Original Paper

Using a Large Margin Context-Aware Convolutional Neural Network to Automatically Extract Disease-Disease Association from Literature: Comparative Analytic Study

Po-Ting Lai[1], PhD; Wei-Liang Lu[2], MSc; Ting-Rung Kuo[2], MSc; Chia-Ru Chung[2], PhD; Jen-Chieh Han[2], MSc; Richard Tzong-Han Tsai[2*], PhD; Jorng-Tzong Horng[2,3*], PhD

[1]Department of Computer Science National Tsing Hua University, Hsinchu, Province of China Taiwan

[2]Department of Computer Science & Information Engineering, National Central University, Taoyuan, Province of China Taiwan

[3]Department of Bioinformatics and Medical Engineering, Asia University, Taichung, Province of China Taiwan

[*]these authors contributed equally

Corresponding Author:
Richard Tzong-Han Tsai, PhD
Department of Computer Science & Information Engineering, National Central University
No 300, Zhongda Road, Zhongli District
Taoyuan
Province of China Taiwan
Phone: 886 3 422 7151 ext 35323
Email: thtsai@csie.ncu.edu.tw

Abstract

Background: Research on disease-disease association (DDA), like comorbidity and complication, provides important insights into disease treatment and drug discovery, and a large body of the literature has been published in the field. However, using current search tools, it is not easy for researchers to retrieve information on the latest DDA findings. First, comorbidity and complication keywords pull up large numbers of PubMed studies. Second, disease is not highlighted in search results. Finally, DDA is not identified, as currently no disease-disease association extraction (DDAE) dataset or tools are available.

Objective: As there are no available DDAE datasets or tools, this study aimed to develop (1) a DDAE dataset and (2) a neural network model for extracting DDA from the literature.

Methods: In this study, we formulated DDAE as a supervised machine learning classification problem. To develop the system, we first built a DDAE dataset. We then employed two machine learning models, support vector machine and convolutional neural network, to extract DDA. Furthermore, we evaluated the effect of using the output layer as features of the support vector machine-based model. Finally, we implemented large margin context-aware convolutional neural network architecture to integrate context features and convolutional neural networks through the large margin function.

Results: Our DDAE dataset consisted of 521 PubMed abstracts. Experiment results showed that the support vector machine-based approach achieved an F1 measure of 80.32%, which is higher than the convolutional neural network-based approach (73.32%). Using the output layer of convolutional neural network as a feature for the support vector machine does not further improve the performance of support vector machine. However, our large margin context-aware-convolutional neural network achieved the highest F1 measure of 84.18% and demonstrated that combining the hinge loss function of support vector machine with a convolutional neural network into a single neural network architecture outperforms other approaches.

Conclusions: To facilitate the development of text-mining research for DDAE, we developed the first publicly available DDAE dataset consisting of disease mentions, Medical Subject Heading IDs, and relation annotations. We developed different conventional machine learning models and neural network architectures and evaluated their effects on our DDAE dataset. To further improve DDAE performance, we propose an large margin context-aware-convolutional neural network model for DDAE that outperforms other approaches.

(JMIR Med Inform 2019;7(4):e14502) doi: 10.2196/14502

KEYWORDS

deep learning; disease-disease association; biological relation extraction; convolutional neural networks; biomedical natural language processing

Introduction

Background

The origin and treatment of disease is an important research field in the life sciences, covering a wide range of research topics such as comorbidity, complication, genetic disorder, drug treatment, and adverse drug reaction. As disease is involved in many areas, new scientific findings are frequently made or updated.

Disease-disease association (DDA) is an important research topic in the biomedical domain [1-5]. The influence of one disease on others is wide ranging and can manifest in any patient. Diabetes, for example, may cause macrovascular diseases [6], such as cardiovascular disease [7] and cerebrovascular disease [8]. Treating a disease without consideration of potential DDAs may result in poor treatment outcomes. Therefore, DDAs are often a prime concern for researchers and doctors involved in drug discovery and disease treatment. Figure 1 illustrates examples of DDAs in the literature (refer to Multimedia Appendix 1 for more examples, including comorbidity, complications, general associations, and risk factors). There have been several studies attempting to generate disease connectivity networks [3-5]. However, the enormous and rapidly growing disease-related literature has not been utilized.

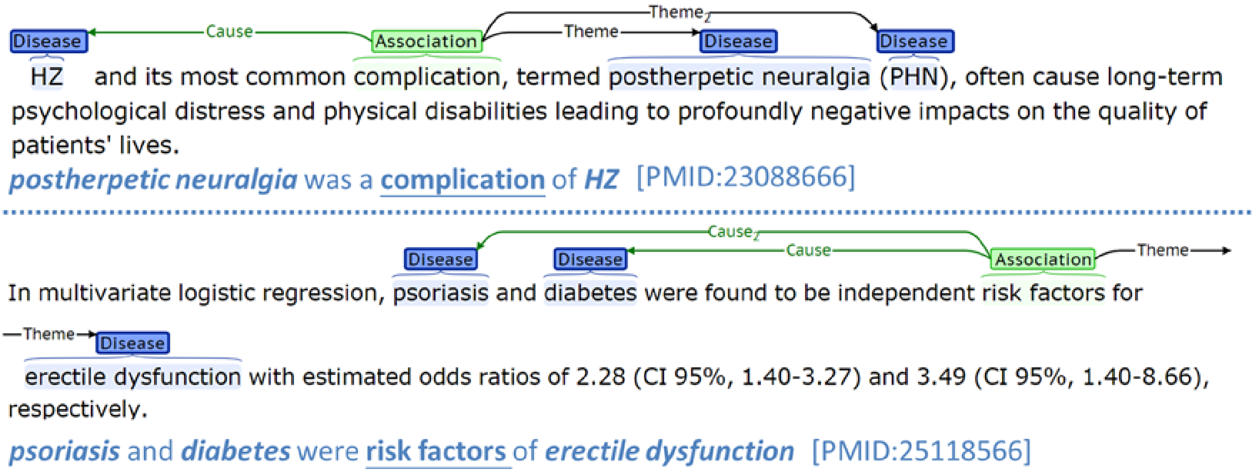
Finding DDA in the literature is a time-consuming and challenging task for researchers. First, there are huge numbers of DDA papers to sort through, and existing search engines, such as PubMed, do not mark up all relevant disease mentions in search results. Although there are text-mining tools available that could automatically identify diseases [9-11], genes [10,12,13], chemicals [14,15], and associations among them [16-22], they have not been integrated into a single interface to assist researchers in searching through the latest DDA findings. The main obstacle in creating a DDA extraction (DDAE) system is the lack of a relevant dataset. Moreover, only a few text-mining approaches [23] are suitable for extracting DDA.

In this study, we compiled a DDAE dataset consisting of 521 annotated PubMed abstracts. As it is hard for a human annotator to distinguish one DDA type from another without reading a broader context, such as a whole paragraph, we therefore annotated only 3 DDA types: positive, negative, and null associations:

1. Positive associations include comorbidity, complications, physical associations, and risk factors.
2. Negative associations are counted when the text clearly states that there is no association between 2 diseases.
3. Null associations are annotated when 2 diseases co-occur in a sentence, but no association is stated, suggested, or apparent.

In this study, we formulated DDAE as a supervised machine learning (ML) classification task in which, given a sentence containing a disease pair, the goal was to classify the pair into one of the DDA types. For classification, we employed 2 machine learning models, support vector machine (SVM) [24] and convolutional neural network (CNN) [25]. We compared different combinations of SVM and CNN to maximize performance, arriving at a novel neural network architecture, which we termed as large margin context-aware CNN (LC-CNN). LC-CNN achieved the highest F1 measure of 84.18% on our DDAE test set.

Figure 1. Disease-disease association extraction examples.


