algorithms

MDPI

Article

Contradiction Detection with Contradiction-Specific Word Embedding

Luyang Li, Bing Qin * and Ting Liu

Research Center for Social Computing and Information Retrieval, School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China; lyli@ir.hit.edu.cn (L.L.); tliu@ir.hit.edu.cn (T.L.)
* Correspondence: qinb@ir.hit.edu.cn; Tel.: +86-186-8674-8930

Academic Editor: Toly Chen
Received: 18 January 2017; Accepted: 12 May 2017; Published: 24 May 2017

Abstract: Contradiction detection is a task to recognize contradiction relations between a pair of sentences. Despite the effectiveness of traditional context-based word embedding learning algorithms in many natural language processing tasks, such algorithms are not powerful enough for contradiction detection. Contrasting words such as "overfull" and "empty" are mostly mapped into close vectors in such embedding space. To solve this problem, we develop a tailored neural network to learn contradiction-specific word embedding (CWE). The method can separate antonyms in the opposite ends of a spectrum. CWE is learned from a training corpus which is automatically generated from the paraphrase database, and is naturally applied as features to carry out contradiction detection in SemEval 2014 benchmark dataset. Experimental results show that CWE outperforms traditional context-based word embedding in contradiction detection. The proposed model for contradiction detection performs comparably with the top-performing system in accuracy of three-category classification and enhances the accuracy from 75.97% to 82.08% in the contradiction category.

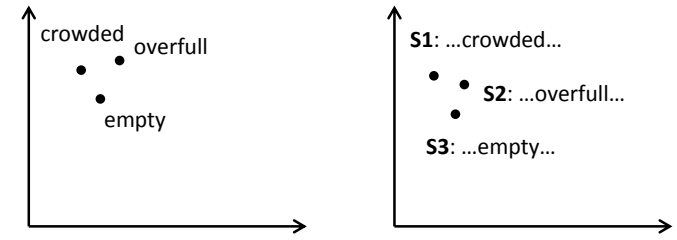
Keywords: contradiction detection; word embedding; training data generation; neural network

1. Introduction

Contradiction is a kind of semantic relation between sentences. Contradiction occurs when sentences are unlikely to be correct at the same time [1]. For example, the contradiction happens between the sentence pair "Some people and vehicles are on a crowded street" and "Some people and vehicles are on an empty street". Contradiction detection aims to recognize the contrasting meanings between two sentences [2]. Contradiction detection is helpful in many fields of natural language processing, such as information integration [3], inconsistency discovery [4,5] and sarcasm detection [6].

Contradiction detection can be regarded as a classification problem. Traditional approaches build classifiers and design effective features to improve classification accuracy. However, feature designing relies on professional knowledge and hardly captures latent semantic features. For the contradiction detection task, an effective feature learning approach is to learn the semantic relation representation from input texts.

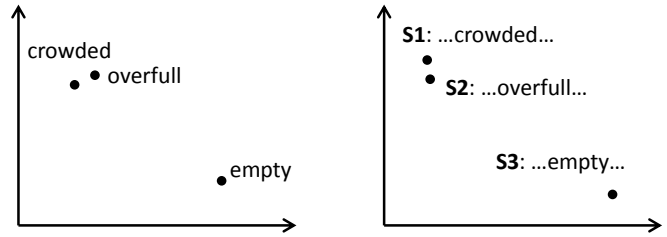
The representation learning of semantic relation is based on word embedding. However, word embedding is challenging to employ in contradiction detection because of the distributional semantics hypothesis. Traditional context-based word embedding learning algorithms typically map words with similar contexts into closed vectors. That means the words with contrasting meanings will be mapped into close vectors as long as they share a similar context [7,8], as shown in Figure 1a. The close vectors of contrasting words will lead to similar representation of contradictory sentences, as shown in Figure 1b. For a contradiction detection task, the ideal situation is that the vectors of contrasting words remain separate from each other, as shown in Figure 2a. The appropriate mapping of contrasting words will lead to a high discrimination of contradictory sentences, as shown in Figure 2b.



(a) TWE maps synonyms and contrasting words closely

(b) Sentence representation with TWE maps paraphrase and contradiction closely

Figure 1. Traditional context-based word embedding (TWE) maps contrasting words or sentences into close vectors. In (**b**) , the three sentences are, "Some people and vehicles are on a **crowded** street", "Some people and vehicles are on a **overfull** street" and "Some people and vehicles are on an **empty** street".



(a) CWE separates contrasting words

(b) Sentence representation with CWE separates contradictions

Figure 2. Contradiction-specific word embedding (CWE) separates contrasting words or sentences in the new semantic space. In (**b**), the three sentences are same as in Figure 1.

To address the issue, there are some recent methods [7–10] currently being investigated. Mrksic et al. [8] apply lexical knowledge such as WordNet [11] and PPDB [12] (Paraphrase Database) (http://www.cis.upenn.edu/~ccb/ppdb/) to revise word embedding by using 12,802 antonym pairs and 31,828 synonymy pairs. The methods of Chen et al. [7] and Liu et al. [10] use WordNet and Thesaurus to get more antonym pairs and synonym pairs as the semantic constraints. These methods can get exact antonym pairs in these lexical resources; however, the number of antonym pairs is limited. In addition, antonym pairs are just part of contrasting word pairs, and many other contrasting word pairs such as "shelve" and "pass" can not be obtained from the lexical resources. Schwartz et al. [9] use patterns such as "from X to Y" and "either X or Y" to extract antonyms from Wikipedia. However, the pattern based methods would meet the data sparsity problem. In the contradiction detection task, the pairs of sentences with contradiction relations always own contrasting words which are hardly discriminated by using antonym-based word embedding learning algorithms.

