Research Article

Attention Neural Network for Biomedical Word Sense Disambiguation

Chun-Xiang Zhang,1 Shu-Yang Pang,1 Xue-Yao Gao,1 Jia-Qi Lu,2 and Bo Yu1

1School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China
2College of Arts and Sciences, Northeast Agricultural University, Harbin 150030, China

Correspondence should be addressed to Xue-Yao Gao; xueyao_gao@163.com

Received 27 October 2021; Accepted 18 December 2021; Published 10 January 2022

1.Introduction

Biomedical texts are now so large that automation tools are needed to process them effectively. But, it is difficult to process biomedical texts automatically. The reason is that there are more ambiguous biomedical words in the biomedical field. It is helpful for the automatic processing of biomedical articles to determine semantic categories of biomedical words. Now, biomedical word sense disambiguation (WSD) is widely applied to biomedical natural language processing tasks, such as text indexing, text categorization, and named entity extraction.

In the field of biomedicine, there is polysemy in professional vocabulary. For example, the biomedical word “ADA” has two semantics, including “American Dental Association” and “Adenosine Deaminase.” We need to determine the correct meanings of biomedical words according to relevant information of contexts.

Biomedical WSD methods can be divided into 3 categories: supervised methods, unsupervised ones, and knowledge-based ones. In supervised methods, a labeled dataset along with lexical and syntactic information in context is used to train a classifier that predicts correct senses of biomedical words in the test dataset. In unsupervised ones, unlabeled biomedical texts are used to provide sense choices for biomedical words. In knowledge-based ones, thesauri and sense inventories are applied to determine semantic categories of biomedical words. WordNet and Unified Medical Language System (UMLS) [1] are two important thesauri, which provide brief definitions of different senses and corresponding synonyms. Knowledge-based approaches do not use any corpus but solely rely on thesauri or sense inventories such as WordNet and UMLS that contain brief definitions of different senses and corresponding synonyms.

This paper intends to combine neural networks and linguistic knowledge for improving the performance of biomedical WSD. There is a lot of linguistic information in contexts around the biomedical word, which can be used to determine its semantics. But, some information is helpful,
and others are noisy. Neural networks are often used to extract discriminative information. Each neural network has its advantages and disadvantages. It is a challenge for the biomedical WSD system to combine multiple neural networks for extracting effective discriminative information from contexts. This paper combines Av-ACNN and Bi-LSTM to extract discriminative features from contexts around the biomedical word and determine its semantics, which improves the disambiguation accuracy of the biomedical WSD system.

In this paper, we take morphology, part of speech, and semantic information of four adjacent lexical units around the biomedical acronym as disambiguation features. Word embeddings [2–4] are used as representations for biomedical WSD problems. Discriminative information embedded in word units is extracted by an attention mechanism to obtain features at a high level. Based on these features, average asymmetric convolutional neural networks (Av-ACNN), bidirectional long short-term memory (Bi-LSTM) networks, and softmax function are used to determine the semantic category of the biomedical acronym. The main contributions of this paper are summarized as follows:

1. Morphology, part of speech, and semantic information of 4 adjacent lexical units around the biomedical acronym are used as disambiguation features. We use word embedding to generate a feature vector.
2. Attention mechanism is adopted to generate weights dynamically by capturing relationships between left and right contiguous words of the biomedical acronym.
3. Multiscale asymmetric convolution neural network can reduce computation quantity and obtain more feature information. Useful information can be obtained forward and backward using bidirectional long short-term memory networks.

The remainder of this paper is organized as follows. Related work is reported in Section 2. Section 3 describes the extraction of disambiguation features and how to generate disambiguation feature vectors. Attention neural network is given in Section 4. The process of training attention neural network is given in Section 5. Experimental analysis is given in Section 6. Section 7 gives conclusions.

2. Related Work

There are 3 kinds of biomedical WSD methods. They are, respectively, supervised methods, unsupervised ones, and knowledge-based ones.

2.1. Supervised Methods. Supervised methods use labeled data to train the biomedical WSD classifier. Liu applies 3 machine learning algorithms to biomedical WSD, including Naïve Bayes and decision lists, adaptation of decision lists, and mixed supervised learning method. Experiments show that the hybrid supervised method with Naïve Bayes performs best in biomedical WSD [5]. Stevenson extracts domain-independent linguistic features around the ambiguous word from the text. These features have been adapted for biomedical text disambiguation by adding CUIs [6] and Medical Subject Heading terms [7]. Son proposes a support vector machine with examplewise weights to solve the WSD task. Weights of training instances are adjusted according to their similarities to test data [8]. Moon builds a clinical sense inventory with 440 common abbreviations. By comparing this inventory with UMLS, Adam, and Stedman, he analyzes these clinical abbreviations and acronyms among diverse resources [9]. Yepes uses word embedding to improve traditional features. At the same time, a recurrent neural network based on LSTM nodes is used for biomedical WSD [10]. Festag applies word embedding and recurrent convolutional neural networks to medical term disambiguation, which map medical terms to multiple concepts in UMLS [11]. Wang proposes an interactive learning algorithm in which expert’s domain knowledge is used to build a medical WSD model. An expert can provide domain knowledge in 3 ways, including labeling instances, specifying indicative words of a sense, and highlighting support evidence in a labeled instance [12]. Bis proposes a novel deep neural network for biomedical WSD, in which a layered bidirectional LSTM network and a max-pooling layer along multiple time steps are built in order to create a dense representation of context [13]. Wei applies CNNs and LSTMs to capture semantic and syntactic features for bioconcept disambiguation [14]. But, a lot of human-annotated corpora are needed in the supervised biomedical WSD method.

