Neurocomputing 323 (2019) 52–61

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Mining predicate-based entailment rules using deep contextual architecture

Maosheng Guo*, Yu Zhang, Dezhi Zhao, Ting Liu

Research Center for Social Computing and Information Retrieval, Harbin Institute of Technology, China

ARTICLE INFO

Article history:
Received 29 December 2017
Revised 6 June 2018
Accepted 21 September 2018
Available online 26 September 2018

Communicated by Dr. T. Mu

Keywords:
Entailment rules mining
Natural language inference
Knowledge acquisition
Deep belief networks
Deep neural networks
Artificial intelligence

ABSTRACT

Semantic inference plays an essential role in numerous Natural Language Processing (NLP) tasks, such as question answering, machine reading and text summarization. Reasoning in natural language is inseparable from the knowledge about inference, which is often represented in the form of predicate-based entailment rules. Many efforts have been dedicated to extracting entailment rules from text corpora by utilizing statistical methodology including distributional hypothesis and Latent Dirichlet Allocation (LDA). However, these studies could not give equal consideration to both coverage and accuracy of the mined rules, which brings instability to downstream applications. To solve this problem, this paper proposes a novel model named Deep Contextual Architecture (DCA), which is driven by Deep Belief Networks (DBNs), for the task of mining predicate-based inference rules from texts. Besides previously used statistical contextual information, we also involve semantic meanings represented by word embeddings into DBNs to learn topic level representation of predicates. Combining benefits from both kinds of information, the proposed DCA model shows potential for better modeling the context of predicates. Evaluation on public datasets demonstrates that our method outperforms several strong baselines.

1. Introduction

As reasoning phenomenon is widely distributed in text corpora, the ability of inference plays an essential role in many NLP applications. Research on reasoning in natural language is called textual entailment, which improves the performance of question answering systems [1], machine reading [2], text summarization [3], parser evaluation [4], and other NLP-related tasks. Specifically, textual entailment is defined as a directional reasoning relationship between two texts. Given a pair of text snippets, T and H, if the meaning of H could be inferred from T, then T entails H, denoted by $T \rightarrow H$. For example, '*Tom bought a book.*' → '*Tom had a book.*'

Furthermore, correct handling reasoning relation is inseparable from the accumulation of knowledge about textual entailment, which is the basis for reasoning in natural language. Knowledge about entailment is often represented in the form of inference rules, as shown in Fig. 1. For example, '*peach* → *fruit*' is the reasoning principle for '*He ate a peach.* → *He ate a piece of fruit.*'; while '*X acquire Y* → *X purchase Y*' makes '*Verizon acquired Yahoo.* → *Verizon purchased Yahoo.*' reasonable. '*Peach* → *fruit*' in the first example is a noun-based entailment rule which is usually in the form of '*hyponym* → *hypernym*' that does not contain variables; whilst '*X acquire Y* → *X purchase Y*' is a predicate-based entailment template which often includes slots, i.e., subject (X) and object (Y). In this paper, we focus on mining the later type of rules from texts.

More formally, predicted-based entailment rules are defined as follows:

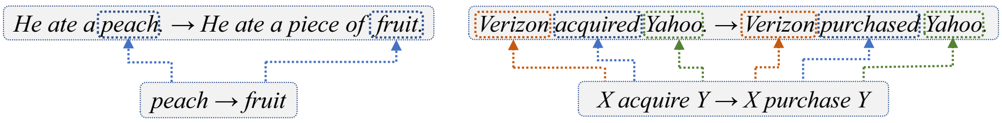
$$X\, V_L\, Y \rightarrow X\, V_R\, Y,$$

where 'V_L' and 'V_R' are two predicates, and X is their subject while Y is their object. The tuple $<X, Y>$ forms the context of a predicate, as illustrated in Fig. 2. The word instances to fill the slots in contexts are called filler words, e.g., '*Verizon*' is the filler word of X in the example in Fig. 1(b).

It is worth noting that the reasoning relation indicated by predicate-based entailment rules might not hold true in all contexts. As shown in Fig. 3, we could not infer the proposition '*Children purchased a good knowledge of English.*' from the sentence '*Children acquired a good knowledge of English.*', because the rule '*X acquire Y* → *X purchase Y*' is not valid in the context $<$*children, English*$>$. In fact, '*acquire*' here means gaining an ability. So a possible entailment rule is '*X acquire Y* → *X learn Y*', i.e., '*Children acquired a good knowledge of English.* → *Children learned a good knowledge of English*'. This problem could be addressed by introducing a confidence score for each context, which is formally

* Corresponding author.
E-mail addresses: msguo@ir.hit.edu.cn (M. Guo), zhangyu@ir.hit.edu.cn (Y. Zhang), dzzhao@ir.hit.edu.cn (D. Zhao), tliu@ir.hit.edu.cn (T. Liu).

https://doi.org/10.1016/j.neucom.2018.09.046


He ate a peach → He ate a piece of fruit
peach → fruit
Verizon acquired Yahoo → Verizon purchased Yahoo
X acquire Y → X purchase Y

(a) Noun-based rule & application

(b) Predicate-based rule & application

Fig. 1. Examples of text pairs with reasoning relation and the corresponding entailment rules.

