electronics

MDPI

Article

Completing WordNets with Sememe Knowledge

Shengwen Li [1,2], Bing Li [2], Hong Yao [1,2,*], Shunping Zhou [2], Junjie Zhu [2] and Zhuang Zeng [2]

[1] State Key Laboratory of Biogeology and Enviromental Geology, China University of Geosciences, Wuhan 430079, China; swli@cug.edu.cn
[2] School of Computer Science, China University of Geosciences, Wuhan 430079, China; 20141002431@cug.edu.cn (B.L.); zhoushunping@cug.edu.cn (S.Z.); 1202121172@cug.edu.cn (J.Z.); ZengZhuang@cug.edu.cn (Z.Z.)
* Correspondence: yaohong@cug.edu.cn

Abstract: WordNets organize words into synonymous word sets, and the connections between words present the semantic relationships between them, which have become an indispensable source for natural language processing (NLP) tasks. With the development and evolution of languages, WordNets need to be constantly updated manually. To address the problem of inadequate word semantic knowledge of "new words", this study explores a novel method to automatically update the WordNet knowledge base by incorporating word-embedding techniques with sememe knowledge from HowNet. The model first characterizes the relationships among words and sememes with a graph structure and jointly learns the embedding vectors of words and sememes; finally, it synthesizes word similarities to predict concepts (synonym sets) of new words. To examine the performance of the proposed model, a new dataset connected to sememe knowledge and WordNet is constructed. Experimental results show that the proposed model outperforms the existing baseline models.

Keywords: WordNet; sememe; word embedding; NLP

Citation: Li, S.; Li, B.; Yao, H.; Zhou, S.; Zhu, J.; Zeng, Z. Completing WordNets with Sememe Knowledge. *Electronics* **2022**, *11*, 79. https://doi.org/10.3390/electronics11010079

Academic Editor: Danda B. Rawat

Received: 27 October 2021
Accepted: 22 December 2021
Published: 27 December 2021

1. Introduction

WordNets [1,2] are electronic lexical databases in which most types of words (including nouns, verbs, adjectives, and adverbs) are stored as synonym sets (synsets) and are interconnected by various semantic relations between words. These relations include antonyms, such as 'wet' and 'dry', subordinate relations, such as 'animal' and 'cat', meronymy relations, such as 'toe' is part of 'foot', etc. The Princeton WordNet (PWN), the first English WordNet, now contains over 11,000 synsets in version 3.0. WordNets have become the gold standard lexical knowledge of representing meanings and concepts of words.

Apart from working as lexical databases, WordNets have become an important knowledge source of natural language processing (NLP). They were originally used as standard databases to annotate corpora (e.g., SemCor) and gradually employed to determine the correct meaning of words in many tasks such as word sense disambiguation [3], text clustering [4], word embedding [5], text classification [6], and human brain emotion processing [7]. The successful applications of the English WordNet in these NLP tasks have boosted the construction and refinement of WordNets in many non-English languages. The Global WordNet Association (GWA) was founded to provide a platform for discussing, sharing, and connecting WordNets crossing languages. In addition, some large research projects, such as Turkish KeNet [8], EuroWordNet [9], and MultiWordNet [10], have been built for constructing synonym sets (synsets) with the help of aligning English words to words in non-English languages.

Actually, building WordNet databases involves a large volume of work. Although the manual construction process can ensure that accurate synsets cover as many concepts as possible, it is time-consuming and labor-intensive, and it may require lexicographers to spend years constructing synsets. For example, PWN has taken decades to build, extend,

and refine. Nowadays, WordNet databases are not yet complete and rich enough, especially for non-English languages. In addition, more and more new words are being produced with the continuous evolution of social communication, science, and technology. How to accurately automatically update WordNet is a very valuable and challenging task.

The continuous accumulation of various text and knowledge bases provides new opportunities of automatic constructing and updating WordNet. Three types of data sources have been focused by researchers recently: namely, online text, aligned bilingual corpora, and pre-trained word-embedding semantics from large-scale unlabeled corpora. For example, WEWNE [11] achieves great match accuracy on expanding WordNet with word embedding. Due to the uneven distribution of words in these data, the semantics and relationships of words are not yet fully captured, and there still remain limitations on automatically updating WordNets accurately.

To address this issue, this study updates WordNets by introducing a sememe [12] knowledge database. Sememe is the smallest semantic unit for describing real-world concepts in Chinese and English and can be represented as a vector such as a word in some studies, such as SPSE [13]. This study first organizes the sememe knowledge of words and relationships in WordNet into a unified graph. Then, the word embedding of each word is learned by combining word relationships in WordNet and sememe annotation. Finally, the sysnets (concepts) are predicted based on the semantic similarities of words. The original contributions of this study are as follows: (1) A new dataset connected to HowNet [14] and WordNet is constructed, which can be applied to various NLP tasks. (2) A novel model to automatically complete WordNet is derived, which improves prediction performance by introducing sememe knowledge. (3) This study develops a word-embedding approach that integrates WordNet and sememe knowledge. Benefiting from facilitating both the original semantic relationship structure in WordNet and sememe relationships, the model achieved great improvement on word-embedding vectors.

The remaining sections are organized as follows. Section 2 reviews related works. Section 3 introduces the proposed model framework and describes how the proposed model works. We perform comparative experiments and conduct results analysis in Section 4, and we discuss the effects of hyperparameters, fusion methods, and time costs in Section 5. Finally, Section 6 concludes this study.

