embedding window size to 2.

Our DAWE models were implemented based on the code of the SE-WRL model (https://github.com/thunlp/SE-WRL). The benchmark models and our models were trained on the same machine.

5.3. Word Similarity

In this section, this study examine the quality of word embeddings through the performance of the proposed models in word similarity tasks. In the evaluation of the word similarity tasks, we used the cosine value between the vectors of two words as their similarity scores to obtain the similarity ranking of all pairs of words in the benchmark datasets (Wordsim-240 and Wordsim-297). By calculating the Spearman correlation coefficient between the similarity ranking obtained by our models and the similarity ranking in the benchmark datasets, we could evaluate the performance of the model in word similarity tasks. The higher the Spearman correlation coefficient is, the better the model performs in the word similarity task.

Table 3 shows the evaluation results on word similarity tasks. (1) On the Wordsim-240 dataset and Wordsim-297 dataset, our models performed better compared to the baseline models. This shows that distinguishing the sememes within the senses can help us to present different senses of the word more accurately and deeply. (2) DAT performed better than DAC. DAT takes context embedding as attention to guide the semantic generation to the target words, thus it can better capture contextual semantic information. Therefore, when the training of words is sufficient, the results of DAT will be better than DAC.
Table 3. Evaluation results of word similarity tasks.

| Model   | Wordsim-240 | Wordsim-297 |
|---------|-------------|-------------|
| CBOW    | 57.987      | 62.063      |
| GloVe   | 57.618      | 57.107      |
| Skip-gram | 55.279     | 60.565      |
| SSA     | 60.410      | 60.167      |
| SAC     | 57.574      | 57.825      |
| SAT     | 60.480      | 62.280      |
| DAC     | 60.410      | 60.167      |
| DAT     | 61.162      | 63.327      |

5.4. Word Analogy

In this section, we examine the quality of word embeddings by the performance of the models in the word analogy reasoning task. In the Chinese word analogy reasoning task, each analogy sample consists of two-word pairs \((w_1, w_2)\) and \((w_3, w_4)\), which satisfy: \(w_2 - w_1 \approx w_4 - w_3\), i.e. \(w_2 - w_1 + w_3 \approx w_4\). Therefore, in the word analogy reasoning task, the score of the candidate word is calculated by replacing \(w_4\) with the candidate word \(w\) and by the following formula:

\[
S_A(w) = \cos(w_2 - w_1 + w_3, w).
\] (16)

After obtaining the ranking of all candidate words, the experiment chose top-ranked words and evaluated the performance of the model by calculating accuracy and mean rank metrics. The higher is the accuracy and the lower is the mean rank, the better is the model.

The results of the word analogy reasoning task are shown in Table 4. From the evaluation results of the word analogy task, we can conclude that:

Table 4. Evaluation results of word analogy task.

| Model   | Capital 677 | City 175 | Relationship 272 | All 1124 | Capital 677 | City 175 | Relationship 272 | All 1124 |
|---------|-------------|----------|------------------|---------|-------------|----------|------------------|---------|
| CBOW    | 45.05       | 86.85    | 84.19            | 61.03   | 60.28       | 1.43     | 41.87            | 46.66   |
| GloVe   | 62.03       | 83.42    | 82.35            | 70.28   | 17.09       | 1.77     | 14.28            | 14.02   |
| Skip-gram | 60.26      | 96.00    | 77.57            | 70.01   | 78.67       | 1.05     | 2.98             | 14.45   |
| SSA     | 72.67       | 80.00    | 74.63            | 74.28   | 21.05       | 7.21     | 2.74             | 13.51   |
| SAC     | 66.24       | 92.28    | 71.87            | 71.66   | 40.86       | 5.74     | 2.56             | 9.80    |
| SAT     | 71.64       | 87.14    | 74.44            | 74.73   | 14.79       | 2.07     | 2.34             | 9.80    |
| DAC     | 68.53       | 93.14    | 72.24            | 73.26   | 14.10       | 1.15     | 2.74             | 9.34    |
| DAT     | 74.00       | 91.42    | 75.36            | 77.04   | 8.87        | 1.71     | 2.58             | 6.23    |

(1) In the word analogy reasoning task, our models are significantly better than the previous models. The accuracy of DAC is 2% higher than that of SAC, and the accuracy of DAT is 3% higher than that of SAT. DAC has increased more than 4% compared to the SAC model and DAT has increased more than 3% compared to the SAT model of mean rank. The experimental results show that both DAC and DAT are more conducive to the accurate description of senses by distinguishing the internal sememes.