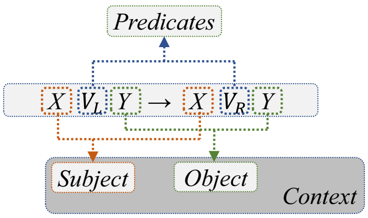


Fig. 2. Context of predicates.

defined in Section 3.1. The confidence of applying ‘*X acquire Y* → *X learn Y*’ is, say, 0.8 which is higher than that of the other rule, indicating that it is more acceptable in the current context.

Numerous efforts [5–12] have been dedicated to mining entailment rules. These methods could be divided into two categories according to whether the confidence score is sensitive to the contexts of predicates or not: context-aware and context-unaware approaches.

Context-aware methods employ LDA (Latent Dirichlet Allocation) as the topic model to detect the contexts, which often get a higher accuracy than the context-unaware ones. However, they require the filler words to be seen in the training set, resulting in a lower coverage than the context-unaware approaches. On the contrary, context-unaware methods could cover more application scenarios without restriction on filler words, but usually receive a lower accuracy during rules application stage due to lack of knowledge about the contexts. In other words, these methods could not give equal consideration to both coverage and accuracy of the extracted rules.

Moreover, only employing statistical information is not robust enough to model the semantic meaning of predicates, especially when the data source is limited so that it could not cover all collocations or contexts, resulting in an unstable performance in downstream applications. More specifically, if a filler word has never appeared in the context of a predicate in the source corpora, LDA-based statistical models might tend to give an underestimated confidence to it. Furthermore, if the filler words have never been seen in the training set, these statistical models could not even make a judgment at all.

To alleviate the instability and gain better generalization capability, we involve pre-trained word embeddings in our model, because word embeddings obtained through neural language models from large corpora can capture both semantic and grammatical behavior of words, and has proved capable of finding relations between words [13,14]. In addition, pre-trained embeddings with millions of words could fill the vocabulary gap between the training set and application scenarios, resulting in a higher coverage.

Deep neural models have been applied to many NLP tasks showing promising results. However, to the best of our knowledge, no prior deep learning-based approach has been proposed for the entailment rules mining task. After exploring various deep neural models, we choose Deep Belief Networks (DBNs) as the topic model to learn latent contextual representations of predicates in this task, according to the following reasons: Firstly, DBNs were proposed by Hinton and Salakhutdinov [15] for nonlinear dimensionality reduction and input data reconstruction, which is ideal for modeling the contexts of predicates in a latent low-dimensional topic perspective. Moreover, DBNs could be trained in an unsupervised manner, which is suitable for the task of entailment rules mining. Recently, DBNs as a topic model outperforms LDA in a digital publishing recommender system [16]. In this paper, DBNs show potential for better modeling the context of predicates by considering information from both statistics and semantics. Evaluation on public datasets demonstrates that our method outperforms several strong baselines on both accuracy and coverage metrics.

This paper describes a novel model named Deep Contextual Architecture (DCA), driven by deep belief networks to mine predicate-based inference rules from texts. Our main contributions could be summarized as follows:

- The proposed DCA model utilizes not only commonly used statistical information but also semantic meanings brought by word embeddings, which can profoundly improve the accuracy of extracted inference rules.
- Entailment rules mined by our method could cover more application scenarios than LDA-based approaches, which has a more strict restriction that all filler words in a context must have appeared in training dataset.
- We have shown that employing DBNs as a topic model is helpful for the task of mining predicate-based entailment rules. As far as we know, it is the first deep neural model achieving state-of-the-art performance on this task.

The remainder of this paper is organized as follows. In Section 2 we discuss the related work. Next, Section 3 introduces our new method for predicate-based entailment rules mining. In addition, Section 4 gives details about our experiment settings and results analysis. Finally, in Section 5 we conclude the paper.