We present a method to construct a large corpus of contrasting pairs, including word pairs and phrase pairs. The large-scale corpus of contrasting pairs are generated from PPDB [12] and WordNet [11] automatically. Through our method, we obtain 1.9 million contrasting pairs and 1.6 million paraphrase pairs, that are one hundred times the size of the corpus Mrksic et al. [8] used. Although the automatically-generated corpus contains noises, we argue that it is effective enough to be leveraged as task-specific supervisions to learn CWE. Based on the corpus, we develop a neural network to learn contradiction-specific word embedding (CWE). In the aim of separating contrasting words in an embedding space, CWE is learnt based on the pairs with paraphrase relation or contradiction relation. The model for learning CWE is optimized by minimizing the semantic gap between paraphrase pairs and maximizes the gap between contradiction pairs.

To detect contradiction relation from sentence pairs, we develop a semantic relation representation learning model, and incorporate CWE to detect contradiction. We run experiments on benchmark datasets from SemEval 2014 [13]. The experiment results show that the proposed method with CWE performs comparably with top-performing systems in terms of overall classification accuracy. Specifically, it outperforms in terms of accuracy in the contradiction category.

The following statements present the major contributions of this work:

- We present a data construction method to generate a large-scale corpus for training contradiction-specific word embedding. The corpus consists of millions of contrasting pairs, which bring more guidance than prior resources with tens of thousands of antonym pairs.
- Based on the large-scale corpus with contrasting pairs, we develop a neural network tailored for learning contradiction-specific word embedding.
- We apply contradiction-specific word embedding in a semantic relation representation learning model to detect contradiction, and the accuracy of contradiction category outperforms the state-of-the-art method on benchmark data set by 6.11%.

2. Contradiction-Specific Word Embedding

A neural-network based method tailored for learning contradiction-specific word embedding (CWE) is presented in this section. We first construct a training corpus which consists of large-scale contrasting pairs. The details of data construction and data analysis are described in Section 2.1. The architecture of the neural network for learning CWE is presented in Section 2.2.

2.1. Corpus Construction for Learning CWE

Contradiction-specific word embedding learning needs a large-scale labeled corpus, which consists of contrasting pairs. The intuitive idea is to acquire them from lexical resources, such as WordNet. However, existing lexical resources only contain antonyms which are a small part of contrasting words. Actually, in the real world, most contrasting pairs are not antonyms. For example, in case (B) in Figure 3, the words "passed" and "shelved" are not antonyms; however, they have contrasting meaning in the sentences. The existance of contrasting word pairs or phrase pairs will lead to contradictory sentences, as shown in Figure 3 case (A) and (B). Thus, constructing a corpus consisting of a large number of contrasting pairs, especially containing non-strict antonyms is deemed as a necessary and helpful step in learning CWE.

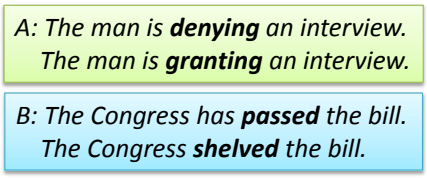


Figure 3. Examples of contradiction sentence pairs. The contradictory parts in case (A) are antonyms, which are non-strict antonyms in case (B).

We use a large paraphrase corpus PPDB to generate a contradiction corpus. The paraphrase corpus consists of word-level and phrase-level paraphrase pairs. We use WordNet to lookup the antonyms of the words in the paraphrase pair. When there is an antonym of the word in the paraphrase pair, we substitute the original word by the antonym. By this method, we can construct a corpus with contrasting pairs. Some examples are shown in Table 1. The corpus comprises about 1.9 million contradiction pairs and 1.6 million paraphrase pairs. The method generates a few noises because of polysemy; however, the corpus effectively provides supervised signals to learn word embedding.

Table 1. Examples of contrasting pairs generated from paraphrase pairs.

Paraphrase	Contradiction
something unimpressive & something strong	something unimpressive & something strong something impressive & something weak
we are on the same side & we support you	we are on the opposite side & we support you

2.2. CWE Learning Model

We present a feedforward neural network model to learn CWE. Our idea is to use large-scale contradiction pairs and paraphrase pairs to guide the embedding learning to better distinguish the contrasting words. Figure 4 illustrates the architecture of the proposed model, which mainly comprises five layers, namely lookup->linear->tanh->sum->linear. The input of the model is a pair of phrases, and the output is a two-dimension vector which consists of the confidence scores of contradiction category and paraphrase category. Through the model, we aim to revise the embeddings for the contradiction detection task.

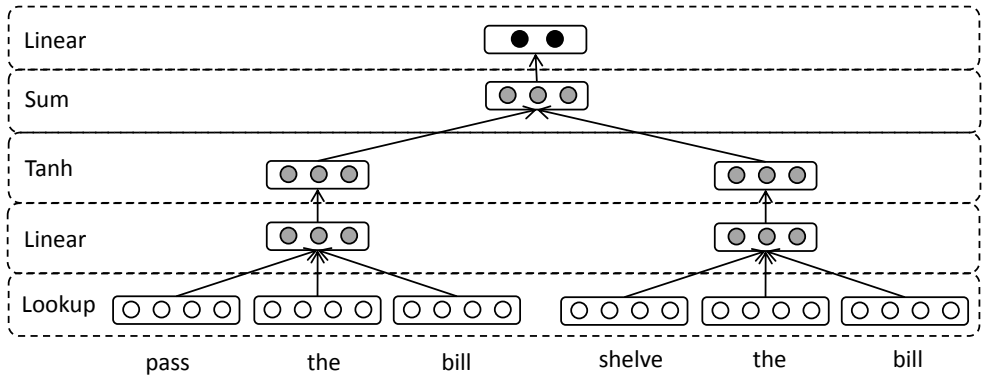


Figure 4. Neural network to learn contradiction-specific word embedding.