Disease
Cause
Association
Theme
Theme
Disease
Disease
HZ and its most common complication, termed postherpetic neuralgia (PHN), often cause long-term psychological distress and physical disabilities leading to profoundly negative impacts on the quality of patients' lives.
postherpetic neuralgia was a complication of HZ [PMID:23088666]
Cause
Disease
Disease
Cause
Association
Theme
In multivariate logistic regression, psoriasis and diabetes were found to be independent risk factors for
Theme
Disease
erectile dysfunction with estimated odds ratios of 2.28 (CI 95%, 1.40-3.27) and 3.49 (CI 95%, 1.40-8.66), respectively.
psoriasis and diabetes were risk factors of erectile dysfunction [PMID:25118566]

Related Work

In this section, we first review published disease annotation datasets. Then, we briefly review different methods of relation extraction in biomedical domains.

Disease Annotation Datasets

Before identifying DDAs, we have to identify diseases in the text first. Fortunately, there are many datasets for developing such disease name recognition and normalization systems. The National Center for Biotechnology Information (NCBI) disease dataset [26] is the most widely used. For instance, Leaman and Lu [9] proposed a semi-Markov model trained on an NCBI disease dataset that achieved an F1 measure of 80.7%. However, DDAs are not annotated in the NCBI dataset abstracts, limiting its usefulness for the DDAE task.

As DDAs can give insights into disease etiology and treatment, many studies focus on generating DDA networks [1-5]. For example, Sun et al [4] used disease-gene associations in the Online Mendelian Inheritance in Man [27] to predict DDAs with similar phenotypes. Bang et al [3] used disease-gene relations to define disease-disease network, and the causalities of disease pairs are confirmed through using clinical results and metabolic pathways. However, the constructed networks lack text evidence and therefore cannot be used to develop a DDAE dataset.

Xu et al [23] proposed a semisupervised iterative pattern-learning approach to learn DDA patterns from PubMed abstracts. They constructed a disease-disease risk relationship knowledge base (dRiskKB) consisting of 34,000 unique disease pairs. However, there are some limitations of dRiskKB that make it hard to use in developing DDAE systems. First, dRiskKB only provides positive DDA sentences. Owing to the lack of negative instances, it cannot be used to train ML-based classifiers. In addition, as the development of dRiskKB is based on a pattern-learning approach, it only includes DDA sentences with very simple structures and thus is not ideal for training a DDA system capable of analyzing complicated sentences.

To solve the above problems, we developed a DDAE dataset. Our dataset was different from dRiskKB in 3 aspects. First, our DDAE dataset contained positive, negative, and null DDAs. Second, it did not use patterns to annotate DDAs and therefore included DDA sentences with more complex expressions. Finally, it annotated DDAs in the entire abstract, allowing an ML-based classifier to use document-level features.

Relation Extraction

Rule-based approaches are commonly used in new domains or tasks that do not have large-scale annotated datasets. Lee et al's [28] approach is an example. They extracted protein-protein interactions (PPIs) from plain text using handcrafted dependency rules. Their approach did not require a training set, but it achieved a high precision of 97.4% on the Artificial Intelligence in Medicine (AIMed) dataset [29]. However, it was difficult for them to create rules that can extract all PPIs, and their system, therefore, achieved a low recall of 23.6%. Moreover, Nguyen et al [30] used predicate-argument structure (PAS) [31] rules to extract more general relations including PPI and drug-drug interaction. Their rules detected PPIs by examining where relation verbs and proteins are located in the spans of predicates and arguments. Their approach required less effort to design rules and was able to adapt to different relation types. Compared with Lee et al's system, it achieved a higher recall of 52.6% on the AIMed dataset but a lower precision of 30.4%.

ML-based approaches can usually achieve relatively higher performance than rule-based ones. For instance, Zhang et al [32] used hybrid feature–based and tree-based kernels implemented with SVM-LIGHT-TK [33] for PPI extraction. The feature-based kernel uses SENNA (Semantic/syntactic Extraction using a Neural Network Architecture)'s pretrained word-embedding model [34]. In the tree-based kernel configuration, the sentence dependency structure is used as input. The structure is decomposed into substructures and then transformed into one-hot encoding features for SVMs. Zhang et al's approach achieved an F score of 69.7% on the AIMed dataset, which is higher than Lee et al's 26.3% and Nguyen et al's 38.5%.

In addition to sentence-level features, document-level features are also useful in relation extraction. Peng et al [17] proposed an SVM-based approach for document-level chemical-disease relation (CDR) extraction. They used statistical features, such as whether a chemical or disease name appears in the title, to classify document-level chemical-disease pairs. By adding the features, they improved their F score from a baseline of 46.82% to 57.51% on the BioCreative V CDR dataset [35]. Our LC-CNN is partly inspired by Peng et al's [17] statistical features; our context vector adopts document-level features for sentence-level DDA classification.

Although the abovementioned feature-based approaches have made gains in many relation extraction tasks [36-38], it is difficult to find novel features to further improve performance. Several researchers are exploring deep learning approaches as a way forward. For instance, Peng and Lu [39] proposed a multichannel dependency-based CNN model (McDepCNN). McDepCNN uses 2 channels to represent an input sentence. One is the word-embedding layer, whereas the other is the head-word-embedding layer. Each embedding layer concatenates pretrained word-embedding vectors, one-hot encodings of part of speech, chunks, named entity labels, and dependency words. In PPI prediction, Peng and Lu's CNN model achieved F scores of 63.5% on AIMed and 65.3% on BioInfer.

For drug-drug interaction extraction, Lin et al [20] proposed a syntax CNN (SCNN) that integrates syntactic features, including words, predicates, and shortest dependency paths into a CNN. They trained their model with word2vec [40] and the Enju parser [31]. The Enju parser breaks the sentence into PASs, and non-PAS words or phrases are removed. The pruned sentences are then used to train the word-embedding model. Their approach achieved an F score of 68.6% on the 2013 DDIExtraction dataset.

Our LC-CNN was also inspired by Zhao et al's [20] SCNN architecture with 3 main differences. First, we replaced the log loss function with the hinge loss function. Second, SCNN uses a fully connected layer for traditional features before merging them with the CNN's output. However, LC-CNN directly merges the CNN’s output with traditional features. Finally, SCNN’s traditional features only use sentence-level information, whereas LC-CNN also uses both sentence-level and document-level features.