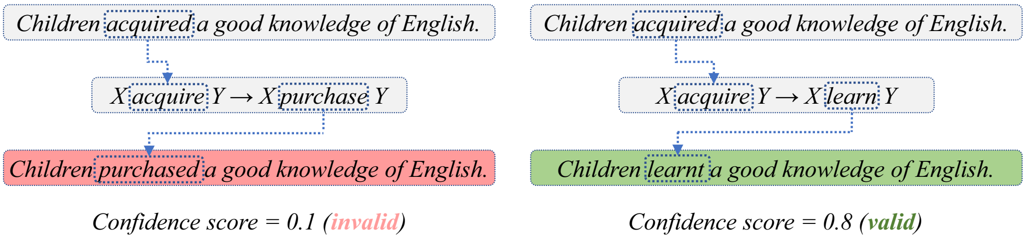

Children acquired a good knowledge of English.
X acquire Y → X purchase Y
Children purchased a good knowledge of English.
Confidence score = 0.1 (invalid)
Children acquired a good knowledge of English.
X acquire Y → X learn Y
Children learnt a good knowledge of English.
Confidence score = 0.8 (valid)

Fig. 3. Rules are not valid in every context.

2. Related work

Since early 2001, learning how to extract predicate-based entailment rules has grown to be one of the most active research areas in NLP. Various methods have been proposed. Lin and Pantel [5] were the first researchers to study how to extract predicate-based entailment rules from raw texts. They proposed the DIRT (Discovery of Inference Rules from Text) method based on the famous distributional hypothesis introduced by Harris [17]. Harris thought that words that occur in the same contexts tend to have similar meanings. Lin et al. [5] extended this theory to the reasoning relation between predicates, that is, predicates with similar contexts have similar meanings and might entail each other. If the similarity of a pair of predicates exceeds some threshold, DIRT will admit the entailment relationship between them.

In fact, there are three defects in DIRT methodology: Firstly, the reasoning relation between two predicates might not hold true even if their contexts are very similar. This phenomenon usually exists in a pair of antonyms, e.g., '*X solve Y* $\nrightarrow$ *X worsen Y*'. Therefore, DIRT sometimes extracts antonyms with similar contexts besides entailment. Secondly, DIRT does not provide information of the direction of entailment. For example, DIRT does not tell us whether '*X buy Y* → *X own Y*' or '*X own Y* → *X buy Y*'. However, textual entailment is a unidirectional relation in most cases. Human annotation shows that only 20% to 25% rules extracted by DIRT could remain true bidirectionally [6]. Thirdly, DIRT puts the same confidence in a rule in different contexts. In other words, DIRT believes that the extracted rules are universally applicable, which is not true because we could easily give counterexamples, e.g. '*X acquire Y* → *X purchase Y*' comes right in the context of $<$*Verizon, Yahoo*$>$, but not in $<$*children, English*$>$. If we apply rules indiscriminately, it will bring uncertainty to downstream applications.

Szpektor et al. [7] proposed another entailment rules extracting approach called TE/ASE (Template Extraction/Anchor Set Extraction) which is based on the idea of bootstrapping. TE/ASE could not provide information of the direction of entailment or context-sensitive confidence either.

To learn the directionality of inference rules, Bhagat et al. [8] proposed an approach named LEDIR (LEarning Directionality of Inference Rules) which uses selectional preference to gather evidence of the direction of entailment rules. Szpektor et al. [9] proposed a directional similarity measure called BInc (Balanced-Inclusion) to identify the directionality of entailment rules.

Ritter [10], Dinu and Lapata [11], Melamud et al. [12] employed LDA (Latent Dirichlet Allocation) as topic models to describe the contexts of predicates. In general, they first treat contexts of predicates as document collections and learn the latent topic distribution of predicates by LDA. Then they utilize a confidence score computed by the topic distribution of two predicates in the given context to recognize entailment relation between them. These LDA-based approaches provide a context sensitive score for each particular rule application, resulting in a higher accuracy than previous context-unaware methods. However, these LDA driven context-aware approaches often suffer from a low coverage at rule application stage due to their strict restriction on filler words.

Our approach is inspired by these prior works. We re-implement those inference rules extraction methods[1] as baselines on the same dataset with our approach and analyze them in Section 4.4. Besides previously used statistical contextual information, we also involve semantic meanings represented by word embeddings that have proved capable of revealing relations between words, which significantly improve rules mining. Moreover, we employ DBNs instead of LDA as our topic model to learn latent contextual representations of predicates. DBNs have outperformed LDA as a topic model in some tasks [16]. Our experiments also show its potential for better modeling the contexts of predicates on this task.