2. Related Work

WordNets are mainly built and updated in two ways [15]: the merge-based method and expand-based method. The merge-based method tries to exhaustively enumerate the senses (meanings) of each word in detail and then creates a synset for each given sense by collecting all applicable words that contain that sense. For example, Bhattacharyya [16] describes how a high-quality WordNet can be produced using a merged approach. However, this process is very labor-intensive. The expand-based method employs the existing synonyms in WordNet as a reference, collecting applicable words that represent the meaning of synonyms. Usually, it iteratively identifies high-frequency words and supplements synonyms of these words into WordNet. Compared with merge-based methods, it is expected to build WordNet more quickly by referencing both the set of synonyms and semantic relations in WordNet. However, in practice, it still costs the lexicographer time to construct language-specific synonym sets, partly because some of the meanings or concepts of words do not appear in the initial WordNet, and partly because some specific concepts that only appeared in some languages may have been overlooked.

Recently, the increasing various new data sources provide us new opportunities for automatically completing WordNet. Related research can be classified into three categories according to the types of their data sources, namely bilingual dictionaries, online text, and pre-trained word semantic vectors from a large-scale text corpus.

Bilingual dictionaries are widely used data sources to automatically expand WordNet. Related works implement cross-language WordNet expansions with aligned words in bilingual dictionaries in the source and target languages. The use of bilingual dictionaries

for completing WordNet can be traced back to ten years ago, such as the construction of WordNet for Catalan and Spanish as part of the EuroWordNet project [17]. Strategies used to extend WordNet in those studies include identifying semantic relationships in various target language resources [18], calculating mutual information of English translated texts in the source and target language corpora [19], and constructing English grammar sets based on word sense disambiguation algorithm [20]. In addition, bilingual dictionaries can be combined with popular NLP techniques to better identify the relationships between words. For example, recent work by Lam et al. [21] has focused on the use of machine translation and (or in some cases instead of) bilingual dictionaries to construct WordNet for various languages (Arabic, Assamese, Sa, Spanish, and Vietnamese).

For online text, researchers have studied to supply synonym sets of PWNs by automatically extracting semantic relations from online encyclopedias (i.e., Wikipedia) [22,23]. With the help of available parallel text corpora, the set of synonyms of one language can be obtained from the WordNet of another language. For example, Oliver [24] first tagged the target language with simple part-of-speech (POS) tags (nouns, verbs, adjectives, and adverbs) by using Freeling and the lexical disambiguation toolkit and then inferring a subset of the English synonym sets to synonym sets of non-English languages.

Pre-trained word semantic vectors (word-embedding vectors), which are emerging recently, can be employed to predict the relationships between words. Usually, word semantic vectors are learned from large-scale text corpora, which have achieved great performance for lots of NLP tasks. Rothe [25,26] reported that embedding models can capture various relations related to WordNet-style word meanings, thus mapping word pairs to WordNet synsets. Word-embedding vectors have been successfully applied in building Norwegian and Arabic WordNet [11,27]. Similar work was conducted by Khodak [28], who used a bilingual dictionary and word embeddings to automatically construct the entire French and Russian WordNet.

In summary, the method of completing WordNet based on automatic expansion is promising because it is very simple and effective. Benefiting from advances in representation (embedding) learning, word-embedding-based methods and sememe embedding methods [13] are developing rapidly, with advantages such as semantic richness and free from manual labor. However, WordNet is a very precise lexical database. It is not insufficient to complement WordNet only relying on word-embedding vectors, especially when some low-frequency words cannot be well represented by those widely used word-embedding models.

3. Model

In this section, we describe the proposed model, completing WordNet with sememe knowledge (CWS), which incorporates sememe knowledge in HowNet [29] for completing WordNet.

The proposed model consists of three components. The first component attempts to use graph structures to characterize the relationships among words and sememes. The second component learns the representation vectors of the word and sememes based on the relationships presented in the graph. The third component predicts synsets of "new words" with two types of representation vectors. The following three subsections detail the three components.

3.1. Building WordNet–Sememe Graph

As a commonsense knowledge database, HowNet presents word semantics in a very concise and clear way, in which each word is labeled with sememes, a minimal semantic unit, that are highly relevant to the semantics of each word. Specifically, it defines about 2000 individual sememes in English and Chinese and explains all the words with these sememes. The simple sememe annotations of words should be effectively linked into WordNet to provide support for WordNet completion.

WordNet and the sememe knowledge database (HowNet) are two different types of knowledge databases. In order to predict the relationships of a new word in WordNet while leveraging sememe, the semantics of the word from WordNet and the semantics from sememe need to be presented effectively firstly. In this section, we will use a unified graph structure, the WordNet–sememe graph, to represent the two semantics among words and sememes.

The constructing of the WordNet–sememe graph is the process of determining nodes and constructing edges between nodes. Algorithm 1 illustrates this process in detail. Firstly, all the words and sememes are selected as nodes of the graph. Secondly, the edges between words are added. Considering that subordination and synonymy relationships are usually the most important relationships, this paper will construct the edges between words based upon the two kinds of relationships. Specifically, if any one of these two relations exists between a pair of words, a word–word edge is added between the pair of words. Finally, each word will be connected with its sememes by adding word–sememe edges.

As shown in Figure 1a, the definition of the word "shirt" in HowNet is related to five sememes: "clothing", "put on", "parts", "body", and "human". In the semantic structure graph, this model will append edges between the sememe nodes ("clothing", "put on", "part", "body", and "human") and the word node "shirt", which means that the word "shirt" is interpreted by these nodes, as shown in Figure 1c. The words "skirt" and "clothes" are also interpreted by the node "clothing" in HowNet, so the sememe node "clothing" and the word nodes "dresses" and "clothes" will also have a connected edge.