(2) Our models perform well in the class of Capital, which is the collection of groups of capital and country around the world. Most of the words of the capital names have distinct meanings in various contexts, such as the word “Washington” may be the name of a capital city, a state, a university, a hotel, or a people. In the training process, the proposed model can dynamic adjustment the weights of both senses and sememes by the “double-attention” mechanism, hence offering more powerful ability on the embedding of those words.
(3) Although the performances of our models are not the best in the classes of City and Relationship, our models are more robust in the overall performance of accuracy and mean rank.

(4) DAWE models are significantly improved in the performance of the word analogy reasoning task, but only a small increase in performance in the word similarity task. Since Skip-gram trains word vectors based on context, the more similar the context is, the closer the word vector is in the semantic space. Thus, with sufficient training, there is no significant difference among the performance of these Skip-gram-based models for the word similarity task. By adding sememe-level attention, our models can more accurately express the sense of the word, resulting in better results in the word analogy reasoning task requiring higher semantic accuracy.

6. Discussions

6.1. Case Study

To illustrate the dynamic semantic generation of our models, we select some specific cases for analysis. Tables 5–7 lists the relative weights of the different senses of the word “Apple” (Sense 1: “Apple Brand” (Sememe: “computer”, “PatternValue”, “able”, “bring” and “SpeBrand (specific brand)”; Sense 2: “Apple” (Sememe: “fruit”; Sense 3: “Apple Tree” (Sememe: “fruit”, “reproduce” and “tree”)) in a specific context and the relative weights of different sememes within the senses. Those weights are calculated by sense-level attention and sememe-level attention of DAT. Tables 5–7 show that: (1) Our model correctly distinguishes the different senses of “Apple” from different contexts. This shows the power of our model in word sense disambiguation (WSD). (2) In the sense “Apple Brand”, the sememe “SpeBrand” gets a large weight. This is consistent with our description in the Introduction. In the process of sense construction, the distribution of weights between sememes should be unequal. (3) When the meanings of “Apple” changing with different contexts (the meaning of “Apple” changes when the sentence changes), both the sense items of the word and the sememe items in each sense of the word do not change, what changes with the context are the weights of those senses and the weights of those sememes. The model of this paper is trained on the large text corpus Clean-SogouT1, and the learned word vectors and model parameters are consistent with the feature distribution of the entire corpus. As a result, the sense representation inside the words will tend to be stable, that is, the weight distribution of sememes inside the senses will also be stable (sememe consists of sense, sense and then word).

Table 5. “You can like apple (Apple Brand) computers, just don’t vilify other brands.”

| Senses       | Sememes   |       |       |       |       |
|--------------|-----------|-------|-------|-------|-------|
| Apple Brand  | bring     | 5.15  | PatternValue | 0.00  | SpeBrand | 6.77 | computer | 0.31  | able     | 8.06  |
| Apple        | fruit     | 0.00  | fruit    | 14.93 | reproduce | 0.00  |          |       |          |       |
| Apple Tree   | tree      | 6.38  | fruit    | 21.28 | reproduce | 0.00  |          |       |          |       |

Table 6. “I just hit you with an apple (Apple) core.”

| Senses       | Sememes   |       |       |
|--------------|-----------|-------|-------|
| Apple Brand  | bring     | 4.22  | PatternValue | 0.00  |
| Apple        | fruit     | 1.20  |       | 4.55  |
| Apple Tree   | tree      | 20.18 | fruit    | 0.00  |

Table 7. “Apple (Apple Brand) is a fruit.”

| Senses       | Sememes   |       |
|--------------|-----------|-------|
| Apple Brand  | bring     | 5.15  |
| Apple        | fruit     | 0.00  |
| Apple Tree   | tree      | 19.93 |
Table 7. “There are many kinds of high-quality apple (Apple Tree) seedlings in southeast Asia.”