3. Method description

In this section, we present the DCA model and the corresponding method for predicate-based inference rules mining. Given two predicates, four problems need to be addressed to complete the task of mining entailment rules from a textual corpus. Firstly, we need to model the context of a predicate (Section 3.2); secondly, we should judge whether one predicate could be inferred from the other, i.e. entailment relation recognition (Section 3.3); thirdly, we should identify the direction of entailment, i.e., which predicate comes first (Section 3.4); lastly, we should give the confidence of the rule, i.e. how probable the rule is applicable under some specific contexts (Section 3.5). Fig. 4 illustrates the four-step workflow by taking '*Verizon acquire Yahoo* → *Verizon purchase Yahoo*' as an example, where sub-steps in each pane will be introduced in the next sections.

3.1. Formalized notations of predicates and their contexts

For a better description of our method, we will use the following formal notations: Our goal is to mine predicate-based entailment rules with two variables, '$X\,V_L\,Y \rightarrow X\,V_R\,Y$', where V_L and V_R are two different predicates, with X as their subject and Y as the object. X and Y form the context of predicates, denoted by the $<X,Y>$ (Fig. 2). Text corpus D is the knowledge source of inference rules. The multiset $d^X_{V_L}$ represents a word bag containing all words that occur in D as subjects (X) of the predicate V_L, as depicted in Fig. 5. Similarly, we have $d^Y_{V_L}$, $d^X_{V_R}$ and $d^Y_{V_R}$. For example, $d^X_{acquire}$ is all subjective filler words of 'acquire' appearing in the dataset, i.e, {*Verizon*, *Children*, *Verizon*, ...}. It is worth noting that each word in a multiset could occur repeatedly and word orders are ignored. Moreover, if we do not mean to distinguish between 'left' and 'right', i.e. the position where a predicate lies in a rule, we could use the single symbol v to represent a predicate. Similarly, we employ the symbol w to indicate a word that could fill the slot X or Y, or use w_x and w_y respectively. Furthermore, we treat the context of a predicate as a probability distribution over topics denoted by the symbol t. In the end, our method should give the confidence score $F(V_L, V_R, w_x, w_y)$ to apply the rule '$X\,V_L\,Y \rightarrow X\,V_R\,Y$' in the context of $<w_x, w_y>$.

3.2. Contexts modeling by deep contextual architecture

A predicate could have more than one meaning in different contexts. For example, the verb 'acquire' has two meanings at least, one for buying something while the other is about learning an ability. We treat the meanings as topics of predicates. For example, if the topic of the word 'acquire' is (*buying, learning*), the probability distribution under the context $<$*Verizon,Yahoo*$>$ might be (0.9,0.1), while it might be (0.2,0.8) for $<$*children,English*$>$.

More formally, the task to model the context of a predicate is to describe the conditional probability $P(t|v, w_x, w_y)$. Due to the duality and symmetry of the context $<w_x, w_y>$, it could be decomposed into two subtasks, i.e., to compute $P(t|v, w_x)$ and $P(t|v, w_y)$ separately, and merge them to obtain the final confidence score. In this subsection, we first try to model the context of a predicate with only one variable (X or Y), i.e., $P(t|v, w)$, which represents the conditional probability distribution over topic t given the predicate v and its filler words w.

[1] All methods mentioned in this section are re-implemented except the bootstrapping-based approach TE/ASE whose performance relies heavily on hand-crafted seeds which are not publicly available.