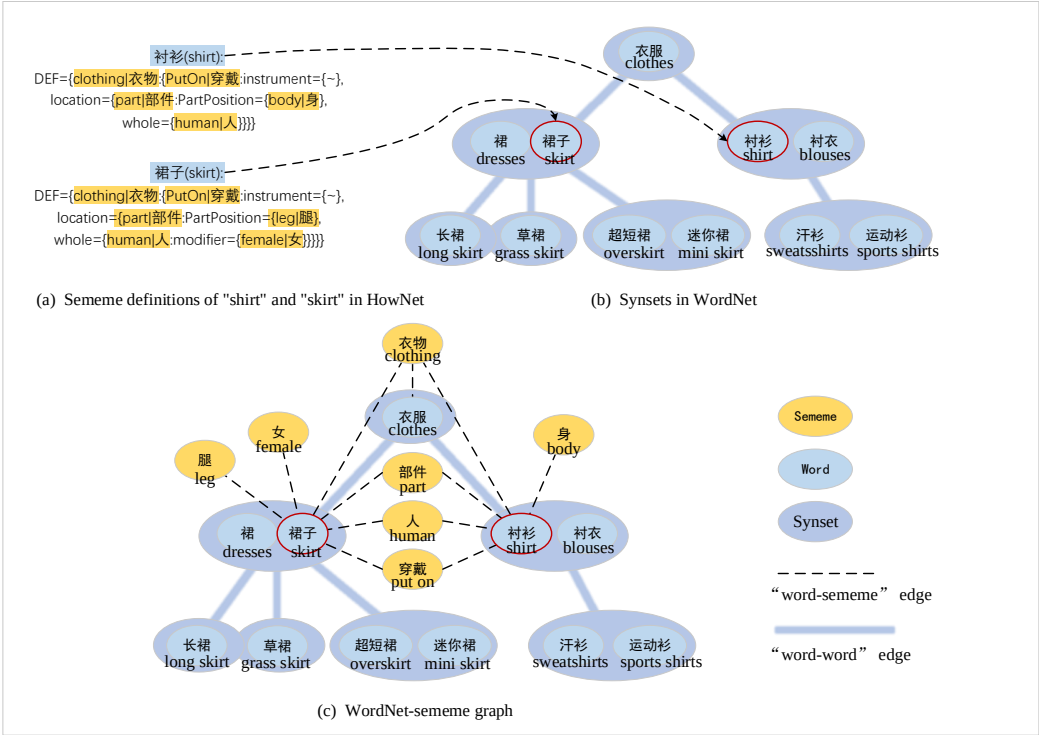

Figure 1. The construction process of WordNet–sememe graph.

Algorithm 1 WordNet-sememe graph construction

Input: Word set of WordNet W_{WN}; word–sememe pairs in HowNet $H = \{[w, S_w]|w \in W_{HN}\}$, W_{HN} is the word set of HowNet, S_w is the sememe set of word w
Output: WordNet–sememe graph $G = (V, E)$

```
1: V=W_WN
2: E =
3: for [w_1, w_2] ∈ W_WN × W_WN do
4:     if relation_type(w_1, w_2) ∈ {subordination, synonymy} then
5:         E.add(w_1, w_2)
6:     end if
7: end for
8: for [w, S_w] ∈ H do
9:     if w ∈ W_WN then
10:        V.add(S_w)
11:        for s ∈ S_W do
12:            E.add(w, s)
13:        end for
14:    end if
15: end for
16: return G = (V, E)
```

3.2. Jointly Learning the Embedding Vectors of Word and Sememe

In this study, word embedding was used to identify the connections between the word embedding semantic space and the WordNet semantic space. High-quality word embeddings will be beneficial to improve the performance of the model. In most previous research, the word semantics from WordNet and word embeddings from text corpus usually are learned independently. The differences between the two spaces raise the uncertainty of word relationships. In addition, the uneven frequency of words in the corpus and the complexity of the contextual environment also lead to the deviation of word embeddings on identifying word semantic and word relationships. This paper tried to improve word embedding vectors by fusing the WordNet structure and sememe knowledge. Specifically, sememe co-occurrence relations in the built WordNet–sememe graph in Section 3.1 are employed for facilitating obtaining the vector representation of words.

In this study, an extended Skip-gram model is derived to learn the embedding vectors of words and sememes. In the Skip-gram model, the embedding vector of a word assumes that it is correlated with the embedding vector of the words that occur in its context [30,31]. The embedding vectors of words can be learned by maximizing the prediction probability of words in their context.

Since original WordNet and sememe knowledge are not organized as text sequences, the classical sequence sliding windows approach does not work in this study. Instead, in this study, the context (neighbor nodes) of a word will be obtained by traversing the built graph in Section 3.1. Algorithm 2 describes the process of building similar word pairs in WordNet with hierarchical traversal. Line 1 and Line 2 are used to initialize the set of similar words and the queue of hierarchical word traversal, respectively. Lines 3–16 define the whole procedure of hierarchical traversal, where lines 5–8 implement the hierarchical traversal of subordinate relations; lines 9–12 implement the traversal of similar relations, and lines 13–15 terminate the traversal when the size of the set of similar words is large enough. Finally, line 17 selects the first N words from the set of similar words to build the final training samples.

Algorithm 2 Get context words from the built WordNet–sememe graph

Input: WordNet–sememe graph, word w
Output: N words that are surrounding to word w

List S=[]
Queue Q=[w]
while queue Q is not empty **do**
    q=the head element of queue Q dequeues
    for w$'$ $\in$ hyperyms and hyponyms of the word q **do**
        S.append(w')
        Q.append(w')
    end for
    for $w'\in$ words that are connected to the word q **do**
        S.append(w')
        Q.append(w')
    end for
    if the size of the list $S \geq N$ **then**
        break
    end if
    Q.pop()
end while
return the first N words of the list S