Methods

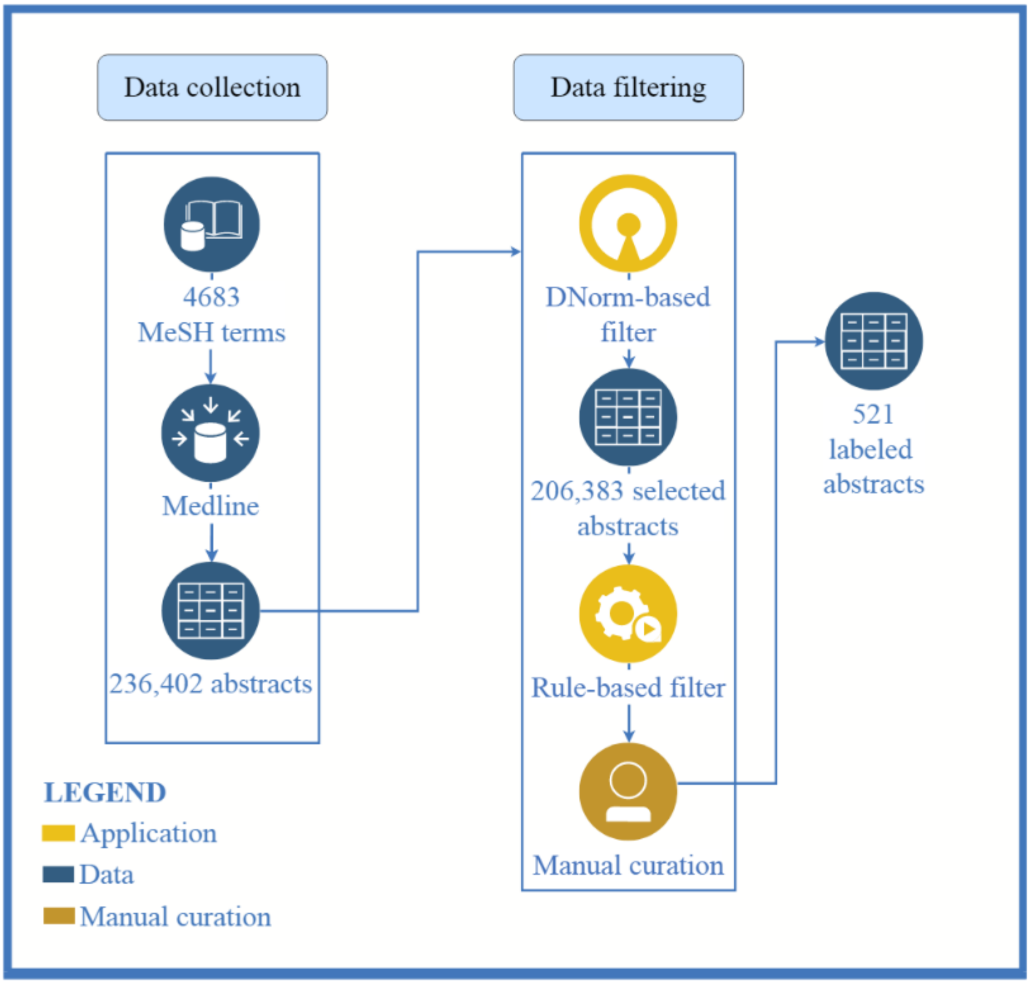
Study Process

In this section, we have first described the process of DDAE dataset construction. We then introduced our LC-CNN architecture in subsection *The Neural Network Architecture*. Further, we described each layer of LC-CNN in subsection *Composite Embedding Vector* to *Output Layer of Combined Sentence and Context Vector*. Finally, we introduced backward propagation for learning parameters of each layer.

Dataset Construction

The process of DDAE dataset construction is illustrated in Figure 2. Our DDAE dataset consisted of abstracts found in PubMed. To generate PubMed search queries related to DDA, we selected all disease nodes of the MeSH [41] tree whose tree number prefix starts with *C* and *F*, indicating diseases. We then selected any nodes related to human diseases. This produced a list of approximately 4700 disease names, which we then used to retrieve 236,000 abstracts whose titles or content contain one or more query terms.

Figure 2. Disease-disease association extraction dataset construction process. MeSH= Mesdical Subject Headings.



As some of these abstracts do not contain any DDAs, we used simple heuristic rules and a disease name recognizer/normalizer to select abstracts with a higher likelihood of containing DDAs. The process was as follows:

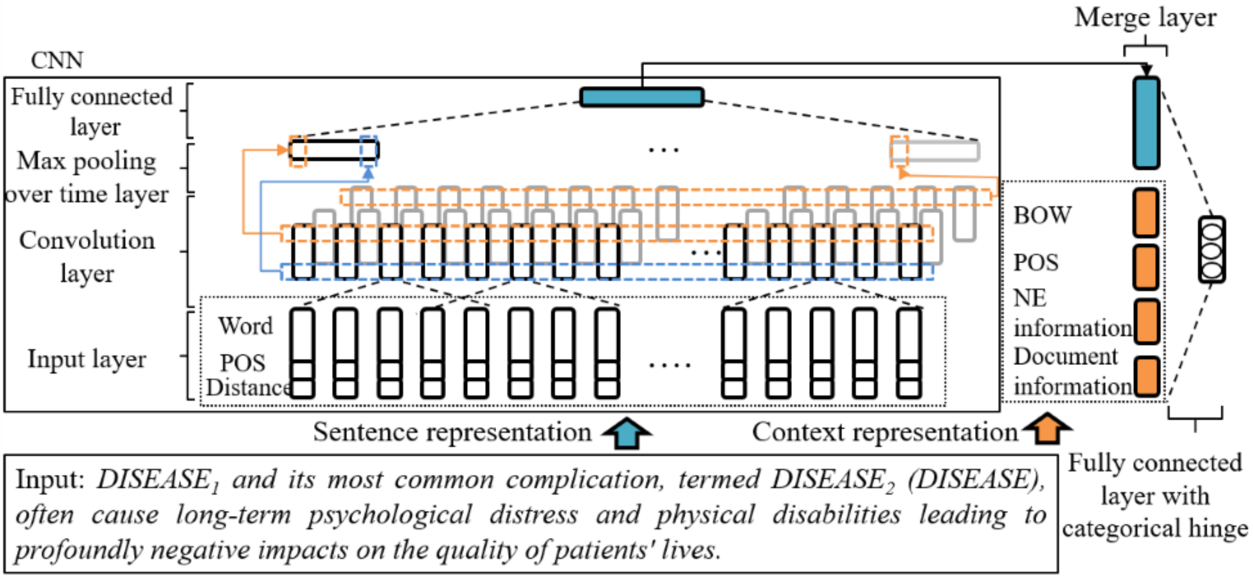
1. We selected only abstracts published from 2013 to 2017.
2. We used DNorm [42] to annotate disease mentions and their Medical Subject Heading (MeSH) IDs in these abstracts.
3. To ensure that the selected abstracts contain rich DDAs for training classifier, we removed abstracts that have fewer than 3 sentences that contain at least two different disease MeSH IDs.
4. To ensure the selected abstracts contain at least one DDA, we applied a DDA-adapted version of Lee et al’s [28] dependency tree-based relation rules and removed any abstract not matched by any rule.
   5. We randomly selected 521 abstracts from the remaining abstracts for annotation.

For the manual annotation step, we employed 2 biomedical specialists. Annotator 1 is a PhD candidate in a bioinformatics program, whereas Annotator 2 is a full-time research assistant in a hospital. Both have at least 6 years of biomedical experience. After agreeing on initial annotation guidelines (refer to Multimedia Appendix 1—Annotation Guideline), they used the brat rapid annotation tool [43] to annotate 10 abstracts and then compare results. In the first independent annotation processing, Cohen kappa value was 34%. Once both annotators agreed that all annotations that indicate consistency is satisfactory, they each annotated all remaining abstracts. Thus, each abstract was annotated independently twice. Inconsistent annotations were resolved afterward through discussion. The final Cohen kappa value was 76%.

The Neural Network Architecture

We formulated relation extraction as a classification problem in which, given a sentence containing a mention pair, the goal was to classify the pair into one of relation types. For classification, we propose an LC-CNN architecture as illustrated in Figure 3. The network is fed input in 2 forms: sentence representation and context representation (CR). Sentence representation is a n_{emb} x T matrix representing the sentence. n_{emb} and T are the length of composite embedding vector and the length of the sentence, respectively. The sentence representation uses only word embedding, part of speech (POS) encoding and Named Entity (NE) distance information, and parameters are learned through the next CNN and max-pool layers, which outputs an *m*-dimension sentence-level feature vector. The CR is a feature-rich *n*-dimension vector containing both syntactic and document-level features, such as whether the disease pair also appears in the title. Next, the *m*-dimension vector and the *n*-dimension vector are concatenated to form the final feature vector with (*m*+*n*) dimension. To compute the confidence of each relation type, the feature vector is fed into a fully connected layer, where we use a linear activation function with categorical hinge loss [44]. The output layer is a three-dimensional vector, with each dimension value representing the confidence of a predefined relation type.

Figure 3. Large margin context-aware convolutional neural network (LC-CNN) architecture. BOW: Bag of words; POS: Part of speech; NE: Named Entity.



Merge layer
CNN
Fully connected layer
Max pooling over time layer
Convolution layer
Input layer
Word
POS
Distance
BOW
POS
NE information
Document information
Sentence representation
Context representation
Input: DISEASE1 and its most common complication, termed DISEASE2 (DISEASE), often cause long-term psychological distress and physical disabilities leading to profoundly negative impacts on the quality of patients' lives.
Fully connected layer with categorical hinge

Composite Embedding Vector

In a sentence, each word is represented as a composite embedding vector, as shown in Figure 3 (or in Multimedia Appendix 2). A composite embedding vector consists of 3 parts: word embedding, POS one-hot coding, and the distance between the word and disease pair. A matrix represents a sentence. The matrix contains the composite embedding vectors in the sentence, each placed in the order in its row. The sentence matrix is a matrix of size n_{emb} x T, wheren_{emb} is the dimension of the composite embedding vector and T represents the maximum length of the sentence in the dataset.