2.2. Unsupervised Methods. Unsupervised methods use an unlabeled corpus to provide sense choice for a word in context. Agirre uses relations in UMLS to create a graph. At the same time, a personalized PageRank algorithm [15] is applied to rank semantic categories of ambiguous words based on their structural importance in graphs and relation to words in context [16]. Duan proposes a graph-based algorithm to cluster words into groups, in which the principle of finding the maximum margin between clusters is adopted [17]. Wanton gives a kernel-based method for biomedical WSD. Information in a knowledge base is used to construct an affinity matrix, and kernels are defined based on the matrix [18]. Rab applies six relation types in UMLS to build a graph for ambiguous words. At the same time, he gives a graph-based algorithm to disambiguate terms in a biomedical text [19]. Fernandez proposes a graph-based unsupervised algorithm to solve the WSD problem in a biomedical domain. When a knowledge base is built, contexts of ambiguous terms are considered [20]. Li proposes a language model based on Bi-LSTM, in which word order is considered and the entire sentential context is described. It generates high-quality context representations in an unsupervised manner [21]. Pesaranghader computes sense embeddings based on their text definitions in the Unified Medical Language System. At the time, he proposes a net to determine the semantic category of the ambiguous term [22]. But, the performance of the unsupervised biomedical WSD method is low.
2.3. Knowledge-Based Methods. Knowledge-based approaches apply external lexical resources to biomedical WSD, such as machine-readable dictionaries, thesauri, and ontologies. Rais considers that the terms in context have the same weight. Then, a modified SenseRelate algorithm [23] is given. He applies semantic similarity and relatedness measures to biomedical WSD. Then, the influence of context window size on WSD is evaluated [24]. Yepes compares 4 knowledge-based WSD methods in the biomedical domain. The method which uses semantic categories assigned to concepts in Metathesaurus performs the best [25]. Plaza studies the influence of 3 WSD algorithms on biomedical summarization, in which documents are mapped onto concepts in UMLS. Three WSD algorithms are, respectively, journal descriptor indexing, machine-readable dictionary, and automatic extracted corpus [26]. McInnes uses semantic similarity and relatedness measures to determine semantic categories of biomedical terms, which does not require human-annotated corpus and yields high disambiguation accuracy [27]. Garla adopts a directed concept graph to compute semantic similarity based on UMLS. Vertices represent concepts, and edges denote taxonomical relationships [28]. Based on neural word and concept embeddings, Sabbir combines cosine similarity, projection magnitude proportion, and a prior knowledge-based approach to determine the semantic category of the biomedical term [29]. Antunes uses unlabeled MEDLINE abstracts to generate word embeddings. Word embeddings are applied to compute embedding vectors in surrounding contexts around the ambiguous term. According to the similarity between context vector and concept vector, meanings of ambiguous terms are determined [30]. But, it is difficult to extract correct knowledge from lexical resources and apply them to biomedical WSD.

These 3 methods have their own shortcomings. The supervised WSD method achieves better performance. But, it needs a lot of annotated biomedical corpus. Unsupervised WSD method need not label biomedical corpus manually. But, disambiguation accuracy is not high. The knowledge-based method applies external lexical resources to biomedical WSD. But, it is difficult to extract correct knowledge from lexical resources and apply them to biomedical WSD correctly.

3. Feature Extraction

3.1. Preprocessing Text. Punctuations in contexts of the biomedical word have less semantic information and have little influence on determining its semantic category. At the same time, they will bring noises in the process of estimating the model's parameters. Regular expression of python is used to delete punctuations from sentences containing the biomedical word. Part of speech refers to grammatical features of a kind of word, which is their grammatical functions. It provides help for determining the relationship between two words. Nltk packet of python is adopted to label each word with part of speech in the sentence. Semantics is the sense of a word. Words with the same or similar sense are classified into a category. The purpose is to decrease data sparsity in the process of parameter estimation. Nltk packet of python is used to label each word with semantics in the sentence.

In the sentence “A message from ADA president Feldman.”, “ADA” is a biomedical word. Firstly, punctuation “.” is deleted. Secondly, every word is labeled with part of speech. For “A,” its part of speech is DT. For “message” and “president,” their part of speech is NN. For “from,” its part of speech is IN. For “ADA” and “Feldman,” their part of speech is NNP. Thirdly, every word is labeled with semantics. “A” is annotated with angstrom.n.01, “message” is labeled with message.n.01, “ADA” is annotated with adenosine_deaminase.n.01, and “president” is labeled with president.n.01. “Feldman” and “from” are annotated with “–1.”

3.2. Disambiguation Feature Extraction. Word nearby ambiguous word has more impact on the sense of the ambiguous word, but words far away from it have less one. In this paper, the biomedical word is viewed as the center. Morphology, part of speech, and semantic information from left and right lexical units are extracted as disambiguation features to determine its meanings. When the number of left or right contiguous vocabulary units is less than 2, the corresponding disambiguation feature is set to −1. The purpose is to ensure that each biomedical word has 12 disambiguation features.

For English sentences containing the biomedical word “ADA,” the process of extracting disambiguation features is shown as follows:

English sentence: A message from “ADA” president Feldman.

Part of speech tagging: A/DT message/NN from/IN ADA/NNP president/NN Feldman/NP

Semantic annotation: A/DT/angstrom.n.01 message/NN/message.n.01 from/IN/-1 ADA/NNP/adenosine_deaminase.n.01 president/NN/president.n.01 Feldman/NP/-1

The process of extracting disambiguation features is shown in Figure 1

3.3. CBOW Model. Word2vec’s CBOW [2] is used to generate a feature vector. The input of the CBOW model is a word vector corresponding to context-related words of a word, and its output is a probability distribution. The dimension of output is the same as that of the input. The gradient descent method is used to update input weights and output weights. After the training process, each word in the input layer is multiplied by input weights to get a word vector. The size of the word window is 4. The window contains its 2 left word units and 2 right ones. Twelve features from these 4 word units are input to the CBOW model. Word vector of “ADA” is computed as shown in Figure 2.

3.4. Generation of Disambiguation Feature Matrix. The feature vector is a real one that maps high-dimensional space into low one, which can represent a large amount of
potential information in a word. Twelve disambiguation features are extracted from a sentence containing the biomedical word “ADA.” CBOW model is used to convert each feature into a 100-dimensional feature vector.

In this paper, we design 3 optional methods to construct a feature matrix:

1. The first one uses the morphology of 4 adjacent vocabulary units around the biomedical word “ADA,” CBOW model is used to construct a feature vector. This feature vector is used to construct a $4 \times 100$ feature matrix $F$.

2. The second one considers positions of left and right adjacent words. Generally, if a context-related word

---

**Figure 1: Extract disambiguation features.**

**Figure 2: CBOW model of “ADA.”**
is closer to the biomedical word “ADA,” it is more important for determining the category of “ADA,” and its weight is bigger. The weighted sum of 12 feature vectors is used as a feature vector of the biomedical word “ADA,” whose dimension is 100. This feature vector is used to construct a $10 \times 10$ feature matrix $M$.