We solve the classification problem through a ranking objective function. The output vector o is a two-dimensional vector, which comprises the predicted scores of contradiction category and paraphrase category. The condition of updating parameters is shown in Equation (1) where m_δ is the margin, t is the target category, t^* is another category, o_t is the confidence score of t category, and o_{t^*} is the confidence score of t^* category.

$$o_t < o_{t^*} + m_\delta \tag{1}$$

We use hinge lossfunction shown as Equation (2), which is used for "maximum-margin" classification. The goal is to maximize the final score of the target category t.

$$Loss_{CWE}(s_1, s_2) = max(0, m_\delta - o_t + o_{t^*}) \tag{2}$$

2.3. Technical Specification and Training Configurations

Word vectors are initialized through existing trained embedding "Glove" [14] and are updated at each iteration. We also attempt to randomly initialize vectors as well. Random initialization does not perform as well as the initialization by the existing trained embedding.

We apply Sum composition function to do the composition and present the semantic relation between two input phrases. Assuming that two vectors y_1, y_2 are the representation of the input phrases, the vector z represent the semantic relation between y_1 and y_2. We evaluate four composition

functions in the experiment, which are Absolute Difference Equation (3), Minus Equation (4), Sum Equation (5) and Concatenation (6). The Sum composition function is verified to be most effective in representing contradictory relation, and is adopted in the contradiction detection experiment.

$$AbsoluteDifference : z = \|y_1 - y_2\| \tag{3}$$

$$Minus : z = y_1 - y_2 \tag{4}$$

$$Sum : z = y_1 + y_2 \tag{5}$$

$$Concatenation : z = [y_1, y_2] \tag{6}$$

We use stochastic gradient descent (SGD) as the parameter optimization approach. Most phrases in the corpus comprise less than five words, and the window size is finally fixed to five in the word embedding learning model. The embedding length is 50. The hidden layer length is 20. The learning rate is 0.001. The margin is 0.2.

3. Contradiction Detection

Contradiction is a type of semantic relation. We present a neural network model to learn the semantic relation representation from each pair of sentences by using CWE. The semantic relation representation serves as features, which are generated automatically from the proposed model rather than from deep syntactic analysis and feature designing in the previous methods.

3.1. Neural Network Architecture

In detecting contradiction, the global semantic features and local semantic features are both important. The global features refer to the semantic meaning of the sentence which is relevant in terms of the word order. Two sentences which consist of similar words in different word orders may have different meanings. We use the representation of sentence-level semantic relation to capture global semantic features. The local features explore the semantic relation between unaligned phrases from the pair of sentences. The unaligned phrases always contain contrasting meanings in the contradictory sentences. For example, the phrases "passed" and "shelved" are unaligned phrases in the sentence pair (B) in Figure 3.

A CNN-based (Convolutional Neural Network) model is exploited to learn the global and local semantic relation from input sentences, as shown in Figure 5. The architecture is extended from the model of learning CWE, and comprises six layers: lookup layer, convolutional layer, average-pooling layer, tanh layer, composition layer, and softmax layer. The pair of sentences and pair of unaligned phrases are the input of the model. Through the first four layers, sentences and phrases are all mapped into corresponding vectors in the same semantic space. Through the composition layer, the sentence-level and phrase-level semantic relation representations are generated and concatenated along with three shallow features. A softmax layer is adopted as a classifier.

Cross-entropy is used as a loss function. Given training data (T), the loss function to learn the classifier is expressed by Equation (7), where $y^{(k)}$ denotes the probability of the gold signal to the kth sample in the data set.

$$Loss_C = -\sum_{k=1}^{T} y^{(k)} \log p(y^{(k)}|x^{(k)}, \theta) + (1 - y^{(k)}) \log(1 - p(y^{(k)}|x^{(k)}, \theta)) \tag{7}$$

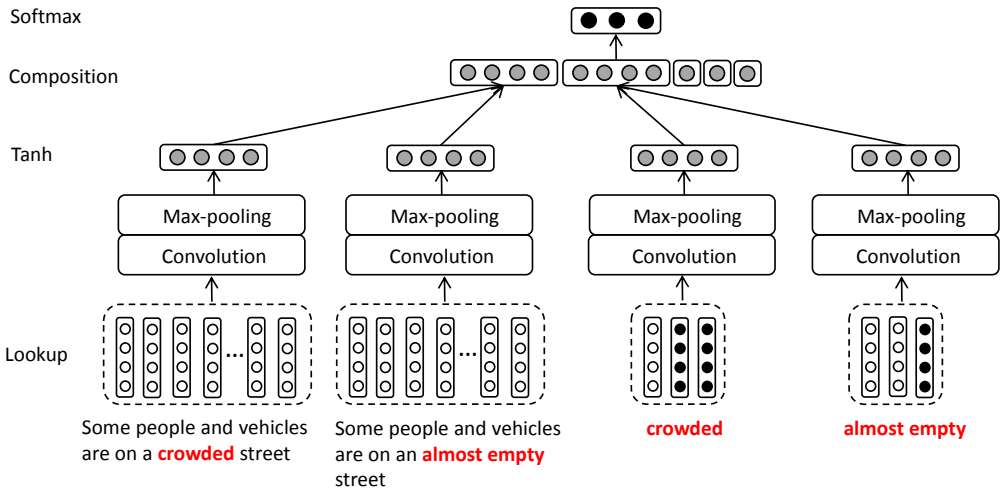


Figure 5. A feed forward neural network model for contradiction detection. The left part of the model learns the representation of the semantic relation from the sentences. The right part learns from the unaligned phrases which are obtained by removing overlapping words from sentences. The vectors are concatenated along with three shallow features in the composition layer. The shaded vectors serve as padding when the unaligned phrased are insufficiently long.