Word Embedding

The embedding of a word is a mapping of the word to a vector of real values. Generally, the word embeddings of semantically similar words are closer together in the vector space. Word embedding learned by neural networks has been demonstrated to be able to capture linguistic regularities and patterns in language models [40]. Therefore, it is commonly used in features in popular NN approaches, such as CNN [20,39] and long-short term memory (LSTM) [19]. In general, word embeddings are learned from large corpora such as Wikipedia or PubMed. For example, Pyysalo et al [45] applied word2vec to learn word embeddings from different texts, including Wikipedia, PubMed abstracts, and PubMed Central full-text papers, and developed a word-embedding lookup dictionary. Here, we employed their dictionary to generate word embeddings.

Part of Speech

The embedding of a word is a single vector and, therefore, cannot fully represent the multiple syntactic/semantic roles of a word like *good*, which can be either an adjective or a noun. The POS feature is designed to provide syntactic information (part of speech) to help the model separate the different semantic senses of a word. We used Zhao et al’s [20] approach, in which similar POSs are assigned to the same group. We divided POSs into 11 groups, including adjectives, adverbs, articles, conjunctions, foreign words, interjections, nouns, prepositions, pronouns, punctuation, and verbs. If a word belongs to a POS group, the corresponding bit value will be 1; otherwise, it will be 0.

Named Entity Distance

Zeng et al [46] proposed the use of NE distance (position features) to improve a CNN by keeping track of how close words are to the target nouns. We adopted their NE distance in this study. The NE distance feature is a two-dimensional vector (d_1, d_2). d_1 and d_2 represent the distance (number of words) between the current word and the first and second diseases of the pair.

Context Representation Layer

Contextual information, such as pair and document information, is very useful for classification and has been widely used in previous research. The purpose of using contextual representation is to introduce traditional contextual features into a neural network architecture through simple representation. We can then apply the fully connected layer to the context vector to obtain a condensed vector that combines 2 different representations.

Here are the features used in our contextual representation (refer to Multimedia Appendix 3 for more details).

Bag of Words

Word embedding has been shown to represent abstract information about words. However, word embedding can sometimes change the original meaning of a word. For example, *not* usually appears in negative relation statements. However, in the word2vec model trained on news, the 3 words most similar to *not* are *do*, *did,* and *anymore*. This violates our intuition that *don't*, *doesn't,* and *isn't* are more similar to *not* in the relation statement. As the embedded vector words of certain words may differ in the news and biomedicine domains, we use BOW features for context vector. Our BOW features include unigram, bigram, and surrounding diseases.

Part of Speech

The POS tags are commonly used for relation extraction. We used one-hot encoding to represent each word's POS tag type.

Named Entity Information

The number of diseases is useful when classifying relations. We used 3 different features to capture information, including the following:

1. The number of tokens between disease pairs.
2. The number of diseases between disease pairs.
3. The number of diseases in the sentence.

Document-Level Information

Biological papers usually follow a certain flow to describe their experimental and scientific findings. Therefore, article structure often provides valuable information about relations. We used 2 types of document-level feature, core pair and pair location. The core pair features indicate whether the current disease is a top-3 frequent disease pair in the article. The 3 most frequent pairs are treated as 3 features. The pair location feature is used to indicate the position of the sentence containing the relation in the article. If the sentence is the article title, it usually contains the subject of the article, which might be a relation investigated in the paper. Similarly, if the sentence is the last sentence of the abstract, it may summarize the main scientific discovery of the article. We used 3 binary features to represent relation pairs that appear in the title, the first sentence of the abstract, the last sentence of the abstract.

Output Layer of Combined Sentence and Context Vector

We used m_{concat} = [sr cr] to represent the concatenation of sentence representation sr and context representation cr. The size of the vector m_{concat} is $n_{concat} = n_{sr} + n_{cr}$. We then applied a fully connected layer to m_{concat} to obtain a 3D vector *out*, each value of which refers to the confidence of a predefined category.

$$out = W_{out} \text{ x } m_{concat} + Bias_{out}$$

W_{out} is a matrix with a size of n_{out} x n_{concat} and $Bias_{out}$ is a bias matrix with a size of n_{out} x 1. n_{out} is the number of predefined categories. out is the output of this fully connected layer and is defined as matrix W_{out} multiplied by matrix m_{concat}, plus bias $Bias_{out}$ Therefore, the size of out is n_{out} x 1. *out* is the final output of the prediction, and each dimension value of out refers to the score of its predefined category. out is calculated by a linear activation function, the values of out could be R × R × R.

Backward Propagation With Large Margin Loss

We used the following parameters:

1. k weight matrices, convWf each with a size of ne x f. Here, ne is the size of the input embedding vector of a word, and f is the window size of the filter.
2. k biases, convBf, each with size of ne x 1.
3. Weight matrix Wsr with a size of nsr x npool. Here, nsr is the output dimension of sentence vector and a hyperparameter.
4. Bias Biassr with a size of nsr x 1.
5. Weight matrix wout with a size of nout x nconcat. Here, nout is the number of relation types.
6. Bias BiasmaxF with a size of nout x 1.

In forward propagation, given those parameters, we calculated out with the methods mentioned in section *The Neural Network Architecture* to *Context Representation Layer*. In backward propagation, gradient descent is used to learn these parameters through minimizing the hinge loss of out. Given a sentence and its disease-disease pair, we defined a vector y as the pair's relation label vector. *y* is a 3D vector, and each dimension value of y represents the score of one relation type. According to the definition of hinge loss [44], the value is either -1 or 1. *1* means that the pair belongs to the relation type, whereas *–1* means it does not. Therefore, one value of the 3D vector must be *1,* and the others must be *–1*. For instance, the 3 vectors <1, –1, –1>, <–1, 1, –1>, and <–1, –1, 1> indicate that 3 vectors are *Positive*, *Negative*, and *Null*, respectively. We used the hinge loss function to evaluate the loss between prediction out and its truth label *y*; a larger loss indicates a larger gap between *out* and *y*. The hinge loss function is defined as follows:

$$\text{loss}(out, y)=\text{sum}_{i=1 \text{ to} nout}(\max(1 - y_i * out_i, 0))/n_{out}$$

Here, y_i is the *i*-th dimension value of *y*. *out* is calculated by using forward propagation (sections *The Neural Network Architecture* to *Context Representation Layer*), and each dimension value of *o* refers to the prediction score of one predefined relation type. out_i is the *i*-th dimension value of out. out_i belongs to *R*. If out_i is a positive value, then the pair may be the *i*-th relation type. Otherwise, if out_i is a negative value, then the pair is less likely to be the *i*-th relation type.

In the equation, 1 is the value of the decision boundary. Ideally, $y_i * out_i$ will be larger than the decision boundary value. If y_i and out_i have the same sign, then $y_i * out_i$ will be a positive value belong to R. If $y_i * out_i$ is larger than the decision boundary value 1, then the loss(*out*, *y*) must be 0. If $y_i * out_i$ is smaller than the decision boundary value 1, then the loss(*out*, *y*) must be 1 - $y_i * out_i$ which is equal to the cost. If y_i and out_i are different signs, then $y_i * out_i$ will be a negative value ε R. Therefore, the loss(*out*, *y*) is a value greater than 1.

Given the training set

$$T=\{(x^{(i)},y^{(i)}) \mid i = 1,\ldots, N \},$$

$x^{(i)}$ is the *i*-th instance in the training set, $y^{(i)}$ is its label vector, and *N* is the number of training instances. Weight learning consists of the following optimization:

$$\text{argmin}_{convWf,\ convBf,\ Wst,\ Biassr,\ Wout,\ Biasout}\ \text{loss}(out,y)$$

Finally, mini-batch stochastic gradient descent [47] is applied to update the learned parameters in each iteration.

Results

Dataset

Currently, there are no available annotated datasets for training DDA extraction systems. To create one, we used our DDAE dataset development process, described in section *Dataset Construction*. The DDAE dataset consists of 521 annotated abstracts. After annotation, we used Cohen kappa coefficient to evaluate annotation consistency. The final kappa value is 76%, suggesting a high level of agreement.

For the experiments in this study, we divided our DDAE dataset into a training set of 400 abstracts and a test set of 121 abstracts. Before testing, we tuned the hyperparameters on one-third of abstracts randomly chosen from the training set called tuning set. Finally, our classifiers were trained on the whole training set and evaluated on the test set. A summary of the final DDAE dataset is shown in Table 1.