(3) The third one does not consider positions of left and right adjacent words. The disambiguation feature vector of the biomedical word “ADA” is denoted as $N = \{O(M_{L2}), O(P_{L1}), O(S_{L2}), O(M_{L1}), O(P_{L1}), O(S_{L1}), O(M_{R1}), O(P_{R1}), O(S_{R1}), O(M_{R2}), O(P_{R2}), O(S_{R2})\}$. Here, $O$ is the output of the Word2vec tool. Feature matrix $V=(V_1^T, V_2^T, \ldots , V_{12}^T)$ is constructed, where $V_1=O(M_{L2}), V_2=O(P_{L2}), V_3=O(S_{L2}), V_4=O(M_{L1}), V_5=O(P_{L1}), V_6=O(S_{L1}), V_7=O(M_{R1}), V_8=O(P_{R1}), V_9=O(S_{R1}), V_{10}=O(M_{R2}), V_{11}=O(P_{R2})$, and $V_{12}=O(S_{R2})$.

4. Disambiguation Model Based on Attention Neural Network

4.1. Attention Layer. The attention mechanism is used to generate weights dynamically to capture the relationship between left and right adjacent words of the biomedical word. Feature matrix $S$ is generated according to weight parameters. Feature matrix $V$ is input to attention layer, and feature vector $s_m$ is calculated as follows:

$$ a_{mn} = \left( V_m \cdot W^Q \right) \left( V_m \cdot W^K \right)^T / \sqrt{d}, \quad (1) $$

$$ \phi_{mn} = \frac{\exp(a_{mn})}{\sum_{l=1}^{12} \exp(a_{ml})}, \quad (2) $$

$$ s_m = \sum_{l=1}^{12} \phi_{ml} \cdot (V_l \cdot W^L), \quad (3) $$

where $W^Q, W^K$, and $W^L$ are weight matrices, $\phi_{mn}$ is the weight coefficient, and $s_m$ is the output feature vector. Here, $d$ is the dimension of the input, which plays a role in adjusting the inner product. Values of $m$ and $n$ are, respectively, set to 1, 2, . . . , 12.

Equation (1) is used to compute the correlation strength of two elements. Weight coefficient $\phi_{mn}$ is calculated based on correlation strength $a_{mn}$ as shown in (2). $s_m$ is computed as shown in (3). Feature matrix $S$ is constructed as follows:

$$ S = \left[ s_1^T \ s_2^T \cdots s_{12}^T \right]^T. \quad (4) $$

For the biomedical word “ADA,” attention operation is used to process feature matrix $S$ as shown in Figure 3.

4.2. Convolutional Layer. The convolutional neural network is a deep learning model, which includes the input layer, convolutional layer, pooling layer, dropout layer, and fully connected layer. CNN can extract local features from data through convolution operations. CNN has representation learning ability, which can classify the input information shift-invariantly based on its hierarchical structure. CNN shares convolution kernels weight and multifeature graph, which can be used to process high-dimension data. After a series of convolution and pooling operations are implemented, the parameters of the model and the risk of overfitting are reduced. The convolutional neural network can capture local correlation of space-time structure, which has achieved excellent performance in natural language processing, computer vision, and image processing.

Here, asymmetric convolution proposed by Szegedy [31] is introduced, where $k_i \times h$ convolution is split into $k_i \times 1$ convolution and $1 \times h$ convolution. The biggest advantage of this method is that it reduces computation quantity, and its effect is similar to that of two-dimensional convolution. The size of the convolutional kernel is $k_i \times 1, i = 1, 2, 3$. Multiple convolution kernels of different sizes are set to get different features. The first convolution operation corresponding to $1 \times h$ convolution kernel is applied to $s_m$ and generates the corresponding feature $z_m^i$ as follows:

$$ z_m^i = R(c) = R\left( w_i^1 \times s_m + b_i^1 \right), \quad (5) $$

where $m = 1, 2, \ldots , 12, w_i^1$ is convolution kernel in which 1 means the first convolution, $i$ is used to distinguish 3 parallel asymmetric convolution operations, $b_i^1$ denotes activation function, and $c$ represents net activation of the convolutional layer. ReLU activation function $R(\bullet)$ is as follows:

$$ R(c) = \max(0, c). \quad (6) $$

After $1 \times h$ convolution kernel is used as shown in equations (5) and (12), eigenvalues are obtained. Feature mapping constructed by these 12 eigenvalues is as follows:

$$ Z_i^c = [z_1^i, z_2^i, \ldots , z_{12}^i]^T. \quad (7) $$

The second convolution operation corresponding to $k_i \times 1$ convolution kernel is applied to $Z_i^c$, and feature value $c_i^c$ is computed as follows:

$$ c_i^c = R(e) = R\left( w_i^2 \times z_m^{i+c} + b_i^2 \right), \quad (8) $$

where $m = 1, 2, \ldots , 12, w_i^2$ is convolution kernel in which 2 means the second convolution, $i$ is used to distinguish 3 parallel asymmetric convolution operations, $b_i^2$ denotes bias, $R(\bullet)$ is activation function, and $e$ represents net activation of the convolutional layer. ReLU activation function $R(\bullet)$ is shown as follows:

$$ R(e) = \max(0, e). \quad (9) $$

In the second convolution, 3 asymmetric operations use the same number of convolution kernels. But, the size of the convolution kernel is different. So, 3 asymmetric convolution operations in the second convolution can output the same number of feature mapping $C_i^c$, as shown in the following equation:

$$ C_i^c = [c_1^c, c_2^c, \cdots, c_{12}^c]^T. \quad (10) $$
Features with the same index in the convolution window are averaged to generate feature mapping $D$ as follows:

$$d_j = \frac{c^1_j + c^2_j + c^3_j}{3},$$

(11)

$$D = [d_1, d_2, \ldots, d_{12}]^T.$$

(12)

where $j = 1, 2, \ldots, 12$.

According to the index in the convolution window, feature mappings $C$ of 3 asymmetric convolution operations are concatenated with feature mapping $D$ to generate $E$ as follows:

$$E_j = [c^1_j, c^2_j, c^3_j, \ldots, d_j],$$

(13)

$$E = [E_1, E_2, \ldots, E_{12}]^T.$$  

(14)

The above process is called Av-ACNN. If feature mappings $C^1, C^2,$ and $C^3$ are concatenated directly and input into Bi-LSTM, the process is called ACNN.