For word w, this paper presents the surrounding nodes (context words) obtained in the algorithm as S_w and will jointly train the embedding vectors of words and sememes, as shown in Figure 2. The loss function of the extended model is derived from the loss function of Skip-gram, that is, the conditional probability of nodes around the target word w_i is calculated by Equation (1):

$$L_{wn}(S_w) = \sum_{i=M}^{n-M} \log P_{wn}(w_{i-M}, \cdots, w_{i+M}|w_i), \tag{1}$$

where $P_{wn}(w_{i-M}, \cdots, w_{i+M}|w_i)$ denotes the conditional probability of predicting surrounding the target word w_i; M is an hyperparameter. The conditional probability P_{wn} can be calculated by the Softmax function as follows,

$$P_{wn}(w_{i-M}, \cdots, w_{i+M}|w_i) = \prod_{w_j \in S_W} P_{wn}(w_j, w_i) = \prod_{w_j \in S_W} \frac{e^{(\hat{v}_j{}^T \cdot \hat{v}_i)}}{\sum_{w_k \in s_{wj}} e^{(\hat{v}_j{}^T \cdot \hat{v}_k)}}, \tag{2}$$

where $\hat{v}_i$ and $\hat{v}_j$ denote the embedding vectors of the target word, w_i, and its surrounding word nodes, w_j. The initial values of v_i is set with the word vector of w_i from pre-training word vectors in large corpus. They are fused with word vectors, v, and sememe vectors, $\tilde{v}$. Specifically, they can be calculated by Equation (3),

$$\hat{v} = \alpha \cdot v + (1 - \alpha)\tilde{v}, \tag{3}$$

where the combined sememe vectors can be calculated by Equation (4),

$$\tilde{v} = \sum \frac{b_i}{||b||_2}, \tag{4}$$

denotes the embedding vectors of the sememe sets of a word in HowNet.

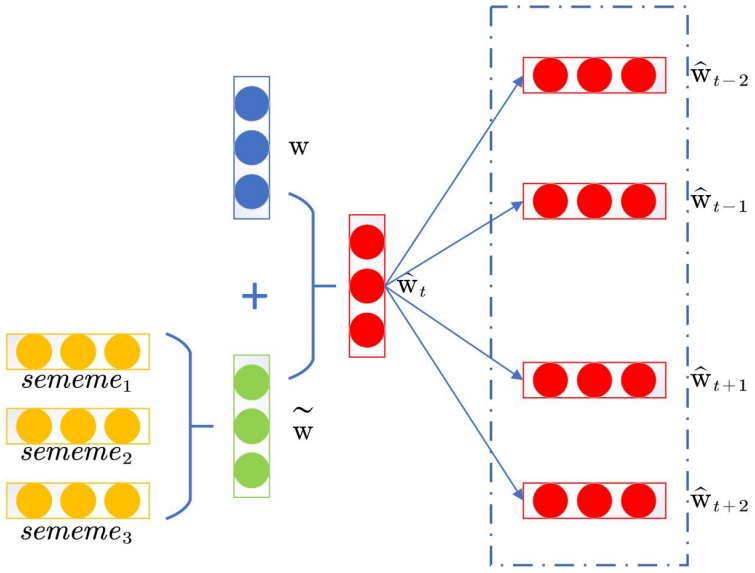

Figure 2. Joint sememe and word-embedding model.

For training, the built WordNet–sememe graph is fed into the derived Skip-gram model. The model parameters, i.e., the vectors of words and sememes, are continuously adjusted during the model training process. After the model has been trained over a number of iterations, the embedding vectors of the words and the sememes are obtained.

3.3. Synthesizing Word Similarity for Synsets Prediction

For a given new word w that does not appear in WordNet, this paper determines its position (predicts the synsets of a new word) in WordNet with the help of the words that have the same/similar semantics. This idea mainly assumes that words with similar semantics have adjacent or even the same position in WordNet. This model is partially derived from collaborative filtering algorithms that are widely used in personalized recommendation tasks [32]. Here, words are analogized to users of collaborative filtering algorithms, and candidate synonym sets are analogized to recommended items. Differing from classical collaborative filtering models in personalized recommendation applications, our model only takes the most similar words as the references of words for word prediction. This is because that irrelevant words in WordNet may be far away, and there should be noise when predicting the position of the "new word".

This study adds two declined confidence factors to the recommendation score function, and the derived score function is shown as follows:

$$score(w,\hat{w}_j) = \sum_{w_j \in S_w} d^{H(\hat{w},w'_j)-1} \cdot sim(v,v_j) \cdot c^{r_j}, \tag{5}$$

$d^{H(\hat{w},w'_j)-1}$ is the declined confidence factor from subordinates; d is a hyperparameter and $d \in (0,1)$; $H(w,w_j)$ denotes the path length between the candidate word and the word w in the WordNet hierarchy. The latter is a semantic distance function, where r indicates the distance between neighboring nodes and the superordinate word in the WordNet hierarchy. That is, the candidate superordinate word can be the father node of the neighboring nodes and the grandfather node. c^{r_j} is the declined confidence factor from the neighbor word w_j, r_j is the rank of word similarity $sim(w,w_j)$, c is a hyperparameter, and $c \in (0,1)$. Finally, the candidate superordinate word with the highest score will be selected and recommended to word w.

4. Experiments and Results

This section introduces the built experimental dataset, experimental settings, and experimental results.

4.1. Dataset

The NTU Computational Linguistics Lab has constructed the Chinese Open WordNet dataset [33] following Princeton WordNet and Global WordNet. It contains 42,315 synsets, 79,812 senses, and 61,536 unique words. The distribution of words and synsets in the WordNet dataset is shown in Table 1.

Table 1. An overview of the words and synsets of nouns, verbs, and adjectives in Chinese Open WordNet.

POS	Words	Synsets
Noun	38,975	27,888
Verb	8412	5157
Adj	14,199	8559
Total	61,586	41,604

In the experiments, word embeddings of all words are learned with a large corpus, the Sogou-T dataset [34]. The dataset is a widely used text corpus developed by Sogou and its partners, which consists of a variety of original web pages from the Internet with a total word count of approximately 2.7 billion.

In order to examine the performance of the proposed model, we collected the aligned words between the Chinese Open WordNet and HowNet knowledge databases and constructed the experimental dataset, in which 10% of the words were randomly extracted as the test set, 10% of the words were randomly extracted as the validation set, and the remaining 80% were randomly extracted as the training set. An overview of the built dataset is listed in Table 2.