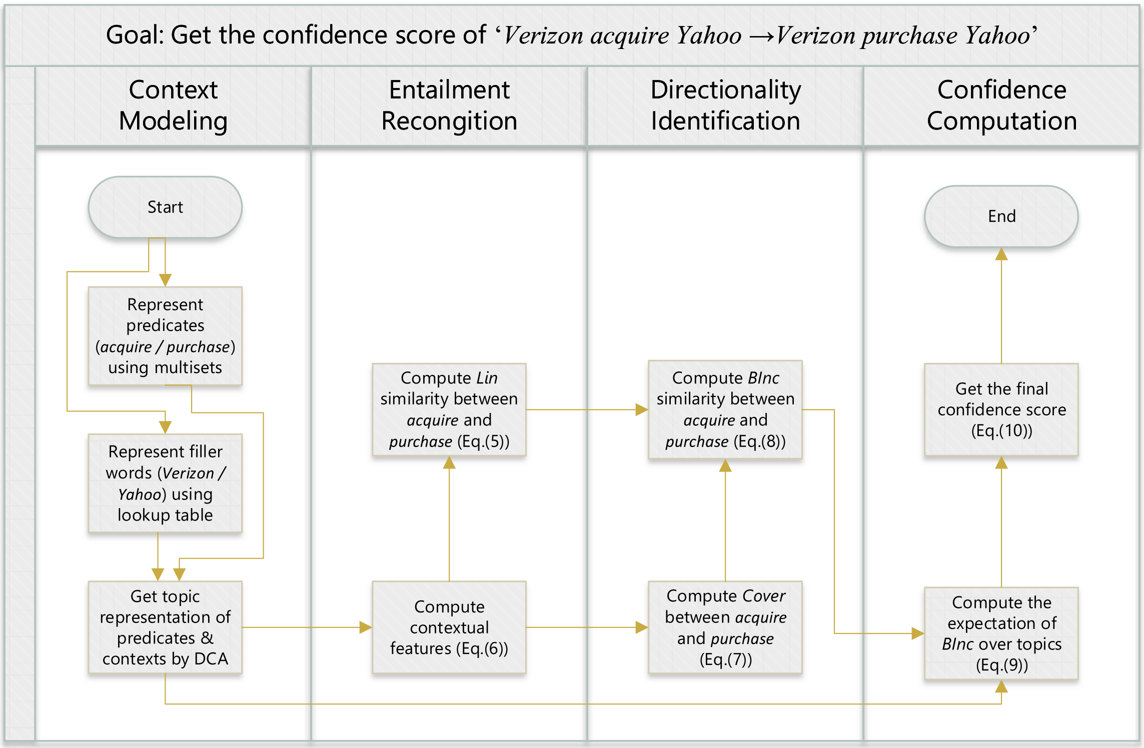

Goal: Get the confidence score of '*Verizon acquire Yahoo* → *Verizon purchase Yahoo*'
Context Modeling
Entailment Recognition
Directionality Identification
Confidence Computation
Start
Represent predicates (*acquire / purchase*) using multisets
Represent filler words (*Verizon / Yahoo*) using lookup table
Get topic representation of predicates & contexts by DCA
Compute contextual features (Eq.(6))
Compute *Lin* similarity between *acquire* and *purchase* (Eq.(5))
Compute *Cover* between *acquire* and *purchase* (Eq.(7))
Compute *BInc* similarity between *acquire* and *purchase* (Eq.(8))
Compute the expectation of *BInc* over topics (Eq.(9))
Get the final confidence score (Eq.(10))
End

Fig. 4. The flowchart of computing confidence score by taking '*Verizon acquire Yahoo* → *Verizon purchase Yahoo*' as an example.

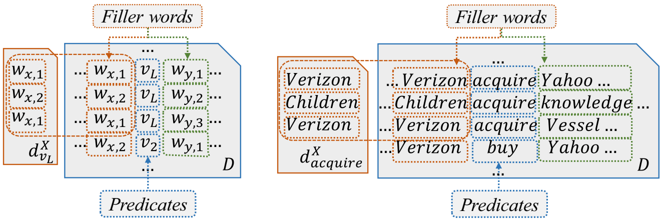


Fig. 5. Illustration of text corpus D and multiset $d^X_{V_L}$, $d^X_{acquire}$.

The DCA model utilize DBNs as its topic model to compute the value of $P(t|v, w)$. As shown in Fig. 6, the model consists of four different components from bottom up, a statistics part which represents the predicate v, a lookup table to get the word embeddings of filler words w, an AverageNN to compute the representation of the filler words, and a DBN to get the probability distribution over topics t given v and w.

When the statistics part is fed with a predicate v, it first extracts all filler words $(w_{x,1}, w_{x,2}, \dots)$ occurred in the knowledge source D to form a multiset d^X_v (without loss of generality, let us say w is from the slot X.[2]), and then compute the normalized term frequency of each $w_{x,\,i}$ as follows:

$$\widehat{\#w}_{x,i} = \frac{\#w_{x,i}}{\max_j \#w_{x,j}}, \tag{1}$$

where $i, j \in 1, \dots, N$, and N is the vocabulary size of filler words in D. Finally it produces a fixed-length vector $\boldsymbol{v} = (\widehat{\#w}_{x,1}, \widehat{\#w}_{x,2}, \dots, \widehat{\#w}_{x,N})$ to represent the predicate v.