4.3. Bi-LSTM Layer. LSTM is a special recurrent neural network (RNN), which can capture longer distance information. It uses a set of gate controllers to solve effectively gradient disappearance and explosion of RNN. Bi-LSTM is composed of forward LSTM and backward LSTM, which makes up for the shortcomings of LSTM.

LSTM infers semantics of biomedical word based on the previous input information but cannot include the subsequent information into the reference, which will affect WSD accuracy. In fact, left and right contexts around the biomedical word can all influence the process of WSD. If you access the right context as you access the left one, it is very beneficial for biomedical WSD. Bi-LSTM is composed of two LSTMs. One is forward LSTM and the other is backward LSTM. They represent, respectively, the left context and the right one of the biomedical word. Bi-LSTM is very suitable for sequence annotation tasks with a top-down relationship. It is often used to model context information in natural language processing and provides help for biomedical WSD.

In this paper, LSTM inputs data at the multistep, and outputs results at the last time step. Results at the last time step are added and input into a fully connected layer. LSTM unit contains memory unit $C_t$ and 3 gate controllers. They are, respectively, input gate $i_t$, forget gate $f_t$, and output gate $o_t$. These 3 gates control update of memory unit $C_t$ and output of hidden layer state $h_t$. The output of $h_t$ is computed as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, E_t] + b_i),$$

(15)

$$f_t = \sigma(W_f \cdot [h_{t-1}, E_t] + b_f),$$

(16)

$$o_t = \sigma(W_o \cdot [h_{t-1}, E_t] + b_o),$$

(17)
\[ a_t = \tanh(W_c \cdot [h_{t-1}, E_t] + b_c), \]
\[ C_t = f_t \odot C_{t-1} + i_t \odot a_t, \]
\[ h_t = o_t \odot \tanh(C_t), \]

where \( W_i, W_f, W_c, \) and \( W_o \) are, respectively, weight matrices of input gate, forget gate, candidate state, and output gate. Here, \( b_i, b_f, b_c, \) and \( b_o \) are, respectively, their bias terms. \( t = 1, 2, \ldots, 12. \) \( \sigma(\cdot) \) represents the sigmoid function. \( \tanh(\cdot) \) is the hyperbolic tangent activation function. \( \odot \) denotes the product of corresponding elements.

At the last time step, the output of the hidden layer in forward LSTM is \( H^\rightarrow \), and the output of the hidden layer in backward LSTM is \( H^\leftarrow \).

### 4.4. Fully Connected Layer and Semantic Classification

Outputs \( \vec{H}^\rightarrow \) and \( H^\leftarrow \) of Bi-LSTM are input to a fully connected layer. The softmax function is applied to map the output of the neuron to the interval \((0, 1)\) as shown in equation (21). The purpose is to determine semantic categories of biomedical words.

\[
Q = f(Y) = f\left(W_{H^\rightarrow} \cdot \vec{H} + W_{H^\leftarrow} \cdot H + B\right). \tag{21}
\]

In (21), \( Q = (P(x_1 | c), P(x_2 | c), \ldots, P(x_r | c)) \) and \( Y = (Y_1, Y_2, \ldots, Y_l) \) represent net activation. \( W_{H^\rightarrow} \) and \( W_{H^\leftarrow} \) denote weight parameters. \( B \) represents the bias term. \( \vec{H} \) is the output in the hidden layer of forward LSTM and \( H \) is output in the hidden layer of backward LSTM.

\( P(x_i | c) \) is the probability of biomedical word \( c \) under the semantic category \( x_i \) as shown in the following equation:
\[ P(x_i|c) = \frac{\exp(Y_i)}{\sum_{j=1}^{r} \exp(Y_j)}. \] (22)

In equation (22), \( c \) has \( r \) semantic categories and \( x_i \) represents the \( i \)th semantic class, \( i = 1, 2, ..., r \).

In probability distribution \( P(x_1|c), P(x_2|c), \ldots, P(x_r|c) \) of biomedical word \( c \), semantic category \( s \) with the highest probability is selected as the predicted one as shown in the following equation:

\[ s = \arg \max_{i=1,2,...,r} (P(x_i|c)). \] (23)

5. Biomedical WSD Based on Attention Neural Network

Attention mechanism, Av-ACNN, and Bi-LSTM are combined to disambiguate biomedical words. The training process of attention mechanism, Av-ACNN, and Bi-LSTM includes forward propagation and backpropagation. Semantic classification is forward propagation. Gradient calculation and parameter optimization are backpropagation. The training process of attention mechanism, Av-ACNN, and Bi-LSTM is shown as follows.

Train_Av-ACNN_Bi-LSTM( ).

Input: feature vector \( N \) of biomedical word \( c \) and artificially labeled semantic category \( h_s \).

Output: optimized parameter set \( \theta = \{W_{\rightarrow}, W_{\leftarrow}, \text{B}, \text{W}, \text{bi}, \text{Wf}, \text{bf}, \text{Wc}, \text{bc}, \text{Wo}, \text{bo}, w^r, b^r, w^i, b^i, W_Q, \text{WK}, \text{WL}\} \).

Step 1: initialize iteration number \( n \) and parameter set \( \theta \).

Step 2: matrix \( V \) is constructed based on \( N \).

Step 3: for \( (m = 1; m \leq n; m++) \) do:

Forward_Propagation( );

Compute_Gradient( );

Update_Parameter( );

Forward_Propagation( ).

In equation (22), \( c \) has \( r \) semantic categories and \( x_i \) represents the \( i \)th semantic class, \( i = 1, 2, ..., r \).

In probability distribution \( P(x_1|c), P(x_2|c), \ldots, P(x_r|c) \) of biomedical word \( c \), semantic category \( s \) with the highest probability is selected as the predicted one as shown in the following equation:

\[ s = \arg \max_{i=1,2,...,r} (P(x_i|c)). \] (23)

Step 6: according to equation (23), the category of biomedical word \( c \) is determined.

Backward propagation includes gradient calculation and parameter optimization. The process of gradient computation is shown as follows.

Compute_Gradient( ).

Step 1: loss value \( J \) is computed as

\[ J = -\sum_{i=1}^{r} y_i \cdot \ln(P(x_i|l)), \] (24)

where \( y_1, y_2, ..., y_r \) are the one-hot codes of \( h_s \).