Table 2. Overview of the built experimental dataset.

# Words	# Subordinations	# Synonyms	# Train Set	# Validation Set	# Test Set
18,123	44,187	8563	14,499	1812	1812

As shown in the table, a total of 18,123 words were extracted from the Chinese Open Word Network as the dataset, of which 14,499 words, 1812 words, and 1812 words were randomly divided into three sets: the training set, validation set, and test set, respectively. A total of 44,187 subordination relations and 8563 synonym relations were collected in the training set.

The sememe dataset is constructed by extracting 18,123 words aligned with the Chinese Open WordNet from HowNet. This extracted sememe dataset is employed to facilitate the concept prediction of "new words" for completing Chinese Open WordNet. An overview of the dataset is listed in Table 3.

Table 3. The overview of the extracted sememe dataset.

# Word	# Sememe	# Sememe of Each Word
18,123	1893	2.7

As shown in Table 3, the extracted sememe dataset removed some of the less frequently occurring sememes. After that, the dataset contains 1893 sememes and 18,123 words. For a word in the dataset, an average of 2.7 sememes were used to explain the word.

The distribution of sememes in the dataset is shown in Figure 3, in which the horizontal coordinates indicate the number of sememe annotations of a word in HowNet, and the

vertical coordinates indicate the frequency of the word. According to the figure, most of the words were interpreted by 1–5 sememes, and very few words were interpreted by more than 10 sememes.

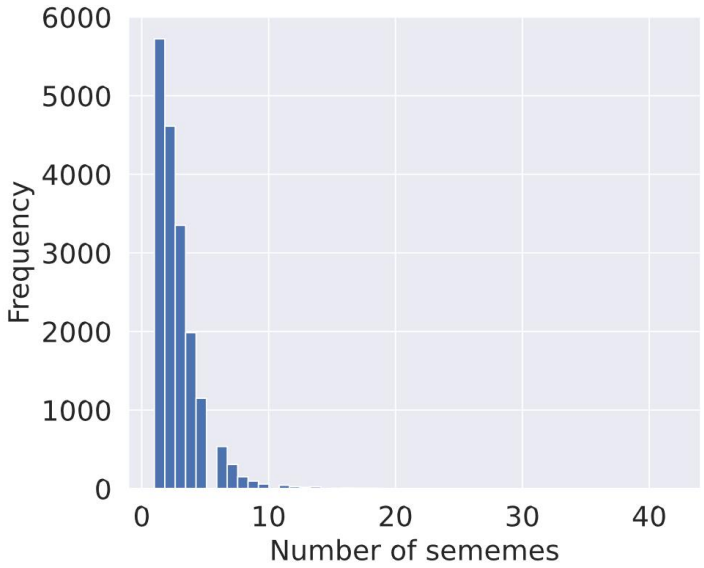

Figure 3. Distribution of the number of annotated sememe per word in HowNet.

4.2. Experimental Settings

In the experiments, the initialized embedding vectors of words are by Glove [35] learned from the Sogou-T corpus. Glove constructs a co-occurrence matrix from the corpus and obtains the word vector presentation by a matrix factorization approach.

In this study, M is set to 2, and d is set to 0.8; $sim(\cdot)$ denotes the similarity between two words. Here, this paper uses the dot product of word embeddings of two words, $sim(a, b) = a \cdot b$, to calculate the semantic similarity of two words in the vector space. The larger the sim value of two words, the more similar the two words are in their semantic space. The hyperparameter c is set to 0.8, where r_i is the rank order of word i according to the word similarity.

For each "new word", this paper selects the neighboring 100 words from the vector space as the current word, which means that the hyperparameter, N, is set to 100, and α is set to 0.3, which means that when fusing sememe vectors, 30% of the word-embedding vectors are transferred.

For training model, the batch update strategy is used to train model parameters. The batch size is set to 1024, the dimensional size of both the word and the sememe vectors is set to 200, the learn rate is set to 0.0001, and 20 negative samples are selected. The model was trained for 500 epochs.

4.3. Evaluation Metrics

To predict concepts (synsets) of a new word, the model calculates the score of each candidate synset with the score function in Equation (5) and finally ranks all the synsets to select top synsets as predicted results of the "new word". We borrow the widely used evaluation metric, Hit@K, from the popular knowledge representation task as the evaluation metric of this study. Hit@K denotes the proportion of the correct synsets in the top K lists. The higher the value of Hit@K, the better the model.

In the experiments, this paper examines the impact of sememe vectors that are obtained based on WEWNE and SPSE. WEWNE is a classical model that is used to expand WordNet with word embedding. SPSE obtains embedding vectors of sememe based on a matrix decomposition approach, while the proposed model uses a jointly trained method to obtain embedding vectors of sememes and words.

4.4. Results

Two variants of our proposed model, CWS-sememe and CWS, are experimented with in this section. The CWS-sememe denotes a variant that removed sememe knowledge. That is, two components of Sections 3.1 and 3.2 are ignored in the variant. Their experimental results are listed in Table 4.

Table 4. Performance comparison of different models; the best results are shown in boldface.

Model	Hit@5	Hit@10	Hit@50	Hit@100
SPSE	0.189	0.234	0.377	0.438
WEWNE	0.195	0.243	0.383	0.445
CWS-sememe	0.201	0.245	0.385	0.443
CWS	**0.244**	**0.311**	**0.441**	**0.488**

Table 4 shows that the prediction accuracy of the proposed model, CWS, is better compared with SPSE. It indicates that the jointly learning method proposed in this paper is effective. The performance of the sememe vectors obtained based on matrix decomposition alone decreased on the WordNet synset prediction task when they were fused into the word embeddings. The reason may be caused by the fact that the training of sememe vectors in SPSE is detached from the semantic space of WordNet; thus, word vectors can not fuse with sememe vectors very well, which diminishes the prediction ability of the model.