Meanwhile, the lookup table is used to find vectors $\boldsymbol{w_k}$ to represent each filler word w_k, which are inputted into an AverageNN to get a fixed-length representation of filler words as follows:

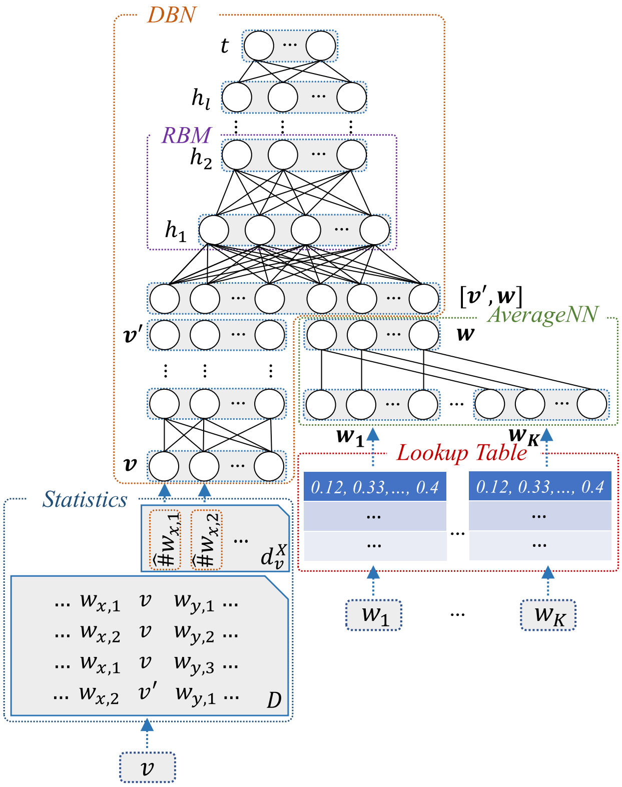


Fig. 6. The DCA model to describe the context of a predicate.

$$\boldsymbol{w} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{w_k}, \tag{2}$$

[2] If w comes from Y, the multiset is d^Y_v. The remaining parts are similar.

where K is the number of words filled into the slot X. Most slots are filled by a single word, e.g., 'Children' in Fig. 3, while some slots might be filled by a phrase of multiple words, e.g. 'a good knowledge of English', which is the reason to introduce AverageNN here.

As depicted in Fig. 6, a DBN is formed by stacks of RBMs (Restricted Boltzmann Machines), where each RBM's hidden layer serves as the input layer for the next. An RBM is an undirected generative energy-based model with an input layer, a hidden layer and connections between but not within layers. For example, h_1 is the input layer and h_2 is the hidden layer of an RBM, and h_2 could be viewed as a re-representation of h_1 at a lower dimension.

The inputs of the DBN are the vectors $\boldsymbol{v}$ and $\boldsymbol{w}$, which contain not only statistical features from the knowledge source but also semantic information from a pre-trained word vectors. Because the dimensionality of $\boldsymbol{v}$ is way larger than that of $\boldsymbol{w}$, the DCA model first employs a stack of RBMs to reduce the dimensionality of $\boldsymbol{v}$ to generate a vector $\boldsymbol{v}'$, which has a similar dimensionality with $\boldsymbol{w}$, and then concatenate them as a single vector $[\boldsymbol{v}', \boldsymbol{w}]$ that is fed to another stack of RBMs to output the topics distribution over t. Finally we have:

$$P(t|v,w) = Softmax(DBN(v,w;\theta)), \tag{3}$$

where θ represents DBN's parameters which are trained using contrastive divergence algorithm [15].

3.3. Entailment relation recognition

To recognize textual entailment relationship between predicates from a corpus, we extend DIRT [5] with the contextual information captured by DCA.

The main idea behind DIRT is the distributional hypothesis, i.e. two predicates with similar contexts have similar meanings and might entail each other. The key is how to measure the similarity between two predicates. As one predicate has two variables, X and Y, we could compute the similarity with two variables by geometric averaging the similarity score with slot X and that with variable Y. In fact, DIRT employs the Lin-similarity. Eq. (4) describes the definition of Lin-similarity $Lin(v_L, v_R)$ with one variable:

$$Lin(v_L, v_R) = \frac{\sum_{w \in d_{v_L} \cap d_{v_R}} [PMI(v_L, w) + PMI(v_R, w)]}{\sum_{w \in d_{v_L}} PMI(v_L, w) + \sum_{w \in d_{v_R}} PMI(v_R, w)}, \tag{4}$$

where $PMI(v, w)$ is the pointwise mutual information of predicate v and filler word w, which is computed by the empirical probability from the source corpus D.