Table 1. Summary of disease-disease association extraction dataset.

Type	Training set, n	Test set, n	Total, n
Abstracts	400	121	521
Sentences	4820	1549	6369
Diseases	9522	2824	12,346
Total pairs	9086	2419	11,505
Positive pairs	2538	623	3161
Negative pairs	126	35	161
Null pairs	6422	1761	8183

Experiment Setup

We conducted 3 experiments to evaluate our LC-CNN. The first experiment was designed to measure the effects of different NN architectures and ML models. In the second experiment, we evaluated the effects of different approaches combining context features with NN methods. In the third experiment, we evaluated the effects of different word embeddings. The hyperparameters are listed in Multimedia Appendix 4. The performances of experiments on the tuning set can be found in Multimedia Appendix 5.

Our system is implemented on TensorFlow with Keras and runs on an Nvidia GTX 1080ti GPU. The process used in our experiments to generate the word-embedding model can be found in Multimedia Appendix 6.

Evaluation Metric

We used the F1 measure to evaluate system performance. The precision and recall are defined as given in Figure 4.

Figure 4. Precision and recall formula.

$$\text{Precision} = \frac{\text{number of correctly predicted positive and negative pairs}}{\text{number of predicted positive and negative pairs}}\P$$

$$\text{Recall} = \frac{\text{number of correctly predicted positive and negative pairs}}{\text{number of positive and negative pairs}}\P$$

$$\text{F1} - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\P$$

Experiment 1—Performance Comparison With Other Models

The performance comparison between LC-CNN and different methods is listed in Table 2. It shows the performances on the tuning and test sets. The NN models (models 1 to 3) use only sentence representation. The $CR_{\text{cross-entropy}}$ and SVM methods use only CR. $CR_{\text{cross-entropy}}$ is implemented using a single hidden fully connected layer with the context vector as its input layer, and its architecture can be found in Multimedia Appendix 7. Furthermore, we also compared LC-CNN with LSTM and bidirectional LSTM (BiLSTM) models. They have been used in many relation extraction tasks, such as those seen in the studies by Hsieh et al and Zhao et al [19,48]. In our experiment, we were surprised to find that LSTM achieved the lowest F1 measure (65.02%) on the test set among all tested models. Furthermore, we also evaluated the performance of SCNN, Bidirectional Transformers for Language Understanding (BERT) [49], and BioBERT [50]. As we would like to compare the architecture of SCNN with LC-CNN, LC-CNN and SCNN use the same sentence representation, CR, and hinge loss function. The architecture of SCNN is illustrated in Multimedia Appendix 8.

As shown in Table 2, NN models trained on the entire training set (models 1 to 3) performed worse on the test set than on the tuning set. One potential reason is that the selected hyperparameters and parameters may be less likely to find unseen data, which could cause the hyperparameters and parameters of the NN models to overfit the tuning set. This problem is especially obvious in the LSTM and BiLSTM models. In contrast, $CR_{\text{cross-entropy}}$, SVM, and LC-CNN models trained on the entire training set with context information performed better on the test set than on the tuning set.

Furthermore, as shown in Table 2, CNN and $CR_{\text{cross-entropy}}$ performed similarly on the tuning set. The F1 measures of CNN and $CR_{\text{cross-entropy}}$ were 75.35% and 75.76%, respectively. CNN's recall rate was better than $CR_{\text{cross-entropy}}$'s recall rate by 2.84%, whereas $CR_{\text{cross-entropy}}$'s precision was 3.95% higher than that of CNN. This may be because the document feature provides $CR_{\text{cross-entropy}}$ with the information on the entire document, thus causing the model to generate fewer false positive cases. As CNN does not directly encode document information, it predicts more FPs. However, as CNN does not use any particular feature to separate positive, negative, and null relation pairs, it may be able to extract potential positive and negative pairs missed by $CR_{\text{cross-entropy}}$, resulting in higher recall rates. In addition, the SVM and $CR_{\text{cross-entropy}}$ use the same input features, but SVM mainly uses large margin for learning. The result shows that the SVM implemented with LibSVM [24] outperforms the $CR_{\text{cross-entropy}}$ by an F1 measure of 2.83%. Moreover, LC-CNN is able to combine the advantages of CNN and SVM to achieve the highest precision/recall/F1 measure among the tested models and outperforms SCNN, BERT, and BioBERT by F1 measures of 3.25%, 2.06%, and 1.91, respectively.

Table 2. Performances of different models. P: Precision; R: Recall; F: F1-Measure.

Input	Model	Tuning set			Test set		
		P (%)	R (%)	F (%)	P (%)	R (%)	F (%)
SR[a]	LSTM[b]	65.53	70.15	67.76	66.13	63.95	65.02
SR	BiLSTM[c]	73.78	70.12	71.90	65.16	65.64	65.40
SR	CNN[d]	75.31	75.39	75.35	74.86	71.84	73.32
CR[e]	$\text{CR}_{\text{cross-entropy}}$	79.26	72.55	75.76	77.78	77.19	77.49
CR	SVM[f]	74.86	81.03	77.86	78.44	82.29	80.32
SR+CR	SCNN[g]	79.23	88.30	83.52	75.31	87.44	80.93
SR+CR	LC-CNN[h]	82.58	87.72	85.07	82.36	85.00	84.18
Sentence+pair	BERT	77.23	80.27	78.72	79.24	85.23	82.12
Sentence+pair	BioBERT	80.22	83.75	81.95	80.24	85.35	82.27

[a] SR: sentence representation.

[b] LSTM: long-short term memory.

[c] BiLSTM: bidirectional long-short term memory.

[d] CNN: convolutional neural network.

[e] CR: context representation.

[f] SVM: support vector machine.

[g] SCNN: syntax convolutional neural network.

[h] LC-CNN: Large margin context-aware convolutional neural network.

Experiment 2—Effect of Different Uses of Context Information

To demonstrate the advantage of integrating CNN and context information in a single LC-CNN architecture, we evaluated different ways of combining them. The performances of these combinations are shown in Table 3. There are 3 baseline models that use only either CNN or context information. Baselines 1 to 3 are $\text{CR}_{\text{cross-entropy}}$, SVM, and CNN and are used in Experiment 1. Only $\text{CR}_{\text{cross-entropy}}$ and SVM use contextual information.

SVM + CNN is an intuitive method in which the output vector of CNN is considered an additional feature vector of SVM, and its architecture is illustrated in Multimedia Appendix 9. As shown in Table 3, the F1-measure of SVM + CNN is significantly lower than that of SVM by 6.98%. One possible reason is that the CNN used in SVM + CNN is adjusted on the tuning set, so it causes the model to overfit CNN predictions, making it difficult to learn feature weights well.

We designed the LC-CNN to learn the model in a single stage. LC-CNN achieves an F1 measure of 84.18% on the test set, which is the highest score among all methods and outperform SCNN. The results showed that LC-CNN can learn CNN and context information well in a single stage.

Table 3. Performance of combined classifiers. P: Precision; R: Recall; F: F1-Measure.

Method	P (%)	R (%)	F (%)
Baseline 1 ($CR^{a}_{\text{cross-entropy}}$)	77.78	77.19	77.49
Baseline 2 (SVM[b])	78.44	82.29	80.32
Baseline 3 (CNN[c])	74.86	71.84	73.32
SCNN[d]	75.31	87.44	80.93
LC-CNN[e]	82.36	85.00	84.18
SVM+CNN (2-stage)	74.45	72.26	73.34

[a] CR: context representation.

[b] SVM: support vector machine.

[c] CNN: convolutional neural network.

[d] SCNN: syntax convolutional neural network.

[e] LC-CNN: large margin context-aware convolutional neural network.