Furthermore, CWS-sememe has made performance improvements compared with the baseline model, WEWNE. This can be explained by the fact that WEWNE only uses the similarity between words more simply and directly, while CWS-sememe takes into account the addition of similarity ranking to the model.

CWS significantly outperforms the CWS-sememe model because the CWS model considers the text-based training word embedding along with the sememe annotation information in HowNet. The traditional text-based word embedding can only consider the word contribution relationship in the corpus, and the semantic space from WordNet and from the corpus are two semantic spaces that are independent of each other, so the word embedding obtained from the text alone cannot accurately capture the similarity relationship in WordNet. HowNet is an artificially defined semantic commonsense knowledge base, which is more in line with the semantic knowledge base compared to the classical word embedding-based methods. The CWS model adds the sememe knowledge in HowNet to the learning of word embeddings by finding the neighbor nodes in WordNet, so that the word embeddings after fusing sememe embeddings can match the semantic space of WordNet more closely; thus, it obtains better performance.

5. Discussion

5.1. Effect of Hypermeters

5.1.1. Effect of α

The training of sememe vectors in this study fuses the sememe knowledge and semantic relationships of WordNet, which contribute to its better prediction performance. To fuse the word-embedding vectors and sememe-embedding vectors, α is employed as a hyper-parameter in Equation (3) to balance the weight of two types of vectors. Consequently, the value of α may affect the prediction accuracy of the proposed model. To examine the effect from α values, we perform experiments by setting α values from 0.1 to 0.9 and calculated their prediction accuracies on the test set. Their results are listed in Table 5.

As shown in Table 5, the proposed model performs great when α is around 0.3. Specifically, the model obtained the highest value of Hit@5 when α is set to 0.2, and the model has the highest value of Hit@100 when α is set to 0.4. When α is set to 0.3, the model obtained the highest predicted Hit@10 and Hit@50, and the values of Hit@5 and Hit@100 are very high values. Consequently, α is set to 0.3 in the other experiments of this paper.

Table 5. Prediction accuracy of CWS model with different α settings; the best results are shown in boldface.

α	Hit@5	Hit@10	Hit@50	Hit@100
0.1	0.151	0.189	0.267	0.297
0.2	**0.246**	0.309	0.435	0.478
0.3	0.244	**0.311**	**0.441**	0.488
0.4	0.232	0.292	0.429	**0.489**
0.5	0.218	0.278	0.420	0.478
0.6	0.200	0.261	0.415	0.469
0.7	0.190	0.250	0.402	0.460
0.8	0.187	0.247	0.391	0.455
0.9	0.184	0.245	0.383	0.450

When the weight of the initial word embedding is low, such as that α is set to 0.1, the prediction accuracy of the model is significantly lower than that when α is set to 0.3, which suggests that sememe knowledge provides a critical semantic link for predicting synsets of "new words". The prediction accuracy decreases when α is larger than 0.3, which indicates that the over-reliance on the sememe annotations in HowNet prevents the model from focusing on the basic semantic associations between words. In summary, both lower and higher α may reduce the prediction accuracy of the model.

5.1.2. Effect of *c*

The hyperparameter *c* serves to adjust the weight of different words for the synset of "new word", which should affect the prediction accuracy of the proposed model. To examine the impact of c, we experiment with different *c* values and list their results in Table 6.

Table 6. Prediction accuracies of the CWS model with different values of c; the best results are shown in boldface.

c	Hit@5	Hit@10	Hit@50	Hit@100
0.1	0.235	0.297	0.437	0.485
0.2	0.238	0.296	0.437	0.486
0.3	0.237	0.298	0.437	0.485
0.4	0.239	0.298	0.437	0.486
0.5	0.242	0.300	0.438	0.487
0.6	0.241	0.299	0.438	0.486
0.7	0.240	0.303	0.44	0.486
0.8	**0.244**	**0.311**	**0.441**	**0.488**
0.9	0.238	0.309	0.44	0.486

As shown in Table 6, the experimental results suggest that the values of *c* have a marginal effect on model performance. As the value of *c* gradually increases, the prediction accuracy of the model first gradually increases and then decreases. When *c* is set to 0.8, the model achieves the best prediction accuracy.

5.1.3. Effect of *N*

The parameter *N* is used to select the number of similar words in Algorithm 1. As a hyperparameter, the values of *N* may affect the prediction accuracy of the proposed model. To examine the effect of *N*, we set *N* from 10 to 100 to evaluate the prediction accuracies of the model on the validation set. The experimental results are listed in Table 7.

As shown in Table 7, the model can obtain excellent prediction results even if *N* is set to 10. It suggests that the most similar words almost cover the semantic meanings of a word and provide the most semantic context for predicting the synsets of "new word".

As the value of N increases, the prediction accuracy of the model shows light improvement, which can be explained as that more words can provide richer semantic information. Consequently, N is set to 100 for experiments in other sections of this paper.

Table 7. Prediction accuracies when M is set to different values; the best results are shown in boldface.

M	Hit@5	Hit@10	Hit@50	Hit@100
10	0.243	0.307	0.401	0.404
20	0.243	0.308	0.436	0.453
30	0.244	0.309	0.440	0.479
40	0.243	0.309	0.440	0.484
50	0.242	0.309	0.440	0.486
60	0.242	0.309	0.440	0.486
70	0.244	0.309	0.440	0.486
80	0.244	0.309	0.439	0.485
90	0.244	0.309	0.440	0.485
100	0.244	**0.311**	**0.441**	**0.488**

5.2. Effect of Sememes Aggregation Methods

In Section 3.2, a normalization(norm) operation can be employed to calculate the sememe-combining vectors. In addition to the normalization operation, sum and mean operations can be used to fuse semantic vectors, as shown in Equation (6),

$$\tilde{v} = \begin{cases} \sum b_i & sum \quad of \quad sememe \quad vector \\ \frac{\sum b_i}{|b|} & mean \quad of \quad sememe \quad vector \end{cases}. \tag{6}$$

To examine the effect of different sememe aggregation methods, including the mean, norm, and sum, we conducted experiments with these methods and plotted their loss curves in their training processes into Figure 4.