3.2. Shallow Features

In order to enhance the capability of classification, we exploit three shallow features as summarized in Table 2. By observing the training set, contradictory sentence pairs are mostly because of negations, except in the case of antonyms. If the number of negation words in the pair of sentences is odd, the semantic relation will have a high possibility of being contradiction.

Table 2. Summary of features used in contradiction detection.

Feature	Description
Negation	Odd number of negation words is deemed to be an indicator of negation existing between a pair of sentences
Difference of word order	The difference of word orders between overlapping words in a sentence pair
Unaligned word number	Average number of unaligned words after removing overlapping words

The word order feature and unaligned word number feature help to recognize entailment relation between two sentences. A large value of the unaligned number roughly indicates that the relation is entailment. In the composition layer, three extra features are directly concatenated with the sentence-level and phrase-level semantic relation representations as the input of softmax layer.

The parameters used in the contradiction detection experiment are set as follows. The lengths of embedding and the hidden layer vectors are both 50. The learning rate is 0.1. The window size is set to 7. The effect of the window size in the proposed method is experimentally studied. The result shows that the accuracy has a peak value when the window size is set to 7, which is applied in the experiment.

4. Experiments

We conduct experiments to show the effectiveness of CWE by incorporating it in a semantic relation learning model to detect contradiction. We make two comparison experiments. One is to evaluate the effectiveness of the proposed model for contradiction detection, and another is made between different embeddings for evaluating the effectiveness of CWE. We make an analysis of the corpus which is used to learn CWE.

4.1. Data Set for Contradiction Detection

Contradiction detection suffers from its lack of a gold standard dataset. In this study, we verify the effectiveness of CWE on the benchmark dataset of a textual entailment recognition task which consists of data with contradiction, entailment or neutral relation. The dataset is from task 1 in SemEval 2014 [13]. The distribution of data is shown in Table 3. The dataset is imbalanced, and the ratio among the contradiction, entailment, and neutral categories is roughly 1:2:4 not only in the training dataset but also in the trial and test datasets.

Table 3. Statistics of the dataset in SemEval 2014 task1.

	Train	Trial	Test
Contradiction	665	74	720
Entailment	1300	144	1414
Neutral	2536	282	2793
All	4501	500	4927

4.2. Baseline Methods

We compare our method with the following baseline methods:

- Illinois-LH: This method is the top-performing system [15] in Task 1 in SemEval 2014. It uses a MaxEnt model and makes deep semantic and syntactic analyses to manually gather features. The features are extracted based on the analysis of the distributional and denotational similarities, word alignment, negation, hypernym, hyponym, synonym, and antonym relations.
- TreeRNN: The full name of the method is tree-structured recursive neural networks. The method uses syntax tree of the sentences as the structure of the recursive neural network. TreeRNN is applied to composite the representations of words or phrases layer by layer according to the syntax tree [16].
- TreeRNTN: The full name of the method is tree-structure recursive neural tensor networks. The struture of the method is syntax tree. The method uses a recursive neural tensor network to do the composition in each layer of the syntax tree of the sentences [16]. This work is presented by Standford NLP group, which aims to verify the effectiveness of the tree-structure recursive neural network or tensor network in identifying logical relationships such as entailment and contradiction.
- SVM (Support Vector Machine): As a good classifier in text classification problems, SVM is utilized to solve the classification problem with CWE and shallow features as a baseline method. CWE is applied to compose two semantic features, which are the semantic relation features between whole sentences and unaligned phrases. First, the representations of the sentences and phrases are computed by averaging the embedding of all words. Second, the semantic relation features are generated by a composition function between sentence representations or unaligned phrase representations.
- LSTM-RNN: RNN (Recurrent Neural Network) can model the input sequences with time series. However, the error gradients vanish exponentially quickly with the size of the time lag in these methods [17]. To solve the problem, Hochreiter first proposed Long Short-Term Memory (LSTM) to learn the representation of the data with long distance [18]. Currently, as a popular neural network model, LSTM-RNN has been verified to be an effective or even the state-of-the-art method in many NLP tasks [19–23].

4.3. Results and Analysis

We first compare different methods for contradiction detection task, then make a comparison among different embedding by incorporating in our model.

Given a pair of sentences, we aim to predict the correct semantic relation between them. The semantic relations include: entailment, contradiction and neutral. The ideal results are that the

model can improve the accuracy of identifying the contradiction category on the premise that the three-category accuracy is at least comparable with the top-performing system.

We compare the proposed method with baseline methods on test data, and the results are shown in Table 4. The comparison results show that our method gains the highest accuracy on the contradiction category. Illinois-LH system gains the best result on the average accuracy of three categories. Our method has a comparable result to the Illinois-LH system on the average accuracy of all categories. The state of the art method, Illinois-LH, applied feature engineering and a MaxEnt model to do the task. Specifically, it made deep semantic and syntactic analyses and recognized antonyms by lexical resources. However, most of contradictory words or phrases in the dataset are not antonyms and are hard to be recognized by lexical resources. Our method uses a neural network based model to learn the representation of the semantic relation between input sentences, and treat the representation as features in the classification. The model incorporates the contradiction-specific word embedding. The advantage of our method is the capability to recognize the contradictory meanings between input sentences. The model consists of two parts, which respectively represent the semantic relation between pairs of sentences and the semantic relation between pairs of unaligned phrases among the sentence pairs. The designing of the model also benefits the contradiction detection task.