Experiment 3—Effect of Composite Embedding Vectors on Large Margin Context-Aware Convolutional Neural Networks

In our third experiment, we evaluated the effect of different composite embedding vectors on LC-CNN (the effect of different features on LC-CNN can be found in Multimedia Appendix 10). The performance on the test set is shown in Table 4. We compared 3 different word embeddings. The word embeddings of $\text{LC-CNN}_{\text{PubMed}}$ are from Pyysalo et al [45], who learned them from Wikipedia, PubMed abstracts, and PubMed Central full texts. The word embeddings of $\text{LC-CNN}_{\text{News}}$ are learned from Google News using word2vec. In contrast, $\text{LC-CNN}_{\text{no pretrain}}$ does not use any pretrained word embeddings. Its word embeddings are treated as parameters and are learned through training $\text{LC-CNN}_{\text{no pretrain}}$ on the training set. Moreover, we also evaluated the effect of 3 different embedding features (word embedding, POS, and NE distance) by removing them individually from the $\text{LC-CNN}_{\text{PubMed}}$.

As shown in Table 4, the model with PubMed word embeddings ($\text{LC-CNN}_{\text{PubMed}}$) outperformed $\text{LC-CNN}_{\text{News}}$ and $\text{LC-CNN}_{\text{no pretrain}}$. In addition, our removal tests indicated that both POS and NE distance have strong impact on performance.

Table 4. The effect of different composite embedding vectors on large margin context-aware convolutional neural network performance. P: Precision; R: Recall; F: F1-Measure.

Method	P (%)	R (%)	F (%)
$\text{LC-CNN}^{a}_{\text{PubMed}}$	82.36	85.00	84.18
$\text{LC-CNN}_{\text{news}}$	79.80	87.36	83.41
$\text{LC-CNN}_{\text{no pretrain}}$	77.83	86.58	81.97
$\text{LC-CNN}_{\text{PubMed}}$—POS[b]	80.23	84.26	82.19
$\text{LC-CNN}_{\text{PubMed}}$—distance	77.68	87.08	82.11

[a] LC-CNN: large margin context-aware convolutional neural network.

[b] POS: part of speech.

Discussion

Large Margin Context-Aware Convolutional Neural Network Error Cases Distribution

We randomly sampled approximately 60 error cases of the LC-CNN's predictions, and their distribution is illustrated in Table 5. FP and FN denote the false positive and false negative cases, respectively. As shown in Table 5, the *symptom/subclass* is a common error category in the FPs, and it contains a ratio of 28% in the sampled error cases. The *symptom/subclass* indicates that a disease is either a subclass or a symptom of another disease in the FP/FN disease pair. For example, an FP case: "Other large-artery aneurysms, including carotid, subclavian, and *iliac artery aneurysms*$_{\text{DISEASE1}}$, have also been associated with *Marfan syndrome*$_{\text{DISEASE2}}$. --- PMID:23891252" [51].

Here, the *carotid*, *subclavian*, and *iliac artery aneurysms* are 3 *Traumatic syndrome* for *Marfan syndrome*. They are the symptoms of *Marfan syndrome*. The symptom is not included in our DDA definition. Therefore, *iliac artery aneurysms*$_{\text{DISEASE1}}$ does not have a relation with the *Marfan syndrome*$_{\text{DISEASE2.}}$ However, in this case, the keyword phrase *been associated with* makes LC-CNN predict it as positive relation, and thus results in an FP case.

In contrast with the FP cases, the FN cases are relatively sparse, and most of them cannot be categorized. For example, “CONCLUSION: *Cataract*$_{\text{DISEASE1}}$, uncorrected refractive error, and fundus diseases are ranked in the top 3 causes of moderate to severe *visual impairment* $_{\text{DISEASE2}}$ and blindness in adults aged 50 years or more in rural Shandong Province. --- PMID: 23714032” [52].

In the sentence, *Cataract* is one cause of *visual impairment*; however, the description also lists the other 2 diseases that cause *visual impairment*. For example, “it can be associated with any type of *vision loss*$_{\text{DISEASE1}}$ including that related to *maculardegeneration*$_{\text{DISEASE2}}$, *corneal disease*$_{\text{DISEASE3}}$, *diabetic retinopathy*$_{\text{DISEASE4}}$, and *occipital infarct*$_{\text{DISEASE5}}$. --- PMID:24339694” [53].

Here, the LC-CNN correctly identifies the relation between DISEASE1 and DISEASE2. However, it failed to identify the relations between DISEASE1 and the other diseases (DISEASE3, DISEASE4, and DISEASE5).

Table 5. The distribution of sampled large margin context-aware convolutional neural network error cases.

Type, category	Description	Ratio (%)
FP[a]		
Symptom/subclass	A disease is a symptom/subclass of another disease	28
Co-occur	2 diseases co-occur in the sentence	24
Negation	2 diseases are negative relation	8
Others	The error cannot be categorized	40
FN[b]		
Simple FN	There is an obvious relation keyword for disease pair	23
Negation	2 diseases are negative relation	16
Others	No obvious relation keyword, or the statements of DDA[c] are too complicated	61

[a]FP: False positive.

[b]FN: False negative.

[c]DDA: disease-disease association.

The Result of Using Automatic Annotated Disease Mentions

In our experiment, we used the manually annotated disease mentions, which may not reflect the actual performance of the fully automated DDAE task. Hence, we conducted an experiment, in which we used the TaggerOne [9], a state-of-the-art disease mention recognizer/normalizer, to annotate the disease mentions of the test set. Then we used the LC-CNN to extract DDAs from the TaggerOne-annotated test set. As the boundaries of some predicted mentions may be inconsistent with the gold mentions, we used an approximate matching to allow this. In the fully automatic process, the LC-CNN achieved a Precision/Recall/F1 measure of 75.28/55.03/63.57, respectively. The recall is significantly lower because it failed to recognize some diseases. However, the performance is reasonable but 7.08% lower than that of the semiautomatic process (using gold disease mentions).

Principal Findings

Our objective was to develop a DDAE dataset and a neural network–based approach to extract DDAs. In our experiments, the LC-CNN trained on our dataset achieved an F1 measure of 84.18%. We also compared LC-CNN with common NN models including CNN, Bi-LSTM, and SVM. The results showed that the LSTM and BiLSTM models achieved relatively lower F1 measures of 65.02% and 65.40%, respectively. This may be because the hyperparameters and parameters tend to overfit the training set. The CNN and SVM models achieved relatively higher F1 measures of 73.32% and 77.49%, respectively, but LC-CNN still outperformed all tested methods. In addition, the results showed that the 2-stage *SVM + CNN* model scored significantly lower in terms of F1 than SVM and LC-CNN by 6.98% and 10.84%, respectively. This suggests that simple methods may achieve better results than complex ones. Furthermore, in our experiments, the model with PubMed word embeddings (LC-CNN$_{\text{PubMed}}$) outperformed the LC-CNN$_{\text{News}}$ and LC-CNN$_{\text{no pretrain}}$ models, indicating that PubMed word embeddings may be more compatible with our DDAE dataset.

Conclusions

In this paper, we proposed a text-mining approach for automatically extracting DDAs from abstracts. We collected disease-related abstracts from PubMed and annotated the first publicly available DDAE dataset consisting of 521 abstracts and 3322 disease-disease pairs. Moreover, to extract DDAs, we used several different ML models, including BiLSTM, CNN, and SVM. We also evaluated the effect of combining CNN and context features. Finally, we implemented a novel neural network called LC-CNN to integrate context features and CNN through the large margin function. Our experiment results showed that LC-CNN achieved an F1 measure of 84.18%, the highest among the tested models.

Acknowledgments

The authors would like to thank the Ministry of Science and Technology, Taiwan, for financially supporting this research. This study was funded by the Ministry of Science and Technology, Taiwan, [105-2221-E-008-115-MY3] and [103-2221-E-008-044-MY3].

Conflicts of Interest

None declared.