Step 2: gradient \( \delta_y \) of the output layer is calculated as

\[ \delta_y = \frac{\partial J}{\partial Y} = f'(Y) \circ \delta_y, \] (25)

where \( \circ \) is the product of corresponding elements.

Step 3: gradients \( \delta_{ht} \) and \( \delta_{ci} \) at the last time step of the forward LSTM and backward LSTM are, respectively, calculated as shown as follows:

\[ \delta_{ht} = \frac{\partial J}{\partial h_t} = W_H^{T} \cdot \delta_y, \] (26)

\[ \delta_{ci} = \delta_{ci} \circ o_t \circ (1 - \tanh^2(C_t)), \] (27)

where \( l \) represents the last time step, \( H = \{\overrightarrow{H}, \overleftarrow{H}\} \), and \( C_t = \{C_t^a, C_t^b\} \).

Step 4: gradients \( h_t \) and \( C_t \) at any time in LSTM are computed as shown as follows:

\[ \delta_{h_t} = \frac{\partial J}{\partial h_t} = \left( \frac{\partial h_{t+1}}{\partial h_t} \right)^T \cdot \delta_{h_{t+1}}, \] (28)

\[ \delta_{c_t} = \delta_{c_{t+1}} \circ f_{t+1} + \delta_{h_t} \circ o_t \circ (1 - \tanh^2(h_t)), \] (29)

where \( C_t \) represents gradient at \( t+1 \) time, \( h_t = \{\overrightarrow{h_t}, \overleftarrow{h_t}\} \), and \( C_t = \{\overrightarrow{C_t}, \overleftarrow{C_t}\} \).

Step 5: gradient \( \delta_{ti}^2 \) of the second convolution layer is calculated as

\[ \delta_{ti}^2 = \frac{\partial J}{\partial e}, \] (30)

where \( i \) is the \( i \)th asymmetric convolution and \( 2 \) is the second convolution layer.

Step 6: gradient \( \delta_{ti}^1 \) of the first convolutional layer is computed as

\[ \delta_{ti}^1 = \delta_{ti}^2 \cdot \text{rot180}(w_i^2) \circ R'(c), \] (31)

where 1 represents the first convolution layer and rot180(\( \cdot \)) is the operation of rotating 180°.

The process of parameter update is shown as follows.

Update_Parameter( ).
Discrete Dynamics in Nature and Society 9

Step 1: update weights $W_{W \rightarrow H}$ and $W_{H}^{\dagger}$ and bias $B$ of the fully connected layer as follows:

$$W_{H} = W_{H}^{\dagger} - \alpha \cdot (\delta_{y} \cdot H^{T}),$$

(32)

$$B = B - \alpha \cdot \delta_{y},$$

(33)

where $\alpha$ is learning rate and $H = [\bar{H}, \bar{H}]$.

Step 2: update weight matrices $W_{v}$ and $W_{c}$ and bias terms $b_{v}$ and $b_{c}$ in LSTM cell as follows:

$$W_{v} = W_{v} - \alpha \cdot \sum_{t=1}^{l} \left[ \delta_{C_{v} \circ C_{t-1}} \circ y_{t} \circ \left(1 - y_{t}\right) \right] \cdot \left(h_{t-1}\right)^{T},$$

(34)

$$W_{c} = W_{c} - \alpha \cdot \sum_{t=1}^{l} \left[ \delta_{C_{c} \circ C_{t-1}} \circ (1 - a_{t})^{2} \right] \cdot \left(h_{t-1}\right)^{T},$$

(35)

$$b_{v} = b_{v} - \alpha \cdot \sum_{t=1}^{l} \left[ \delta_{C_{v} \circ C_{t-1}} \circ y_{t} \circ \left(1 - y_{t}\right) \right],$$

(36)

$$b_{c} = b_{c} - \alpha \cdot \sum_{t=1}^{l} \left[ \delta_{C_{c} \circ C_{t-1}} \circ (1 - a_{t})^{2} \right],$$

(37)

where $l$ is the last time step, $\alpha$ is learning rate, and $y = \{i, f, o\}$.

Step 3: update weight $w_{m}^{2}$ and bias $b_{m}^{2}$ in the second convolutional layer as follows:

$$w_{m}^{2} = w_{m}^{2} - \alpha \cdot \left(\delta_{m} \cdot Z'\right),$$

(38)

$$b_{m}^{2} = b_{m}^{2} - \alpha \cdot \sum_{k,j} \delta_{C_{k} \circ C_{j}}^{2},$$

(39)

where $Z'$ is the output of the first convolution layer.

Step 4: Update weight $w_{m}^{1}$ and bias $b_{m}^{1}$ in the first convolutional layer as follows:

$$w_{m}^{1} = w_{m}^{1} - \alpha \cdot \left(\delta_{m} \cdot S\right),$$

(40)

$$b_{m}^{1} = b_{m}^{1} - \alpha \cdot \sum_{p,q} \delta_{C_{p} \circ C_{q}}^{1},$$

(41)

When the semantic category of biomedical word $c$ is determined, disambiguation features are extracted from its 4 adjacent lexical units. Feature matrix $V$ is constructed and input into the attention layer. According to equation (4), the attention layer outputs feature matrix $S$, which is input into the convolutional layer. Based on equation (10), feature mappings $C_{1}^{i}$, $C_{2}^{i}$, and $C_{3}^{i}$ of the asymmetric convolution are computed. According to indices of the convolution window, $C_{1}^{i}$, $C_{2}^{i}$, and $C_{3}^{i}$ are fused twice to obtain feature mapping $E$, which is input into the Bi-LSTM layer. The output of the Bi-LSTM layer is computed based on equation (20). According to equation (22), probability distribution $P(x_{i} | c)$ of biomedical word $c$ under semantic category $x_{i}$ is calculated.

According to equation (23), its semantic category is determined.

6. Experiments

6.1. Training and Test Corpus. MSH dataset is a WSD dataset in the biomedical domain. It consists of 106 ambiguous abbreviations, 88 ambiguous terms, and 9 words, which are a combination of both, for a total of 203 ambiguous entities. For each ambiguous term and abbreviation, the dataset contains a maximum of 100 sentences per sense obtained from MEDLINE.