As shown in the figure, all three aggregation methods can help the proposed model converge to the optimized result within dozens of epochs. Among them, the normalization-based (norm) aggregation approach can obtain its optimized parameters within 20 epochs of training. Yet, the mean and sum aggregation methods need more epochs.

By comparing the three sememe fusion approaches in the CWS model, it can be concluded that the normalized fusion approach achieves the best results, which may be explained by it reducing the noise caused by the varying number of semantic annotations for different words.

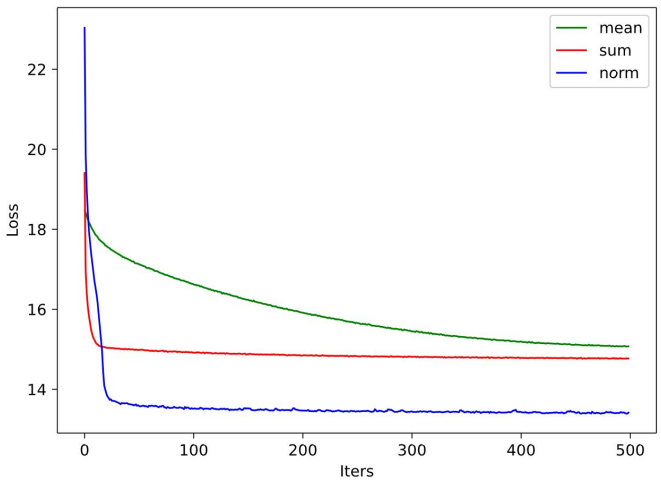

Figure 4. Loss curves of the three sememe aggregation methods during the training process.

5.3. Time Cost Analysis

In the experiments, this paper also compares the time costs of several models on predict "synset" tasks, as shown in Table 8.

Table 8. The time cost of different models.

Model	Training Time Cost (Seconds)	Predicting Time Cost (Seconds)	Total Time Cost (Seconds)
WEWNE	——	52	52
CWS-sememe	——	54	54
CWS	277	2141	2418

As shown in the table, the training time required by the WEWNE model and the CWS-sememe model is very close. They can both predict synsets in tens of seconds. Yet, CWS takes a little more time to learn model parameters, because the vector information of sememe needs to be trained during the vector calculation of "new words" and neighboring words, which consumes most of the time of the training process. Although CWS takes more time, it is still within an acceptable time range. With this in mind, the better prediction accuracy of the model is more valuable than saving a little extra time.

6. Conclusions and Future Work

To complete WordNet accurately, this paper jointly learns the embedding words in WordNet and sememe, enhancing the ability to automatically predict synsets (concept) of new words. The experimental results show that the proposed method of inferring word relations by fusing the original knowledge of meaning is promising, which improves the task of concept prediction. The research in this paper not only provides a new method reference for the constructing and updating of WordNet but also makes a new exploration for enhancing the semantic representations of words.

The main limitation of this study is that only two languages, English and Chinese, have sememe-based linguistic knowledge bases. How to update WordNet in more languages by introducing the cross-language sememe prediction methods needs to be investigated. In addition, the classical Skip-gram model is used in this paper to jointly train the vector representation of sememes and words. The performance of updating Wordnet is expected to be improved if newer advanced methods are employed to replace the Skip-gram model. Furthermore, the types of relationships between words in WordNet may contribute to improving the performance of the model.

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

Funding: This work was supported in part by the National Natural Science Foundation of China (NSFC) (No. 61972365, 42071382) and in part by the CCF-NSFOCUS Kun-Peng Scientific Research Fund (CCF-NSFOCUS 2021002), and in part by the Natural Science Foundation of Hubei Province, China (No. 2020CFB752), and in part by the Open Research Project of the Hubei Key Laboratory of Intelligent Geo-Information Processing (No. KLIGIP-2021B01, KLIGIP-2018B02).

Data Availability Statement: Data of this research are available from the authors.

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

References

1. Kilgarriff, A. *Wordnet: An Electronic Lexical Database*; MIT Press: Cambridge, UK, 2000.
2. Beckwith, R.; Fellbaum, C.; Gross, D.; Miller, G.A. WordNet: A lexical database organized on psycholinguistic principles. In *Lexical Acquisition: Exploiting on-Line Resources to Build a Lexicon*; Psychology Press: London, UK, 2021; pp. 211–232.