Our extension to Lin-similarity is to add contextual information captured by DCA to it, i.e. $Lin_t(v_L, v_R)$, which could be computed as below:

$$Lin_t(v_L, v_R) = \frac{\sum_{w \in d_{v_L} \cap d_{v_R}} [u_t(v_L, w) + u_t(v_R, w)]}{\sum_{w \in d_{v_L}} u_t(v_L, w) + \sum_{w \in d_{v_R}} u_t(v_R, w)}, \tag{5}$$

where $u_t(v, w)$ is the contextual feature function computed by:

$$u_t(v,w) = PMI(v,w)P(t|v,w), \tag{6}$$

where $P(t|v, w)$ is the topic distribution modeled by DCA. Note that after complementing contextual information, the similarity in different contexts varies, in other words, the extended Lin-similarity is context sensitive, which is desirable for entailment rules extraction.

3.4. Entailment directionality identification

It is easy to conclude that both Lin-similarity $Lin(v_L, v_R)$ and our contextual extension $Lin_t(v_L, v_R)$ are symmetrical about (v_L, v_R). If we exchange the position of v_L and v_R, the similarity measurement remains the same, i.e., $Lin_t(v_L, v_R) = Lin_t(v_R, v_L)$. In other words, we could not distinguish the direction of entailment rules only using Lin-similarity.

We absorb the idea from BInc [9] to utilize the coverage of filler words features by the other as a directional similarity:

$$Cover_t(v_L, v_R) = \frac{\sum_{w \in d_{v_L} \in d_{v_R}} u_t(v_L, w)}{\sum_{w \in d_{v_L}} u_t(v_L, w)}. \tag{7}$$

Compared with Lin-similarity, $Cover_t(v_L, v_R)$ is asymmetric and capable of capturing directional information. However, it tends to prefer rules with infrequent v_L. If a rare predicate v_L has common filler words with another predicate v_R, the coverage of its features is usually high, even if there is no entailment relationship between them. Thus, only employing $Cover_t(v_L, v_R)$ could not extract entailment rules effectively either. Similar to the vanilla BInc value, our contextual BInc-similarity employs both $Lin_t(v_L, v_R)$ and $Cover_t(v_L, v_R)$ by geometric mean:

$$BInc_t(v_L, v_R) = \sqrt{(Lin_t(v_L, v_R)Cover_t(v_L, v_R)}. \tag{8}$$

3.5. Confidence computation

Up to now, we have the topic distribution given the entailing (left) predicate v_L with its filler word w, i.e. $P(t|v_t, w)$, and the contextual similarity between the two predicates v_L and v_R for each topic t, i.e., $BInc_t(v_L, v_R)$. It is intuitive to compute the mathematical expectation of the similarity with respect to the distribution $P(t|v_L, w)$, that is, the probability-weighted averaged $BInc_t(v_L, v_R)$ of all possible topics, which is described in Eq. (9):

$$\begin{aligned} sim(v_L, v_R, w) &= \mathbb{E}_{t \sim P(t|v_L, w)}[BInc_t(v_L, v_R)] \\ &= \sum_t [P(t|v_L, w) BInc_t(v_L, v_R)]. \end{aligned} \tag{9}$$

We obtained the similarity between the two predicates with just one slot, i.e., $sim(v_L, v_R, w)$. In fact, there are two variables, i.e. X and Y, in an entailment rule. According to the symmetry about slots X and Y, the confidence score $F(V_L, V_R, w_x, w_y)$ of the rule '$X V_L Y \rightarrow X V_R Y$' under the context of $< w_x, w_y >$ is computed as follows:

$$F(V_L, V_R, w_x, w_y) = \sqrt{sim(v_L, v_R, w_x) sim(v_L, v_R, w_y)}. \tag{10}$$

4. Experiments and results analysis

Comparative experiments were conducted to evaluate the effectiveness of our method. In this section, we describe the experimental design and analyze the results to demonstrate the superiority of the DCA-driven approach for inference rules mining task.

4.1. Dataset construction

4.1.1. Training set

To evaluate our proposed entailment rules mining approach, we choose the ReVerb ClueWeb Extractions corpus[3] [18] as the data source D. ReVerb is a corpus consisting of 15 million tuples $< w_x, v, w_y >$, which are extracted from English web pages.

To reduce the number of predicates and enrich the context of them, we filtered out stop words to normalize predicates. For example, the predicate of '*X can accommodate up to Y*', '*X will accommodate up to Y*' and '*X accommodate up to Y*' are treated as the same predicate '*accommodate up to*' [12]. Furthermore, we removed predicates with less than 30 distinct filler words.

[3] http://reverb.cs.washington.edu/.

Table 1
Inference rules examples from the test set.