51 biomedical words are selected from the MSH dataset. There are 38 biomedical words with two semantic categories, including AA, ADA, ADP, ALS, BAT, BLM, CDR, Cilia, DI, eCG, EM, EMS, FAS, Fish, FTC, GAG, Ganglion, HCl, HR, IA, INDO, IP, ITP, JP, MCC, Medullary, MRS, NM, OCD, ORI, PAF, Phosphorylase, RB, SARS, SCD, THYMUS, and WT1. There are 11 biomedical words with 3 semantic categories. They are, respectively, Ala, Cold, Cortical, CP, DDS, Ice, len, Lupus, PCP, RA, and TAT. There are only one biomedical word with 4 semantic categories and one biomedical word with 5 semantic categories in the MSH dataset. They are, respectively, Ca and PCA, which are all selected. Sentences containing these 51 biomedical words are used as training corpus and test corpus to measure the proposed method’s performance.

6.1.1. Experiment Analysis. Ten groups of experiments are carried out, and average disambiguation accuracy is used to evaluate the performance of the WSD classifier, which is defined as
where $N$ is the number of all biomedical words, $m_i$ is the number of test sentences correctly classified for the $i$th biomedical word, $n_i$ is the number of all test sentences containing the $i$th biomedical word, $p_i$ is disambiguation accuracy of the $i$th biomedical word, and $p_{avg}$ is average disambiguation accuracy.

In Experiment 1 and Experiment 3, method (1) is used to construct feature matrix $F$. In Experiment 1, CNN is applied to determine the semantic category of the biomedical word. In Experiment 2, LSTM is used to determine its semantic class. In Experiment 3, Bi-LSTM is adopted to determine its semantic category. In Experiment 4, morphology, part of speech, and semantic information in 4 adjacent vocabulary units of the biomedical word are used as disambiguation features. Feature matrix $V$ is constructed by method (3). The proposed framework is adopted to determine its semantic class. Disambiguation accuracies from Experiment 1 to Experiment 4 are shown in Table 1.

The average disambiguation accuracy of Experiment 2 is 1.19% higher than that of Experiment 1. Experiments show that, compared to CNN, LSTM is more suitable for the disambiguation of the MSH corpus. The average disambiguation accuracy of Experiment 3 is higher 1.35% than that of Experiment 2. The reason is that Bi-LSTM takes account of contextual information from two directions. In Experiment 4, the proposed network is used to disambiguate biomedical words. The disambiguation accuracy of Experiment 4 is 5.6% higher than that of Experiment 3. The reason is that morphology, part of speech, and semantic information are extracted as disambiguation features in Experiment 4, but Experiment 3 only considers morphology. The proposed network is better than Bi-LSTM.

In Experiment 5, Experiment 6, and Experiment 7, morphology, part of speech, and semantic information in 4 left and right lexical units of the biomedical word are selected as disambiguation features. At the same time, feature matrix $M$ is constructed by method (2) in Experiment 5, Experiment 6, and Experiment 7. In Experiment 5, ACNN and LSTM are combined to determine the semantic class of the biomedical word. In Experiment 6, ACNN and Bi-LSTM are fused to determine its semantic category. In Experiment 7, Av-ACNN and Bi-LSTM are combined to determine the semantic class of the biomedical word. The training corpus of MSH is used to optimize the WSD classifier, and the optimized WSD model is applied to classify the test corpus of MSH. Disambiguation accuracies from Experiment 5 to Experiment 7 are shown in Table 2.

The average disambiguation accuracy of Experiment 6 is 1.25% higher than that of Experiment 5. The reason is that Bi-LSTM is composed of forward LSTM and backward LSTM, in which feature information in context is obtained, respectively, from two directions. But, LSTM only obtains feature information from one direction. Therefore, the disambiguation accuracy of Bi-LSTM is higher than that of LSTM. The average disambiguation accuracy of Experiment 7 is 0.72% higher than that of Experiment 6. The reason is that the weighted average $D$ of $C_1$, $C_2$, and $C_3$ is computed according to indexes of the convolutional window in Experiment 7. Then, $C_1$, $C_2$, $C_3$, and $D$ are concatenated according to indexes.
Table 2: Disambiguation accuracies from Experiment 5 to Experiment 7.

| Word      | Exp5 (%) | Exp6 (%) | Exp7 (%) |
|-----------|----------|----------|----------|
| AA        | 91.94    | 93.55    | 93.55    |
| ADA       | 92.31    | 93.85    | 96.92    |
| ADP       | 94.12    | 94.12    | 94.12    |
| ALS       | 92.86    | 92.86    | 88.10    |
| BAT       | 94.44    | 94.44    | 96.30    |
| BLM       | 90.63    | 95.31    | 95.31    |
| CDR       | 96.30    | 96.30    | 96.30    |
| Cilia     | 84.21    | 84.21    | 86.84    |
| DI        | 95.24    | 90.48    | 95.24    |
| eCG       | 84.38    | 82.81    | 87.50    |
| EM        | 93.94    | 93.94    | 93.94    |
| EMS       | 80.00    | 81.54    | 86.15    |
| FAS       | 96.92    | 98.46    | 100.00   |
| Fish      | 82.81    | 85.94    | 85.94    |
| FTC       | 96.77    | 96.77    | 98.39    |
| GAG       | 89.29    | 82.14    | 85.71    |
| Ganglion  | 80.00    | 83.08    | 83.08    |
| HCl       | 93.85    | 96.92    | 96.92    |
| HR        | 94.29    | 94.29    | 94.29    |
| IA        | 82.14    | 82.14    | 85.71    |
| INDO      | 97.37    | 94.74    | 100.00   |
| IP        | 87.50    | 91.67    | 91.67    |
| ITP       | 87.50    | 85.00    | 87.50    |
| JP        | 91.67    | 95.83    | 95.83    |
| MCC       | 94.29    | 94.29    | 94.29    |
| Medullary | 88.89    | 90.48    | 92.06    |
| MRS       | 91.67    | 94.44    | 94.44    |
| NBS       | 92.11    | 92.11    | 92.11    |
| NM        | 84.21    | 89.47    | 84.21    |
| OCD       | 89.23    | 87.69    | 100.00   |
| ORI       | 79.41    | 79.41    | 79.41    |
| PAF       | 100.00   | 100.00   | 100.00   |
| Phosphorylase | 81.13  | 90.57    | 81.13    |
| RB        | 95.24    | 92.06    | 93.65    |
| SARS      | 83.08    | 86.15    | 86.15    |
| SCD       | 95.45    | 96.97    | 96.97    |
| THYMUS    | 89.86    | 86.96    | 89.86    |
| WTI       | 70.69    | 74.14    | 74.14    |
| Ala       | 74.36    | 74.36    | 76.92    |
| Cold      | 80.72    | 81.93    | 81.93    |
| Cortical  | 69.39    | 78.57    | 79.59    |
| CP        | 92.54    | 95.52    | 94.03    |
| DDS       | 92.45    | 94.34    | 96.23    |
| Ile       | 78.57    | 85.71    | 85.71    |
| lens      | 74.74    | 74.74    | 75.79    |
| Lupus     | 84.69    | 87.76    | 86.73    |
| PCP       | 85.57    | 91.75    | 85.57    |
| RA        | 93.62    | 91.49    | 92.55    |
| TAT       | 60.00    | 64.62    | 66.15    |
| Ca        | 64.29    | 62.70    | 64.29    |
| PCA       | 87.86    | 90.00    | 88.57    |
| $P_{avg}$(%) | 87.15  | 88.40    | 89.12    |