3. Hasan, A.M.; Noor, N.M.; Rassem, T.H.; Noah, S.A.M.; Hasan, A.M. A proposed method using the semantic similarity of WordNet 3.1 to handle the ambiguity to apply in social media text. In *Information Science and Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 471–483.
4. Marcińczuk, M.; Gniewkowski, M.; Walkowiak, T.; Będkowski, M. Text document clustering: Wordnet vs. TF-IDF vs. word embeddings. In Proceedings of the 11th Global Wordnet Conference, Pretoria, South Africa, 18 January 2021; pp. 207–214.
5. Lee, Y.Y.; Ke, H.; Yen, T.Y.; Huang, H.H.; Chen, H.H. Combining and learning word embedding with WordNet for semantic relatedness and similarity measurement. *J. Assoc. Inf. Sci. Technol.* **2020**, *71*, 657–670. [CrossRef]
6. Özçelik, M.; Arıcan, B.N.; Bakay, Ö.; Sarmış, E.; Ergelen, Ö.; Bayezit, N.G.; Yıldız, O.T. HisNet: A Polarity Lexicon based on WordNet for Emotion Analysis. In Proceedings of the 11th Global Wordnet Conference, Pretoria, South Africa, 18 January 2021; pp. 157–165.
7. Kocoń, J.; Maziarz, M. Mapping WordNet onto human brain connectome in emotion processing and semantic similarity recognition. *Inf. Process. Manag.* **2021**, *58*, 102530. [CrossRef]
8. Bakay, Ö.; Ergelen, Ö.; Sarmış, E.; Yıldırım, S.; Arıcan, B.N.; Kocabalcıoğlu, A.; Özçelik, M.; Sanıyar, E.; Kuyrukçu, O.; Avar, B.; et al. Turkish wordnet kenet. In Proceedings of the 11th Global Wordnet Conference, Pretoria, South Africa, 18 January 2021; pp. 166–174.
9. Vossen, P. Eurowordnet: A multilingual database of autonomous and language-specific wordnets connected via an inter-lingualindex. *Int. J. Lexicogr.* **2004**, *17*, 161–173. [CrossRef]
10. Pianta, E.; Bentivogli, L.; Girardi, C. MultiWordNet: Developing an aligned multilingual database. In Proceedings of the First International Conference on Global WordNet, Mysore, India, 21–25 January 2002; pp. 293–302.
11. Sand, H.; Velldal, E.; Øvrelid, L. Wordnet extension via word embeddings: Experiments on the Norwegian Wordnet. In Proceedings of the 21st Nordic Conference on Computational Linguistics, Gothenburg, Sweden, 22–24 May 2017; pp. 298–302.
12. Yang, L.; Kong, C.; Chen, Y.; Liu, Y.; Fan, Q.; Yang, E. Incorporating sememes into chinese definition modeling. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2020**, *28*, 1669–1677. [CrossRef]
13. Xie, R.; Yuan, X.; Liu, Z.; Sun, M. Lexical Sememe Prediction via Word Embeddings and Matrix Factorization. In Proceedings of the IJCAI, Melbourne, Australia, 19–25 August 2017; pp. 4200–4206.
14. Dong, Z.; Dong, Q. HowNet-a hybrid language and knowledge resource. In Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering, Beijing, China, 26–29 October 2003; pp. 820–824.
15. Neale, S. A survey on automatically-constructed wordnets and their evaluation: Lexical and Word Embedding-based Approaches. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation, Miyazaki, Japan, 7–12 May 2018; pp. 1705–1710.
16. Bhattacharyya, P. IndoWordNet. In Proceedings of the Seventh International Conference on Language Resources and Evaluation, Valletta, Malta, 19–21 May 2010; pp. 3785–3792
17. Farreres, X.; Rigau, G.; Rodriguez, H. Using wordnet for building wordnets. In Proceedings of COLING-ACL Workshop "Usage of WordNet in Natural Language Processing Systems", Montreal, QC, Canada, 16 August 1998; pp. 65–72
18. Barbu, E.; Mititelu, V.B. Automatic building of Wordnets. In Proceedings of the International Conference on Recent Advances in Natural Language Processing, Borovets, Bulgaria, 21–23 September 2005; pp. 99–106.
19. Montazery, M.; Faili, H. Automatic Persian wordnet construction. In Proceedings of the Coling, Beijing, China, 23–27 August 2010; pp. 846–850.
20. Taghizadeh, N.; Faili, H. Automatic wordnet development for low-resource languages using cross-lingual WSD. *J. Artif. Intell. Res.* **2016**, *56*, 61–87. [CrossRef]
21. Lam, K.N.; Al Tarouti, F.; Kalita, J. Automatically constructing Wordnet synsets. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 22–27 June 2014; pp. 106–111.
22. Oliver, A. Aligning Wikipedia with WordNet: A Review and Evaluation of Different Techniques. In Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, 11–16 May 2020; pp. 4851–4858.
23. Ruiz-Casado, M.; Alfonseca, E.; Castells, P. Automatic extraction of semantic relationships for wordnet by means of pattern learning from wikipedia. In Proceedings of the International Conference on Application of Natural Language to Information Systems, Alicante, Spain, 15–17 June 2005; pp. 67–79.
24. Oliver, A.; Climent, S. Automatic creation of WordNets from parallel corpora. In Proceedings of the LREC, Reykjavik, Iceland, 26–31 May 2014; pp. 1112–1116.
25. Panchenko, A. Best of both worlds: Making word sense embeddings interpretable. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, 23–28 May 2016; pp. 2649–2655.
26. Rothe, S.; Schütze, H. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. *arXiv* **2015**, arXiv:1507.01127.
27. Al Tarouti, F.; Kalita, J. Enhancing automatic wordnet construction using word embeddings. In Proceedings of the Workshop on Multilingual and Cross-lingual Methods in NLP, San Diego, CA, USA, 17 June 2016; pp. 30–34.
28. Khodak, M.; Risteski, A.; Fellbaum, C.; Arora, S. Automated WordNet construction using word embeddings. In Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications, Valencia, Spain, 4 April 2017; pp. 12–23.
29. Dong, Z.; Dong, Q. Hownet and the Computation of Meaning. In Proceedings of the 23rd International Conference on Computational Linguistics, Beijing, China, 23–27 August 2010; pp. 12–23.

30. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:1301.3781.
31. Shu, X.; Yu, B.; Zhang, Z.; Liu, T. DRG2vec: Learning Word Representations from Definition Relational Graph. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, Glasgow, UK, 19–24 July 2020; pp. 1–9.
32. Yu, X.; Jiang, F.; Du, J.; Gong, D. A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains. *Pattern Recognit.* **2019**, *94*, 96–109. [CrossRef]
33. Wang, S.; Bond, F. Building the chinese open wordnet (cow): Starting from core synsets. In Proceedings of the 11th Workshop on Asian Language Resources, Nagoya, Japan, 14–18 October 2013, pp. 10–18.
34. 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, Canada, 30 July–4 August 2017; pp. 2049–2058.
35. Pennington, J.; Socher, R.; Manning, C.D. 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.