Table 4. The results of contradiction detection on the test dataset. Contradiction-specific word embedding (CWE) can be fine-tuned during training the model, CWE* represents the fine-tuned CWE. Acc(all) stands for the average accuracy over three categories and Acc(contra) stands for the accuracy of the contradiction category.

Model	Test Data	
	Acc(all)	Acc(contra)
Illinois-LH	**84.43**	75.97
TreeRNN	74.89	69.86
TreeRNTN	76.88	71.00
SVM:shallow features	76.46	72.92
SVM:CWE + shallow features	82.95	77.22
LSTM-RNN:CWE	56.07	61.39
LSTM-RNN:CWE + shallow features	75.79	70.69
CNN:CWE + shallow features	82.60	80.28
CNN:CWE* + shallow features (All)	84.27	**82.08**

TreeRNTN outperforms the LSTM-RNN models; however, it cannot beat other methods. The results of the SVM methods show that CWE is significantly beneficial, not only in detecting contradiction but also in the three-category classification. The LSTM-RNN model has a more than 90% accuracy on the training data and 76% in the test data. LSTM-RNN does not perform very well on the current dataset because of overfitting on the training data. LSTM-RNN has more parameters than other baseline methods and the proposed method, which needs a larger training dataset to avoid the overfitting problem.

Our proposed method is used by incorporating CWE and shallow features. The accuracy of the contradiction category is 80.28%, which is far higher than 75.97% of the Illinois-LH system. When we fine-tune CWE during training the proposed model, the average accuracy of all categories and the accuracy of the contradiction category both improve. The final accuracy on the contradiction category is 82.08%.

The effectiveness of CWE and three shallow features is analyzed in Table 5. The accuracies clearly decrease when CWE is abandoned. This shows the effectiveness of CWE in representing semantic relations. When we abandon CWE, the model actually turns to be a softmax classifier with shallow features. Thus, the comparison between CWE and other features just show the effectiveness of the proposed model to learn semantic relation representation of the sentence pairs. Because the negation phenomenon has a high coverage in the whole dataset, the negation feature is also useful. The two

features, which are the differences of the word order and the unaligned word number, are both helpful in three-category classification, but play a minor role in identifying contradiction category.

Table 5. The comparison of the effectiveness among CWE and shallow features.

Model	Test Data	
	Acc(all)	Acc(contra)
CNN:CWE* + shallow features (All)	84.27	**82.08**
CNN:All − CWE	76.44	78.47
CNN:All − negation	78.02	76.39
CNN:All − relative distance	80.05	77.92
CNN:All − unaligned number	80.47	79.30

CWE and task-independent word embedding such as Glove are compared in the contradiction detection task on the test dataset, as shown in Table 6. The proposed model is utilized by incorporating shallow features for comparison. We also use Mrksic's embeddding [8] to make a comparison with CWE in the same model. We aim to verify two things, which are: the necessity of learning a contradiction-specific word embedding for contradiction detection task and the advantage of our method in learning embedding. The results show that CWE performs better than Glove and Mrksic's embedding in a contradiction detection task.

Table 6. Accuracy on the test dataset with different embeddings.

Embedding	Acc(all)	Acc(contra)
Glove	80.07	79.44
Mrksic's embedding	81.99	80.00
CWE	84.27	82.08

4.4. Effect of Window Size

The effect of the window size in the proposed method for the contradiction detection task is experimentally studied. In the task 1 of SemEval 2014, there is no development set. The systems participating in the task used trial data to tune parameters. Thus, we tune parameters of the proposed model on trial data. In Figure 6, the accuracy is varied by different values of the window size. The result shows that the accuracy has a peak value when the window size is set to 7, which is applied in the experiment.

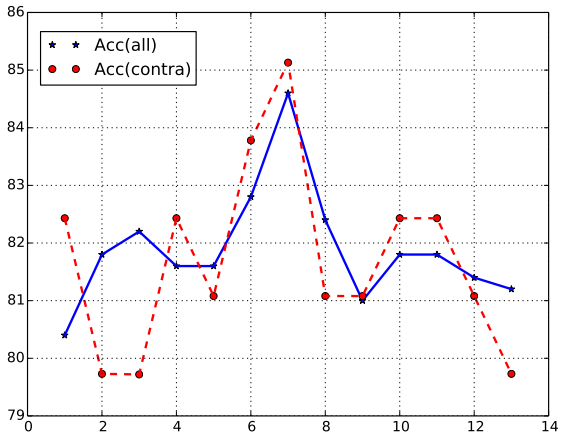


Figure 6. Effect of window size in the contradiction detection experiment.

4.5. Effect of Composition Functions

In the CWE learning model and the semantic relation representation learning model, the composition function is used to composite the vectors of input phrases or sentences. A comparison experiment of different composition functions is made on an antonym dataset. Because antonym is a special type of lexical-level contradiction relation, we gather 6335 word pairs from WordNet which are antonyms or synonyms. The dataset has a balanced ratio of positive and negative samples. The ratio of training data and test data is 2:1.

The semantic relation of each pair of words is represented by one of the composition functions in Figure 7. The representation of the semantic relation can be used as features, and be incorporated in SVM. Figure 7 shows the classification result. As we can see, the sum operation performed best, which is adopted in the contradiction detection experiment.