of the convolutional window to get feature mapping $E$. Because more disambiguation information is included in $E$, the disambiguation effect is improved.

In Experiment 8, morphology, part of speech, and semantic information are extracted as disambiguation features from 4 adjacent lexical units of the biomedical word. Feature matrix $V$ is generated by method (3). Av-ACNN and Bi-LSTM are combined to determine the semantic class of the biomedical word. Disambiguation accuracies of Experiment 4, Experiment 7, and Experiment 8 are shown in Table 3.

The disambiguation effect of Experiment 7 is better than that of Experiment 8. This shows that the disambiguation ability of the feature matrix constructed by method (2) is better than that of the feature matrix constructed by method (3). The reason is that the position of the adjacent word is considered in method (2). Generally, if a context-related word is closer to the biomedical word, it is more important for determining the category of the biomedical word and its weight is bigger. The disambiguation accuracy of Experiment 4 is 2.26% higher than that of Experiment 7. This is because that attention layer generates dynamically weight coefficients, which are applied to construct the feature matrix in Experiment 4. In Experiment 7, weight coefficients are manually set to construct a feature matrix. Therefore, the disambiguation effect of Experiment 4 is better than that of Experiment 7. The disambiguation accuracy of Experiment 4 is 3.14% higher than that of Experiment 8. In Experiment 7, the attention layer is not used. This shows that the attention layer is helpful for improving the disambiguation effect.

The size of the convolution kernel has a great influence on feature extraction. If the convolution kernel is too small, it is not able to extract effective features. If the convolution kernel is too large, the computation quantity will be increased. Three groups of experiments have been conducted in which 3 convolution kernels with different sizes are used to obtain discriminative features. In Experiment 4, Experiment 9, and Experiment 10, morphology, part of speech, and semantic information in 4 left and right lexical units of the biomedical word are selected as disambiguation features. At the same time, method (3) is used to construct feature matrix $V$ and the attention network proposed in this paper is applied to determine the semantic class of the biomedical word. In Experiment 4, the second convolution kernels for 3 asymmetric convolution operations are, respectively, 2, 3, and 5 in size. In Experiment 9, the second convolution kernels for 3 asymmetric convolution operations are, respectively, 2, 3, and 5 in size. In Experiment 10, the second convolution kernels for 3 asymmetric convolution operations are, respectively, 1, 2, and 3 in size. The disambiguation accuracies of Experiment 4, Experiment 9, and Experiment 10 are shown in Table 4.

The average disambiguation accuracy of Experiment 4 is higher than that of Experiment 9 and Experiment 10. It shows that the size of the convolution kernel is suitable in Experiment 4.

From Experiment 1 to Experiment 8, the average disambiguation accuracies of biomedical words with 2, 3, 4, and 5 semantic categories are calculated. Experimental results are shown in Figure 6.

From Figure 6, it can be seen that the average disambiguation accuracy of the proposed method is better than that of other methods under 2, 3, 4, and 5 semantic classes. The average disambiguation accuracy under 2 semantic categories is higher than that under 3 and 4 ones. The reason is that when the number of semantic classes is less, the
difficulty of biomedical WSD is smaller. But, the disambiguation accuracy of the five categories is relatively high. The reason is that there is only one ambiguous word with five categories, and the distribution of its training corpus may be in accordance with that of its test one.

Finally, the time cost of the proposed model is analyzed. Here, \( n \) is the sequence length, \( d \) is the representation dimension, \( k \) is the kernel size of convolutions, \( c \) is the number of categories, and \( s \) is the number of support vectors. The run-time complexity of CNN is \( O(k \cdot n \cdot d) \). The run-time
complexity of LSTM and Bi-LSTM is $O(n \cdot d^2)$. The run-time complexity of attention is $O(n^2 \cdot d)$. So, the run-time complexity of the proposed method is $O(n^2 \cdot d + k \cdot n \cdot d + n \cdot d^2)$. The run-time complexity of KNN is $O(n \cdot d)$. The run-time complexity of logistic regression is $O(d)$. The run-time complexity of Naive Bayes is $O(n \cdot d)$. The run-time complexity of SVM is $O(s)$. Although the run-time complexity of the proposed method is the biggest, its average disambiguation accuracy is the highest.

7. Conclusions

Morphology, part of speech, and semantic information are extracted as disambiguation features from 4 adjacent lexical units of the biomedical word in this paper. The attention mechanism is used to generate a feature matrix, from which Av-ACNN and Bi-LSTM are used to extract discriminative features. Based on discriminative features, the softmax function is applied to determine the category of biomedical word. Experimental results show that the proposed method is more suitable for biomedical WSD than other methods.

Data Availability

MSH data can be downloaded from https://wsd.nlm.nih.gov/collaboration.shtml.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grants nos. 61502124 and 60903082), the Fundamental Research Foundation for Universities of Heilongjiang Province (Grant no. LGYC2018JC014), and the Natural Science Foundation of Heilongjiang Province of China (Grant no. F2017014).