| Senses   | Sememes         |     | Sememes         |     | able  |
|----------|-----------------|-----|-----------------|-----|-------|
| Apple Brand | bring          | 0.00 | PatternValue    | 0.00 | computer | 1.85 | 5.97 |
| Apple    | fruit           | 5.58 |                 |     |       |
|                | 0.00            |     |                 |     |       |
| Apple Tree | tree           | 0.08 | fruit           | 12.60 | reproduce | 12.30 | 0.00 |

In the above cases, we take the word “Apple (Apple/Apple Brand/Apple Tree)” as an example to examine the weight distribution of sememes and verify the effectiveness of our model in WSD. We take the word “Notebook (Notebook/Laptop Computer)” as an example to study the impact of sememe’s weight distribution in a specific context. As shown in Table 8, when the meaning of word “Notebook” in the context tends to the sense “Laptop Computer”, we observe the following:

Table 8. The impact of context on the weight of sememes. The values in this table represent relative weights and the relative weight of the sense “Notebook” is 0. Word: “Notebook” (Sense 1: “Notebook” (Sememe: “account”); Sense 2: “Laptop Computer” (Sememe: “bring”, “PatternValue”, “computer” and “able”)).

| Context                                                                 | Bring | PatternValue | Computer | Able |
|------------------------------------------------------------------------|-------|--------------|----------|------|
| Laptop Computer (0.57): Those who want to buy a notebook (Laptop Computer) can write down my contact information | 4.69  | 0.00         | 1.82     | 5.62 |
| Laptop Computer (4.59): HP business notebook (Laptop Computer) has industry-leading security technology | 5.85  | 0.00         | 2.07     | 4.49 |
| Laptop Computer (1.21): Our shop can provide you with notebook (Laptop Computer) repair service | 4.88  | 0.00         | 1.98     | 4.10 |
| Laptop Computer (4.53): There are two notebook (Laptop Computer) computers in the computer room. They are very old and slow | 5.82  | 0.82         | 0.00     | 6.63 |
| Laptop Computer (8.73): Everyone has the chance to get refrigerator, notebook (Laptop Computer) computer, LCD TV, etc | 5.82  | 0.98         | 0.00     | 4.97 |
| Laptop Computer (6.77): This notebook (Laptop Computer) computer with a strong display is a real eye-opener | 5.20  | 0.43         | 0.00     | 6.20 |

(1) When the word “Notebook” and the word “Computer” appear together, that is, “Notebook Computer”, the weight of the sememe “computer” is the lowest among all the sememes of the sense “Laptop Computer”. It can be explained that, when “Notebook Computer” appear together, “Notebook” is mainly used as a modifier of “Computer” to indicate that “Computer” is light, thin and portable. Therefore, “Notebook” will have less “computer” meaning.

(2) When “Notebook” appears alone, sememe “computer” has more weight than when “Notebook Computer” appear together. At this point, “Notebook” no longer appears as a modifier of “Computer” but as a separate entity, thus it should cover the semantics that tends to favor “computer”.

(3) When “Notebook” appears alone, the weight of the sense “Laptop Computer” is generally lower than when “Notebook Computer” appear together because, when “Notebook Computer” appear together, the context carries more semantics that tends to the sense “Laptop Computer”, thus “Laptop Computer” is generally weighted more heavily. (Note the second example of Table 8, where the weight
of “Laptop Computer” reached 4.59. This is because “HP” is a computer brand, which results in the “Laptop Computer” weight more than the other case of “Notebook” appearing alone).

The results in Table 8 also show the effectiveness of our model. In the DAWE model, the representation of words depends on senses, the weight distribution of the sememes cannot directly determine the final representation of words. As the word “Notebook” appears alone, the weight of the sense “Laptop Computer” is lower than that when “Notebook Computer” appear together, although the weight of the sememe “Computer” is higher than when “Notebook Computer” appear together.