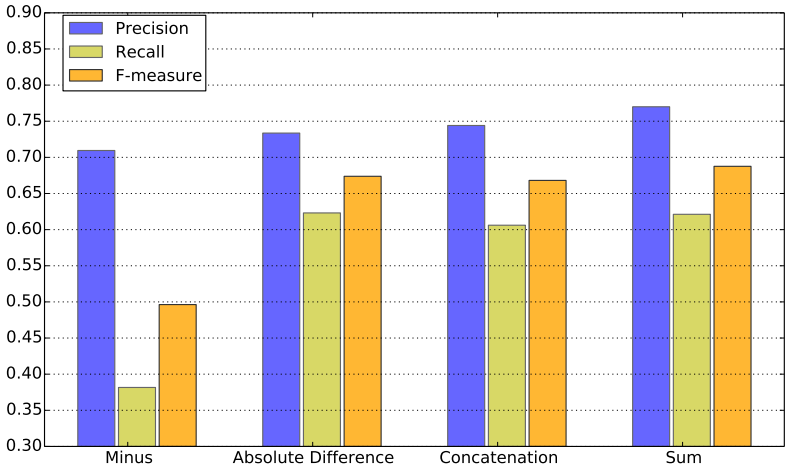


Figure 7. Comparison experiment of four composition functions in compositing semantic relation between sentences.

4.6. Analysis of the Corpus for Learning CWE

The corpus comprises about 1.9 million contradiction pairs and 1.6 million paraphrase pairs. The examples are shown in Table 7. We can see some examples are indeed contrasting, such as "ingratitude" and "loyalty", "inconsequence" and "rationality", etc. Our automatically generating contradiction method can not only capture the nominal contrasting word pairs but can also generate contrasting words cross different POSs (Part of Speech), such as "inconspicuous" and "utter". It is normal in the real world that an adjective and a verb share contrasting meanings. However, the contradiction relation between them is hard to be captured by lexicons. Because of the polysemy and the situation of individual words with multiple POSs, there are several examples without contrasting meanings. Nevertheless, the examples with opposite polarities are also meaningful for the research of contradiction detection, such as "meagerly" and "gratefully". The same phenomena show in the examples of contrasting phrases in Table 7.

To estimate the accuracy of generating contradiction corpus, we randomly choose a thousand samples and judge the correctness of each generated contradiction pair artificially by three persons. We use a voting method to annotate the labels of the samples based on the three personal judgments. When no less than two persons affirm the contradiction relation of a sample, the sample will be targeted as a correct sample (namely a real contradictory pair). Otherwise, it will be targeted as a wrong sample. Finally, the accuracy of constructing contradictory pairs is 71% in the randomly sampled subset.

Table 7. Examples of contrasting words and phrases in the contradiction corpus. The examples colored by blue are indeed contrasting, while the examples colored by red are actually not contrasting.

Contrasting Words	Contrasting Phrases
ingratitude & loyalty	have not done anything right & did not do anything wrong
inconsequence & rationality	funded wholly & partially funded
inconspicuous & utter	domestic investment & foreign investments
meagerly & gratefully	the decrease demand & the growing demand
unusually & initially	she's coming advance & she'll be back

5. Related Work

A brief review of related works is presented from two perspectives: contradiction detection and learning continuous representations for a specific task.

Contradiction Detection. A strict logical definition of contradiction is that two sentences are contradictory if they cannot both be true in any world. The definition is loosened to capture human intuitions of incompatibility and better fit applications of recognizing discrepancies of the same event [1]. The looser definition of contradiction is that two sentences are contradictory when they are unlikely to be true at the same time. Contradiction detection aims to detect the semantic relation of contradiction among sentences. Condoravdi et al. first argued the importance of handling contradiction in text understanding [2]. As a kind of relation in the entailment recognition problem, contradiction has been studied continually in Recognizing Textual Entailment Challenges [13,24–29]. In previous years, researchers undertook the task by resolving some of the contradiction phenomena [1,30], like negation, antonyms, data/number mismatch and different structure. Ritter et al. utilized functional relations to recognize whether contradictions were apparent or actual [31]. In recent years, some researchers have focussed on finding contradictory parts in a pair of sentences [32–35]; however, they could not capture the contradictory relation between whole sentences. Our method captures both the global and local semantic relations.

As a crucial part of contradiction analysis, antonyms detection has been gaining increasing attention from researchers [7,36]. Lin et al. used a few "incompatibility" patterns to acquire antonyms [37]. Marneffe et al. expanded an antonym list for a word by adding words from the same synset in WordNet according to the direct antonym of the word [1]. VerbOcean is also used as a lexical resource. However, employing lexical resources is limited by low coverage. Hashimoto et al. presented a kind of sematic orientation, namely, excitatory or inhibitory, such as the words "cause" and "ruin". The authors argue that excitation is useful in extracting antonyms. However, the approach still cannot overcome the low coverage problem [38]. A multi-relational latent semantic analysis presented by Chang et al. [39] uses a three-way tensor to combine multiple relations between two words, in which one of the relations is antonymous. The authors use continuous space representations to capture lexical semantics through tensor decomposition techniques.

Word Representation Learning. Word representation is central to natural language processing (NLP). Harris states a distributional hypothesis that words in similar contexts have the same meanings [40]. Based on the distributional hypothesis, many methods are context-based learning word representations [41,42]. Since the development of the neural language model [43–47], it has become a popular approach to represent a word through a low-dimensionality continuous real-valued vector [10,48–51].

Traditional representation learning methods aim to capture semantic and syntactic similarities between two words [52]. A graph-based learning method is used for retrofitting word embedding by utilizing semantic lexicons [53]. However, contrasting relation is also a semantic relation between words. With the aim to resolve the sentiment contrast, a sentiment-specific word embedding [54] is learnt by weakly-supervised tweets collected by positive and negative emotions. Some neural network based models are proposed to revisit word embedding for lexical contrast [7–10]. There are two ways

to get the contrasting pairs for learning embeddings. Chen et al. [7] and Mrksic et al. [8] both use lexical resources to get antonym pairs; however, this is the small part of contrasting pairs. Schwartz et al. [9] apply patterns to get contrasting pairs from web text such as a Wikipedia page. This method also meet the low coverage problem, because the number of the contrasting pairs which can be described by "from X to Y" or "either X or Y" is limited.