References

[1] O. Bodenreider, “The unified medical language system (UMLS): integrating biomedical terminology,” Nucleic Acids Research, vol. 32, no. 1, pp. D267–D270, 2004.
[2] T. Mikolov, K. Chen, and G. Corrado, “Efficient estimation of word representations in vector space,” in Proceedings of the 1st International Conference On Learning Representations, Scottsdale, Arizona, May 2013.
[3] T. Mikolov, I. Sutskever, and K. Chen, “Distributed representations of words and phrases and their compositionality,” in Proceedings of the 27th Annual Conference On Neural Information Processing Systems, pp. 3111–3119, NV, USA, May 2013.
[4] J. Pennington, R. Socher, and C. D. Manning, “Glove: global vectors for word representation,” in Proceedings of the 2014 Conference On Empirical Methods In Natural Language Processing, pp. 1532–1543, Doha, Qatar, October 2014.
[5] H. Liu, V. Teller, and C. Friedman, “A multi-aspect comparison study of supervised word sense disambiguation,” Journal of the American Medical Informatics Association, vol. 11, no. 4, pp. 320–331, 2004.
[6] B. T. McInnes, T. Pedersen, and J. Carlis, “Using UMLS concept unique identifiers (CUIs) for word sense disambiguation in the biomedical domain,” in Proceedings of the AMIA Annual Symposium, pp. 533–537, Chicago, IL, USA, October 2007.
[7] M. Stevenson, Y. Guo, and R. Gaizauskas, “Disambiguation of biomedical text using diverse sources of information,” BMC Bioinformatics, vol. 9, no. S1, 2008.
[8] J. W. Son and S. B. Park, “Learning word sense disambiguation in biomedical text with difference between training and test distributions,” International Journal of Data Mining and Bioinformatics, vol. 6, no. 2, pp. 216–237, 2012.
[9] S. Moon, S. Pakhomov, N. Liu, J. O. Ryan, and G. B. Melton, “A sense inventory for clinical abbreviations and acronyms created using clinical notes and medical dictionary resources,”
A. D. Fernandez, M. Stevenson, and J. M. Romo, “Co-occurrence graphs for word sense disambiguation in biomedical documents,” *Journal of the American Medical Informatics Association*, vol. 23, no. 5, pp. 882–886, 2013.

A. Sabbir, A. J. Yepes, and R. Kavuluru, “Knowledge-based biomedical word sense disambiguation with neural concept embeddings,” in *Proceedings of the 17th IEEE International Conference On Bioinformatics And Bioengineering*, pp. 163–170, Washington D.C., USA, October 2017.

R. Antunes and S. Matos, “Supervised learning and knowledge-based approaches applied to biomedical word sense disambiguation,” *Journal of Integrative Bioinformatics*, vol. 14, no. 4, 2017.

C. Szegedy, V. Vanhoucke, S. Ioffe et al., “Rethinking the inception architecture for computer vision,” in *Proceedings of the 29th IEEE Conference On Computer Vision And Pattern Recognition*, pp. 2818–2826, Las Vegas, NV, USA, June 2016.

A. Jimeno Yepes, “Word embeddings and recurrent neural networks based on long-short term memory nodes in supervised biomedical word sense disambiguation,” *Journal of Biomedical Informatics*, vol. 73, pp. 137–147, 2017.

S. Festag and C. Spreckelsen, “Word sense disambiguation of medical terms via recurrent convolutional neural networks,” *Studies in Health Technology and Informatics*, vol. 236, pp. 8–15, 2017.

Y. Wang, K. Zheng, H. Xu, and Q. Mei, “Interactive medical word sense disambiguation through informed learning,” *Journal of the American Medical Informatics Association*, vol. 25, no. 7, pp. 800–808, 2018.

D. Bis, C. L. Zhang, and X. W. Liu, “Layered multistep bidirectional long short–term memory networks for biomedical word sense disambiguation,” in *Proceedings of the 2018 IEEE International Conference On Bioinformatics And Biomedicine*, pp. 313–320, Madrid, Spain, December 2018.

C. H. Wei, K. Lee, R. Leaman et al., “Biomedical mention disambiguation using a deep learning approach,” in *Proceedings of the 10th ACM International Conference On Bioinformatics, Informatcis*, pp. 307–313, New York, NY, USA, September 2019.

E. Agirre and A. Soroa, “Personalizing PageRank for word sense disambiguation,” in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 33–41, Athens, Greece, March 2009.

E. Agirre, A. Soroa, and M. Stevenson, “Graph-based word sense disambiguation of biomedical documents,” *Bioinformatics*, vol. 26, no. 22, pp. 2889–2896, 2010.

W. S. Duan, M. Song, and A. Yates, “Fast max-margin clustering for unsupervised word sense disambiguation in biomedical texts,” *BMC Bioinformatics*, S3, vol. 10, 2009.

T. M. Wanton, R. B. Llavori, and A. J. Yepes, “Preliminary results for biomedical word sense disambiguation based on semantic clustering,” in *Proceedings of the 22th National Workshop On Database And Expert Systems Applications*, pp. 460–464, Toulouse, France, August 2011.

W. G. E. Rab, O. R. Zaiane, and M. E. Hajj, “Unsupervised graph-based word sense disambiguation of biomedical documents,” in *Proceedings of the 15th International Conference on E–Health Networking Applications and Services*, pp. 649–652, Lisbon, Portugal, October 2013.

A. D. Fernandez, M. Stevenson, and J. M. Romo, “Co-occurrence graphs for word sense disambiguation in the biomedical domain,” *Artificial Intelligence in Medicine*, vol. 87, pp. 9–19, 2018.

Z. Li, F. Yang, and Y. Luo, “Context embedding based on Bi-LSTM in semi-supervised biomedical word sense disambiguation,” *IEEE Access*, vol. 7, pp. 72928–72935, 2019.

A. Pesaranghader, S. Matwin, M. Sokolova, and A. Pesaranghader, “deepBioWSD: effective deep neural word sense disambiguation of biomedical text data,” *Journal of the American Medical Informatics Association*, vol. 26, no. 5, pp. 438–446, 2019.

S. Patwardhan, S. Banerjee, and T. Pedersen, “SenseRelate: target word-a generalized framework for word sense disambiguation,” in *Proceedings of the ACL Interactive Poster And Demonstration Sessions*, pp. 73–76, MI, USA, June 2005.

M. Rais and A. Lachkar, “Biomedical word sense disambiguation context-based: improvement of SenseRelate method,” in *Proceedings Of the 2016 International Conference On Information Technology For Organizations Development*, Fez, March 2016.

A. J. Yepes and A. R. Aronson, “Knowledge-based biomedical word sense disambiguation: comparison of approaches,” *BMC Bioinformatics*, vol. 11, 2010.