In summary, in the training process of word embeddings, the semantics of words are affected not only by the semantic accumulation in corpus, but also by the context in the current slide window. (1) The impact of semantic accumulation is mainly reflected in the gradual stabilization of the representation of the inherent senses within the word. As shown in the examples in Tables 5–7, the weight distribution of the sememes used to represent the internal senses of the word “Apple” is consistent in different contexts. (2) The current context is mainly used to select the appropriate senses and can affect the weight distribution of sememes. As shown in Tables 5–7, although the representation of the inherent senses inside the word “Apple” tends to stable, the weight of these senses is varied in different sentences. Besides, the senses of the target word “Notebook” and the weights of their sememes in Table 8 also illustrate this point.

6.2. Integrating DAWE with Other Models

DAWE is a general encoding framework. In this paper, we integrate and train DAWE based on the Skip-gram model. DAWE can be extended for other models many by the following steps:

1. Data pre-processing. Using “word–sense–sememe” knowledge to annotate text corpus.
2. Determine the encoding “target” of DAWE. For example, in DAC, the “target” is the context, while, in DAT, the “target” is the target word.
3. Determine the “object” of “double-attention”. For example, in DAC, the “object” is the target word, while, in DAT, the “object” in the context.
4. Forward propagation (encoding). According to the “target” and “object” determined in Steps 2 and 3, in the DAWE framework, “object” is used to guide the encoding of the “target” through the “double-attention” mechanism.
5. Back propagation. Model parameters (word embeddings, sense embeddings and sememe embeddings) are updated according to the model optimization objective.

Among them, Steps 1 and 5 are relatively easy to implement. The core step is Step 4, which depends on Steps 2 and 3. Therefore, in expanding DAWE, the parts that are difficult and require careful design are Steps 2 and 3. Once Steps 2 and 3 are established, DAWE can be easily extended to other models.

7. Conclusion and Future Work

In this paper, double attention-based word embedding (DAWE) model is proposed to encode sememes into words by a “double attention” mechanism, resulting in going deep into the senses of a word to describe the word. Our proposed DAWE model is a general framework that can be applied to other existing word embedding training frameworks, such as Word2Vec. In this paper, we extend the DAWE model to get two specific training models. In the experiments of word similarity task and word analogy task, the validity of our models was demonstrated. To further explore the models proposed in this paper, some cases were analyzed in the experiment. The results show that word semantics are not only affected by the global semantic accumulation, but also by the context of a word. Experimental results show that DAWE models can effectively capture the semantic changes of words through dynamic semantic generation, which means that our model is also effective in word sense disambiguation. The findings of this study suggest it could get performance improvement of NLP tasks if words are processed in a more fine-grained perspective.
A limitation of this study is that the DAWE model requires more training time than baseline models because it increases training parameters as it integrates the “double attention” mechanism. Additionally, the values of hyperparameters in this study are set following previous research; further experimental investigations are needed to estimate the impacts of those hyperparameters.

**Author Contributions:** Conceptualization, S.L. and H.Y.; methodology and software, R.C.; writing—original draft preparation, R.C.; and writing—review and editing, S.L., H.Y., B.W., J.G. and L.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the NSF of China (Grant Nos. 61972365, 61673354, 41801378 and 61672474), and Open Fund of Key Laboratory of Urban Land Resources Monitoring and Simulation, Ministry of Natural Resources (Grant No. KF-2019-04-033).