6. Conclusions

A neural network is presented in this study to learn contradiction-specific word embedding (CWE) for a contradiction detection task. Traditional context-based word embedding algorithms typically map contrasting words into close vectors in an embedding space, which is problematic in contradiction detection direction. This issue is addressed by exploring CWE to maximize the semantic gap between contrasting words. A massive contradiction corpus is used to learn CWE. We develop a semantic relation representation learning model to detect the contradiction relation between sentences. CWE is then applied in this model to perform contradiction detection on a benchmark dataset from SemEval 2014. The experimental results show that CWE outperforms the traditional context-based word embedding in terms of contradiction detection.

Acknowledgments: This work was supported by the National High Technology Development 863 Program of China via grant 2015AA015407, the State Key Program of National Science Foundation of China via grant 61632011.

Author Contributions: Luyang Li conceived, designed and performed the experiments; Bing Qin supervised her student Luyang Li to write the paper; Ting Liu read and approved the final manuscript.

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

References

1. De Marneffe, M.C.; Rafferty, A.N.; Manning, C.D. *Finding Contradictions in Text*; ACL: Columbus, OH, USA, 2008; pp. 1039–1047.
2. Condoravdi, C.; Crouch, D.; De Paiva, V.; Stolle, R.; Bobrow, D.G. Entailment, intensionality and text understanding. In Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning, Edmonton, AB, Canada, 27 May–1 June 2003; pp. 38–45.
3. Kawahara, D.; Inui, K.; Kurohashi, S. Identifying contradictory and contrastive relations between statements to outline web information on a given topic. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Beijing, China, 23–27 August 2010; pp. 534–542.
4. Xu, L.; Yumoto, T.; Aoki, S.; Ma, Q.; Yoshikawa, M. Discovering Inconsistency in Multimedia News Based on a Material-Opinion Model. In Proceedings of the 2011 44th Hawaii International Conference on System Sciences, Koloa, HI, USA, 4–7 January 2011; pp. 1–10.
5. Tsytsarau, M.; Palpanas, T.; Denecke, K. Scalable discovery of contradictions on the web. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010; pp. 1195–1196.
6. Poria, S.; Cambria, E.; Hazarika, D.; Vij, P. A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks. *arXiv* **2016**, arXiv:1610.08815.
7. Chen, Z.; Lin, W.; Chen, Q.; Chen, X.; Wei, S.; Zhu, X.; Jiang, H. Revisiting word embedding for contrasting meaning. In Proceedings of the 53th Annual Meeting of the Association for Computational Linguistics (ACL 2015), Beijing, China, 26–31 July 2015.
8. Mrksic, N.; Seaghdha, D.; Thomson, B.; Gasic, M.; Rojasbarahona, L.; Su, P.H.; Vandyke, D.; Wen, T.H.; Young, S. Counter-Fitting Word Vectors to Linguistic Constraints. *arXiv* **2016**, arXiv:1603.00892.
9. Schwartz, R.; Reichart, R.; Rappoport, A. Symmetric Pattern Based Word Embeddings for Improved Word Similarity Prediction. In Proceedings of the Nineteenth Conference on Computational Natural Language Learning, Beijing, China, 30–31 July 2015; pp. 258–267.
10. Liu, Q.; Jiang, H.; Wei, S.; Ling, Z.H.; Hu, Y. Learning semantic word embeddings based on ordinal knowledge constraints. In Proceedings of the ACL, Beijing, China, 26–31 July 2015; pp. 1501–1511.
11. Miller, G.A.; Miller, G.A. WordNet: An on-line lexical database. *Int. J. Lexicogr.* **2010**, *3*, 235–244.

12. Ganitkevitch, J.; Van Durme, B.; Callison-Burch, C. PPDB: The Paraphrase Database. In Proceedings of the NAACL-HLT, Atlanta, GA, USA, 9–15 June 2013; pp. 758–764.
13. Marelli, M.; Bentivogli, L.; Baroni, M.; Bernardi, R.; Menini, S.; Zamparelli, R. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In Proceedings of the SemEval-2014, Dublin, Ireland, 23–24 August 2014.
14. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014.
15. Lai, A.; Hockenmaier, J. Illinois-lh: A denotational and distributional approach to semantics. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014),Dublin, Ireland, 23–24 August 2014; p. 329.
16. Bowman, S.R.; Potts, C.; Manning, C.D. Recursive Neural Networks Can Learn Logical Semantics. *arXiv* **2015**, arXiv:1406.1827.
17. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166.
18. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780.
19. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to Forget: Continual Prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471.
20. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. *Adv. Neural Inf. Process. Syst.* **2014**, *4*, 3104–3112.
21. Sak, H.; Senior, A.; Beaufays, F. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition. *arXiv* **2014**, arXiv:1402.1128.
22. Palangi, H.; Deng, L.; Shen, Y.; Gao, J.; He, X.; Chen, J.; Song, X.; Ward, R. Deep Sentence Embedding Using the Long Short-Term Memory Networks. *arXiv* **2015**, arXiv:1502.06922.
23. Schmidhuber, K.G.S.K.S. LSTM: A Search Space Odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, doi:10.1109/TNNLS.2016.2582924.
24. Dagan, I.; Glickman, O.; Magnini, B. The PASCAL recognising textual entailment challenge. In Proceedings of the Machine Learning Challenges: Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment; Southampton, UK, 11–13 April 2005; pp. 177–190.
25. Giampiccolo, D.; Magnini, B.; Dagan, I.; Dolan, B. The third pascal recognizing textual entailment challenge. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic, 28–29 June 2007; pp. 1–9.
26. Giampiccolo, D.; Magnini, B.; Dang, H.T. The fourth pascal recognising textual entailment challenge. *J. Nat. Lang. Eng.* **2009**, *3944*, 177–190.
27. Voorhees, E.M. Contradictions and Justifications: Extensions to the Textual Entailment Task. In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, Columbus, OH, USA, 15–20 June 2008; pp. 63–71.
28. Bentivogli, L.; Dagan, I.; Dang, H.T.; Giampiccolo, D.; Magnini, B. The fifth pascal recognizing textual entailment challenge. In Proceedings of the TAC, Gaithersburg, MD, USA, 16–17 November 2009; pp. 14–24.
29. Dagan, I.; Dolan, B.; Magnini, B.; Roth, D. Recognizing textual entailment: Rational, evaluation and approaches—erratum. *Nat. Lang. Eng.* **2010**, *15*, doi:10.1017/S1351324909990234.
30. Harabagiu, S.; Hickl, A.; Lacatusu, F. Negation, contrast and contradiction in text processing. In Proceedings of the 21st National Conference on Artificial Intelligence, Boston, MA, USA, 16–20 July 2006; pp. 755–762.
31. Ritter, A.; Downey, D.; Soderland, S.; Etzioni, O. It's a contradiction—No, it's not: A case study using functional relations. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Honolulu, HI, USA, 25–27 October 2008; pp. 11–20.
32. Magnini, B.; Cabrio, E. Contradiction-focused qualitative evaluation of textual entailment. In Proceedings of the Workshop on Negation and Speculation in Natural Language Processing, Uppsala, Sweden, 10 July 2010; pp. 86–94.
33. Shih, C.; Lee, C.; Tsai, R.T.; Hsu, W. Validating Contradiction in Texts Using Online Co-Mention Pattern Checking. *ACM Trans. Asian Lang. Inf. Process.* **2012**, *11*, 17.