Multimedia Appendix 1

Annotation Guideline.
[PDF File (Adobe PDF File), 764 KB-Multimedia Appendix 1]

Multimedia Appendix 2

Composite Embedding Vector.
[PDF File (Adobe PDF File), 92 KB-Multimedia Appendix 2]

Multimedia Appendix 3

Context Representation Layer.
[PDF File (Adobe PDF File), 151 KB-Multimedia Appendix 3]

Multimedia Appendix 4

Hyperparameters.
[PDF File (Adobe PDF File), 78 KB-Multimedia Appendix 4]

Multimedia Appendix 5

Performances on Tuning Set.
[PDF File (Adobe PDF File), 81 KB-Multimedia Appendix 5]

Multimedia Appendix 6

Generating Word Embedding.
[PDF File (Adobe PDF File), 133 KB-Multimedia Appendix 6]

Multimedia Appendix 7

Architecture of CRcross-entropy.
[PDF File (Adobe PDF File), 217 KB-Multimedia Appendix 7]

Multimedia Appendix 8

Architecture of SCNN.
[PDF File (Adobe PDF File), 196 KB-Multimedia Appendix 8]

Multimedia Appendix 9

Architecture of SVM + CNN.
[PDF File (Adobe PDF File), 191 KB-Multimedia Appendix 9]

Multimedia Appendix 10

Effect of Different Features on LC-CNN.
[PDF File (Adobe PDF File), 78 KB-Multimedia Appendix 10]

Multimedia Appendix 11

DDAE Dataset.

[ZIP File (Zip Archive), 864 KB-Multimedia Appendix 11]

References

1. Žitnik M, Janjić V, Larminie C, Zupan B, Pržulj N. Discovering disease-disease associations by fusing systems-level molecular data. Sci Rep 2013 Nov 15;3:3202 [FREE Full text] [doi: 10.1038/srep03202] [Medline: 24232732]
2. Liu C, Tseng Y, Li W, Wu C, Mayzus I, Rzhetsky A, et al. DiseaseConnect: a comprehensive web server for mechanism-based disease-disease connections. Nucleic Acids Res 2014 Jul;42(Web Server issue):W137-W146 [FREE Full text] [doi: 10.1093/nar/gku412] [Medline: 24895436]
3. Bang S, Kim J, Shin H. Causality modeling for directed disease network. Bioinformatics 2016 Sep 1;32(17):i437-i444. [doi: 10.1093/bioinformatics/btw439] [Medline: 27587660]
4. Sun K, Gonçalves JP, Larminie C, Przulj N. Predicting disease associations via biological network analysis. BMC Bioinformatics 2014 Sep 17;15:304 [FREE Full text] [doi: 10.1186/1471-2105-15-304] [Medline: 25228247]
5. Yang J, Wu SJ, Yang SY, Peng JW, Wang SN, Wang FY, et al. DNetDB: The human disease network database based on dysfunctional regulation mechanism. BMC Syst Biol 2016 May 21;10(1):36 [FREE Full text] [doi: 10.1186/s12918-016-0280-5] [Medline: 27209279]
6. Chawla A, Chawla R, Jaggi S. Microvasular and macrovascular complications in diabetes mellitus: distinct or continuum? Indian J Endocrinol Metab 2016;20(4):546-551 [FREE Full text] [doi: 10.4103/2230-8210.183480] [Medline: 27366724]
7. Leon BM, Maddox TM. Diabetes and cardiovascular disease: epidemiology, biological mechanisms, treatment recommendations and future research. World J Diabetes 2015 Oct 10;6(13):1246-1258 [FREE Full text] [doi: 10.4239/wjd.v6.i13.1246] [Medline: 26468341]
8. Tang B, Cao H, Wang X, Chen Q, Xu H. Evaluating word representation features in biomedical named entity recognition tasks. Biomed Res Int 2014;2014:240403-240406 [FREE Full text] [doi: 10.1155/2014/240403] [Medline: 24729964]
9. Leaman R, Lu Z. TaggerOne: joint named entity recognition and normalization with semi-Markov Models. Bioinformatics 2016 Sep 15;32(18):2839-2846 [FREE Full text] [doi: 10.1093/bioinformatics/btw343] [Medline: 27283952]
10. Zhu Q, Li X, Conesa A, Pereira C. GRAM-CNN: a deep learning approach with local context for named entity recognition in biomedical text. Bioinformatics 2018 May 1;34(9):1547-1554. [doi: 10.1093/bioinformatics/btx815] [Medline: 29272325]
11. Luo ZH, Shi MW, Yang Z, Zhang HY, Chen ZX. pyMeSHSim: an integrative python package to realize biomedical named entity recognition, normalization and comparison. bioRxiv 2018:459172. [doi: 10.1101/459172]
12. Wei CH, Kao HY, Lu Z. GNormPlus: an integrative approach for tagging genes, gene families, and protein domains. Biomed Res Int 2015;2015:918710 [FREE Full text] [doi: 10.1155/2015/918710] [Medline: 26380306]
13. Lai P, Huang M, Yang T, Hsu W, Tsai RT. Statistical principle-based approach for gene and protein related object recognition. J Cheminform 2018 Dec 17;10(1):64 [FREE Full text] [doi: 10.1186/s13321-018-0314-7] [Medline: 30560325]
14. Leaman R, Wei C, Zou C, Lu Z. Mining chemical patents with an ensemble of open systems. Database (Oxford) 2016;2016:baw065 [FREE Full text] [doi: 10.1093/database/baw065] [Medline: 27173521]
15. Tsai RT, Hsiao YC, Lai P. NERChem: adapting NERBio to chemical patents via full-token features and named entity feature with chemical sub-class composition. Database (Oxford) 2016 Oct 25;2016:baw135 [FREE Full text] [doi: 10.1093/database/baw135] [Medline: 31414701]
16. Li L, Guo R, Jiang Z, Huang D. An approach to improve kernel-based protein-protein interaction extraction by learning from large-scale network data. Methods 2015 Jul 15;83:44-50. [doi: 10.1016/j.ymeth.2015.03.026] [Medline: 25864936]
17. Peng Y, Wei CH, Lu Z. Improving chemical disease relation extraction with rich features and weakly labeled data. J Cheminform 2016;8:53 [FREE Full text] [doi: 10.1186/s13321-016-0165-z] [Medline: 28316651]
18. Ravikumar K, Rastegar-Mojarad M, Liu H. BELMiner: adapting a rule-based relation extraction system to extract biological expression language statements from bio-medical literature evidence sentences. Database (Oxford) 2017 Jan 1;2017(1):baw156 [FREE Full text] [doi: 10.1093/database/baw156] [Medline: 28365720]
19. Hsieh Y, Chang Y, Chang N, Hsu W. Identifying Protein-protein Interactions in Biomedical Literature using Recurrent Neural Networks with Long Short-Term Memory. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers). 2017 Presented at: IJCNLP'17; 2017; Taipei, Taiwan p. 240-245.
20. Zhao Z, Yang Z, Luo L, Lin H, Wang J. Drug drug interaction extraction from biomedical literature using syntax convolutional neural network. Bioinformatics 2016 Nov 15;32(22):3444-3453 [FREE Full text] [doi: 10.1093/bioinformatics/btw486] [Medline: 27466626]
21. Lai P, Lo YY, Huang MS, Hsiao YC, Tsai RT. BelSmile: a biomedical semantic role labeling approach for extracting biological expression language from text. Database (Oxford) 2016;2016:- [FREE Full text] [doi: 10.1093/database/baw064] [Medline: 27173520]
22. Hoyt CT, Domingo-Fernández D, Hofmann-Apitius M. BEL Commons: an environment for exploration and analysis of networks encoded in Biological Expression Language. Database 2018;2018:-. [doi: 10.1101/288274]
23. Xu R, Li L, Wang Q. dRiskKB: a large-scale disease-disease risk relationship knowledge base constructed from biomedical text. BMC Bioinformatics 2014 Apr 12;15:105 [FREE Full text] [doi: 10.1186/1471-2105-15-105] [Medline: 24725842]
24. Chang CC, Lin CJ. LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol 2011 Apr 1;2(3):1-27. [doi: 10.1145/1961189.1961199]