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

**References**

1. Bian, J.; Gao, B.; Liu, T.-Y. *Knowledge-Powered Deep Learning for Word Embedding*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 132–148.
2. Cao, S.; Lu, W. *Improving Word Embeddings with Convolutional Feature Learning and Subword Information*; AAAI: San Francisco, CA, USA, 2017; pp. 3144–3151.
3. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems, Sydney, Australia, 12–15 December 2013; Volume 2, pp. 3111–3119.
4. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. In Proceedings of the International Conference on Learning Representations, Scottsdale, Arizona, 2–4 May 2013.
5. Goikoetxea, J.; Soroa, A.; Agirre, E. Bilingual embeddings with random walks over multilingual wordnets. *Knowl. Based Syst.* 2018, 150, 218–230. [CrossRef]
6. Mancini, M.; Camacho-Collados, J.; Jacobacci, I.;Navigli, R. Embedding Words and Senses Together via Joint Knowledge-Enhanced Training. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, BC, Canada, 3–4 August 2017; pp. 100–111.
7. Niu, Y.; Xie, R.; Liu, Z.; Sun, M. Improved word representation learning with sememes. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 2049–2058.
8. Tissier, J.; Gravier, C.; Habrard, A. Dic2vec: Learning Word Embeddings using Lexical Dictionaries. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2017), Copenhagen, Denmark, 7–11 September 2017; pp. 254–263.
9. Yang, L.; Sun, M. Improved learning of chinese word embeddings with semantic knowledge. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*; Springer: Guangzhou, China, 13–14 November 2015; pp. 15–25.
10. Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; Liu, Q. ERNIE: Enhanced Language Representation with Informative Entities. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 1441–1451.
11. Dong, Z.; Dong, Q. HowNet-a hybrid language and knowledge resource. In *Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering, Proceedings 2003*; IEEE: Piscataway, NJ, USA, 2003; pp. 820–824.
12. 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*; The MIT Press: Cambridge, MA, USA, 2013; pp. 3111–3119.
13. Chen, X.; Xu, L.; Liu, Z.; Sun, M.; Luan, H. Joint learning of character and word embeddings. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
14. Sun, F.; Guo, J.; Lan, Y.; Xu, J.; Cheng, X. Inside out: Two jointly predictive models for word representations and phrase representations. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016.
15. Xu, J.; Liu, J.; Zhang, L.; Li, Z.; Chen, H. Improve chinese word embeddings by exploiting internal structure. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, CA, USA, 12–17 June 2016; pp. 1041–1050.

16. Cao, S.; Lu, W. Improving word embeddings with convolutional feature learning and subword information. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.

17. Li, S.; Zhao, Z.; Hu, R.; Li, W.; Liu, T.; Du, X. Analogical reasoning on chinese morphological and semantic relations. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 2, pp. 138–143.

18. Shi, X.; Zhai, J.; Yang, X.; Xie, Z.; Liu, C. Radical embedding: Delving deeper to chinese radicals. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, Beijing, China, 26–31 July 2015; Volume 2, pp. 594–598.

19. Yin, R.; Wang, Q.; Li, P.; Li, R.; Wang, B. Multi-granularity chinese word embedding. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–4 November 2016; pp. 981–986.

20. Yu, J.; Jian, X.; Xin, H.; Song, Y. Joint embeddings of chinese words, characters, and fine-grained subcharacter components. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9–11 September 2017; pp. 286–291.

21. Cao, S.; Lu, W.; Zhou, J.; Li, X. Cw2vec: Learning chinese word embeddings with stroke n-gram information. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.

22. Miller, G.A. Wordnet: A lexical database for english. Commun. ACM 1995, 38, 39–41. [CrossRef]

23. Navigli, R.; Ponzetto, S.P. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif. Intell. 2012, 193, 217–250. [CrossRef]

24. Speer, R.; Chin, J.; Havasi, C. Conceptnet 5.5: An open multilingual graph of general knowledge. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.

25. Liu, F.; Lu, H.; Lo, C.; Neubig, G. Learning character-level compositionality with visual features. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017; Volume 1, pp. 2059–2068.

26. Wang, S.; Zhang, J.; Zong, C. Learning multimodal word representation via dynamic fusion methods. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.

27. Mei, J.; Zhu, Y.; Gao, Y.; Yin, H. Tongyici Cilin (Dictionary of Synonymous Words); Shanghai Cishu Publishing: Shanghai, China, 1983.

28. Liu, W.; Zhou, P.; Zhao, Z.; Wang, Z.; Ju, Q.; Deng, H.; Wang, P. K-Bert: Enabling Language Representation with Knowledge graph; AAAI: New York, NY, USA, 7–12 February 2020; pp. 2901–2908.

29. Camacho-Collados, J.; Pilehvar, M.T. From word to sense embeddings: A survey on vector representations of meaning. J. Artif. Intell. Res. 2018, 63, 743–788.

30. Pennington, J.; Socher, R.; Manning, C. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1532–1543.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).