34. Kloetzer, J.; Saeger, S.D.; Torisawa, K.; Hashimoto, C.; Oh, J.H.; Sanok, M.; Ohtake, K. Two-stage Method for Large-Scale Acquisition of Contradiction Pattern Pairs using Entailment. In Proceedings of the EMNLP 2013, Seattle, WA, USA, 18–21 October 2013.
35. Liu, L.L.Q. Generating Triples Based on Dependency Parsing for Contradiction Detection. *SMP* **2015**, doi:10.1007/978-981-10-0080-5_19.
36. Mohammad, S.M.; Dorr, B.J.; Hirst, G.; Turney, P.D. Computing Lexical Contrast. *Comput. Linguist.* **2013**, *39*, 555–590.
37. Lin, D.; Zhao, S.; Qin, L.; Zhou, M. Identifying synonyms among distributionally similar words. In Proceedings of the 18th International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 9–15 August 2003; pp. 1492–1493.
38. Hashimoto, C.; Torisawa, K.; De Saeger, S.; Oh, J.H.; Kazama, J. Excitatory or inhibitory: A new semantic orientation extracts contradiction and causality from the web. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, 12–14 July 2012; pp. 619–630.
39. Chang, K.W.; Yih, W.T.; Meek, C. Multi-Relational Latent Semantic Analysis. In Proceedings of the EMNLP, 18–21 October 2013; pp. 1602–1612.
40. Harris, Z.S. Distributional structure. *Word* **1954**, *10*, 146–162.
41. Brown, P.F.; Desouza, P.V.; Mercer, R.L.; Pietra, V.J.D.; Lai, J.C. Class-based n-gram models of natural language. *Comput. Linguist.* **1992**, *18*, 467–479.
42. Uszkoreit, J.; Brants, T. Distributed Word Clustering for Large Scale Class-Based Language Modeling in Machine Translation. In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, Columbus, OH, USA, 15–20 June 2008; pp. 755–762.
43. Bengio, Y.; Ducharme, R.; Vincent, P.; Janvin, C. A neural probabilistic language model. *J. Mach. Learn. Res.* **2003**, *3*, 1137–1155.
44. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 160–167.
45. Mnih, A.; Hinton, G.E. A scalable hierarchical distributed language model. *Adv. Neural Inf. Process. Syst.* **2009**, *1*, 1081–1088.
46. Mikolov, T.; Karafiát, M.; Burget, L.; Cernockỳ, J.; Khudanpur, S. Recurrent neural network based language model. In Proceedings of the 11th Annual Conference of the International Speech Communication Association, Makuhari, Japan, 26–30 September 2010; pp. 1045–1048.
47. Poria, S.; Cambria, E.; Gelbukh, A. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowl.-Based Syst.* **2016**, *108*, 42–49.
48. Hinton, G.E.; Roweis, S.T. Stochastic neighbor embedding. *Adv. Neural Inf. Process. Syst.* **2002**, *41* 833–840.
49. Turney, P.D.; Pantel, P. From frequency to meaning: Vector space models of semantics. *J. Artif. Intell. Res.* **2010**, *37*, 141–188.
50. 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.
51. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed representations of words and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 3111–3119.
52. Levy, O.; Goldberg, Y. Dependencybased word embeddings. In Proceedings of the ACL, Baltimore, MD, USA, 22–27 June 2014; pp. 302–308.
53. Faruqui, M.; Dodge, J.; Jauhar, S.K.; Dyer, C.; Hovy, E.; Smith, N.A. Retrofitting word vectors to semantic lexicons. *arXiv* **2014**, arXiv:1411.4166.
54. Tang, D.; Wei, F.; Yang, N.; Zhou, M.; Liu, T.; Qin, B. Learning sentiment-specific word embedding for twitter sentiment classification. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Baltimore, MD, USA, 22–27 June 2014; pp. 1555–1565.