25. Krizhevsky A, Sutskever I, Hinton GE. ImageNet Classification With Deep Convolutional Neural Networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1. 2012 Presented at: NIPS'12; December 3-6, 2012; Lake Tahoe, Nevada p. 1097-1105. [doi: 10.1145/3065386]
26. Doğan RI, Leaman R, Lu Z. NCBI disease corpus: a resource for disease name recognition and concept normalization. J Biomed Inform 2014 Feb;47:1-10 [FREE Full text] [doi: 10.1016/j.jbi.2013.12.006] [Medline: 24393765]
27. Hamosh A, Scott AF, Amberger JS, Bocchini CA, McKusick VA. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic Acids Res 2005 Jan 1;33(Database issue):D514-D517 [FREE Full text] [doi: 10.1093/nar/gki033] [Medline: 15608251]
28. Lee J, Kim D, Lee S, Lee S, Kang J. On the efficacy of per-relation basis performance evaluation for PPI extraction and a high-precision rule-based approach. BMC Med Inform Decis Mak 2013;13 Suppl 1:S7 [FREE Full text] [doi: 10.1186/1472-6947-13-S1-S7] [Medline: 23566263]
29. Bunescu R, Ge R, Kate RJ, Marcotte EM, Mooney RJ, Ramani AK, et al. Comparative experiments on learning information extractors for proteins and their interactions. Artif Intell Med 2005 Feb;33(2):139-155. [doi: 10.1016/j.artmed.2004.07.016] [Medline: 15811782]
30. Nguyen NT, Miwa M, Tsuruoka Y, Chikayama T, Tojo S. Wide-coverage relation extraction from MEDLINE using deep syntax. BMC Bioinformatics 2015 Apr 1;16:107 [FREE Full text] [doi: 10.1186/s12859-015-0538-8] [Medline: 25887686]
31. Miyao Y, Tsujii J. Feature forest models for probabilistic HPSG parsing. Comput Linguist 2008 Mar;34(1):35-80. [doi: 10.1162/coli.2008.34.1.35]
32. Zhang Y, Lin H, Yang Z, Wang J, Li Y. Hash subgraph pairwise kernel for protein-protein interaction extraction. IEEE/ACM Trans Comput Biol Bioinform 2012;9(4):1190-1202. [doi: 10.1109/TCBB.2012.50] [Medline: 22595237]
33. Moschitti A. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In: Proceedings of the 17th European conference on Machine Learning. 2006 Presented at: ECML'06; September 18-22, 2006; Berlin, Germany p. 318-329. [doi: 10.1007/11871842_32]
34. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. J Mach Learn Res 2011;12:2493-2537 [FREE Full text]
35. Wei C, Peng Y, Leaman R, Davis AP, Mattingly CJ, Li J, et al. Assessing the state of the art in biomedical relation extraction: overview of the BioCreative V chemical-disease relation (CDR) task. Database (Oxford) 2016;2016 [FREE Full text] [doi: 10.1093/database/baw032] [Medline: 26994911]
36. Liu S, Tang B, Chen Q, Wang X, Fan X. Feature engineering for drug name recognition in biomedical texts: feature conjunction and feature selection. Comput Math Methods Med 2015;2015:913489 [FREE Full text] [doi: 10.1155/2015/913489] [Medline: 25861377]
37. Thomas P, Neves M, Rocktäschel T, Leser U. WBI-DDI: Drug-Drug Interaction Extraction using Majority Voting. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013). 2013 Presented at: SEMEVAL'13; 2013; Atlanta, Georgia, USA p. 628-635.
38. Thuy Phan TT, Ohkawa T. Protein-protein interaction extraction with feature selection by evaluating contribution levels of groups consisting of related features. BMC Bioinformatics 2016 Jul 25;17(Suppl 7):246 [FREE Full text] [doi: 10.1186/s12859-016-1100-z] [Medline: 27454611]
39. Peng Y, Lu Z. Deep Learning for Extracting Protein-Protein Interactions From Biomedical Literature. In: Proceedings of the Biomedical Natural Language Processing Workshop (2017). 2017 Presented at: BioNLP'17; 2017; Vancouver, Canada p. 29-38. [doi: 10.18653/v1/w17-2304]
40. Mikolov T, Yih W, Zweig G. Linguistic Regularities in Continuous Space Word Representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2013 Presented at: NAACL'13; 2013; Atlanta, Georgia.
41. Lipscomb CE. Medical Subject Headings (MeSH). Bull Med Libr Assoc 2000 Jul;88(3):265-266 [FREE Full text] [Medline: 10928714]
42. Leaman R, Dogan RI, Lu Z. DNorm: disease name normalization with pairwise learning to rank. Bioinformatics 2013 Nov 15;29(22):2909-2917 [FREE Full text] [doi: 10.1093/bioinformatics/btt474] [Medline: 23969135]
43. Stenetorp P, Pyysalo S, Topic G, Ohta T, Ananiadou S, Tsujii J. BRAT: A Web-based Tool for NLP-Assisted Text Annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics. 2012 Presented at: EACL'12; April 23-27, 2012; Avignon, France p. 102-107.
44. Tang Y. Deep learning using linear support vector machines. arXiv preprint arXiv 2013:13060239 [FREE Full text]
45. Pyysalo S, Ginter F, Moen H, Salakoski T, Ananiadou S. editors. Distributional Semantics Resources for Biomedical Text Processing 2013.
46. Zeng D, Liu K, Lai S, Zhou G, Zhao J. Relation Classification via Convolutional Deep Neural Network. In: Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers. 2014 Presented at: COLING'14; August 23-29, 2014; Dublin, Ireland p. 2335-2344.

47. Li M, Zhang T, Chen Y, Smola A. Efficient Mini-Batch Training for Stochastic Optimization. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014 Presented at: KDD'14; August 24-27, 2014; New York, New York, USA p. 661-670. [doi: 10.1145/2623330.2623612]
48. Miwa M, Bansal M. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016 Presented at: ACL'16; August 7-12, 2016; Berlin, Germany p. 1105-1116. [doi: 10.18653/v1/p16-1105]
49. Devlin J, Chang MW, Lee K, Toutanova K. Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv 2018:181004805 [FREE Full text]
50. Lee J, Yoon W, Kim S, Kim D, Kim S, So C, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 2019 Sep 10. [doi: 10.1093/bioinformatics/btz682] [Medline: 31501885]
51. Awais M, Williams DM, Deeb GM, Shea MJ. Aneurysms of medium-sized arteries in Marfan syndrome. Ann Vasc Surg 2013 Nov;27(8):1188.e5-1188.e7. [doi: 10.1016/j.avsg.2012.12.002] [Medline: 23891252]
52. Li Y, Bi HS, Wang LH, Wang T, Yang SY, Liu LP, et al. [Causes of moderate to severe visual impairment and blindness in population aged 50 years or more in rural Shandong province]. Zhonghua Yan Ke Za Zhi 2013 Feb;49(2):144-150. [Medline: 23714032]
53. Zhang J, Waisbren E, Hashemi N, Lee AG. Visual hallucinations (Charles Bonnet syndrome) associated with neurosarcoidosis. Middle East Afr J Ophthalmol 2013;20(4):369-371 [FREE Full text] [doi: 10.4103/0974-9233.119997] [Medline: 24339694]

Abbreviations

BERT: Bidirectional Transformers for Language Understanding
BiLSTM: bidirectional long-short term memory
BOW: bag of words
CDR: chemical-disease relation
CNN: convolutional neural network
CR: context representation
DDA: disease-disease association
DDAE: disease-disease association extraction
dRiskKB: disease-disease risk relationship knowledge base
LSTM: long-short term memory
McDepCNN: multichannel dependency-based convolutional neural network
MeSH: Medical Subject Headings
ML: machine learning
NCBI: National Center for Biotechnology Information
PAS: predicate-argument structure
POS: part of speech
PPI: protein-protein interaction
SCNN: syntax convolutional neural network
SVM: support vector machine

Edited by C Lovis; submitted 26.04.19; peer-reviewed by CH Wei, L Zhang, G Lim, B Polepalli Ramesh; comments to author 31.05.19; revised version received 26.07.19; accepted 11.08.19; published 26.11.19

Please cite as:
Lai PT, Lu WL, Kuo TR, Chung CR, Han JC, Tsai RTH, Horng JT
Using a Large Margin Context-Aware Convolutional Neural Network to Automatically Extract Disease-Disease Association from Literature: Comparative Analytic Study
JMIR Med Inform 2019;7(4):e14502
URL: http://medinform.jmir.org/2019/4/e14502/
doi: 10.2196/14502
PMID: 31769759