Entailment rules	The corresponding applications	Annotation
X hurt Y → *X harm Y*	*Neutering hurt my pet* → *Neutering harm my pet*	Correct
X need to use Y → *X need more than Y*	*Most users need to use this feature* → *Most users need more than this feature*	Incorrect

Besides, we preprocessed filler words too, by deleting stop words, rare words and non-ASCII characters. At last, our dataset has 6,607 different predicates and 30,000 unique filler words.[4]

4.1.2. Test set

The test set was extracted from Zeichner's dataset[5] [19], which contains 6,567 human annotated applications of inference rules labeled either correct or incorrect, as shown in Table 1. For example, '*Neutering hurt my pet.*→ *Neutering harm my pet.*' was annotated as correct for '*X hurt Y* → *X harm Y*', while '*Most users need to use this feature.* → *Most users need more than this feature.*' was labeled as incorrect for the rule '*X need to use Y* → *X need more than Y*'.

We implemented all inference rules mining approaches described in Section 2. For the reason that Zeichner's dataset could not cover all rules extracted by these methods and these approaches are not guaranteed to extract all rules in that dataset, we only chose the intersection among sets of rules extracted by each method and Zeichner's dataset as our test set, which consists of 214 positive inference rules applications and 397 negative entailment rules applications, resulting in 611 examples in total. Formally as below:

$$\begin{aligned}\text{test\_set} = &(\cap_i\{\text{rules extracted by method}_i\})\\ &\cap \text{Zeichner}'\text{s dataset}.\end{aligned} \tag{11}$$

4.1.3. Development set

Although our method could be trained in an unsupervised manner, there exist some hyper-parameters in the DCA model, e.g., the depth of DBNs and the dimensionality of each layer, which should be tuned using a development set. We discard all the rules included in the test set from the intersection between rules extracted by our approach and Zeichner's dataset, and employ the remaining as the development set, which consists of 243 positive examples and 463 negative examples, resulting in 706 examples in total. Formally as follow:

$$\begin{aligned}\text{dev\_set} = &\{\text{rules mined by our method}\}\\ &\cap \text{Zeichner}'\text{s dataset} - \text{test\_set}.\end{aligned} \tag{12}$$

4.2. Evaluation metrics

To facilitate comparison with previous works, we adopted the evaluation metric used by Melamud et al. [12], i.e., *MAP* (Mean Average Precision). We followed their assessment settings to split the test set randomly into 30 subsets $\{S_n\}_{n=1}^{30}$, and then ranked the samples in subsets according to the confidence scores provided by our approach or baseline methods, then computed the average precision of each subgroup, and finally calculated the mean of average precision, as follows:

$$MAP = \frac{1}{N}\sum_{n=1}^{N} AveragePrecision(S_n), \tag{13}$$

where $N = 30$.

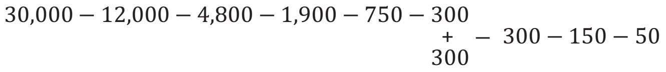


Fig. 7. The structure of DBNs in the DCA model.

In the ideal case, the confidence scores of positive applications are higher than those of negative samples. Therefore, all positive applications are in front of negative ones. If there exists some undesirable situation that confidence scores of negative samples are higher than correct applications, negative cases will be inserted before positive samples, so that the *MAP* score will decrease. In conclusion, *MAP* is suitable for evaluation of entailment rules mining approaches.

In addition to *MAP*, we introduce the coverage of Zeichner's dataset as a evaluation metric to measure the applicability of each methods:

$$Coverage = \frac{\#(\{\text{rules extracted}\} \cap \text{Zeichner}'\text{s dataset})}{\#\text{Zeichner}'\text{s dataset}}. \tag{14}$$

In fact, *Coverage* uses Zeichner's dataset to stimulate practical application scenarios to evaluate the generalization ability of methods. Thus, higher *Coverage* is better.

4.3. Experiments settings

To guarantee the precision of extracted inference rules and reduce the amount of computation, we only kept the most confident 1,500[6] rules per predicate. Furthermore, to reduce the dimensionality of input vectors to DCA, we only kept the most frequent 30,000 filler words from knowledge source D to represent a predicate v in the statistics part of the model. We employed the 300-dimensional pre-trained GloVe[7] word vectors, which was commonly used in previous recognizing textual entailment tasks [20–22], as the representation of filler words w in the lookup table parts. The number of topics for both DCA and LDA was fixed to 50 which is under the same setting of [12]. Fig. 7 describes the structure of the DBNs in the DCA model, which was tuned on the development set.

4.4. Results analysis

4.4.1. Effect of the depth of DBNs

We first tuned the depth of the DBNs in DCA on the development set to find the effect of depth on this task. Table 2 shows the structures we evaluated on the development set to find the best depth of the DBNs to model contexts of predicates, where the numbers in the second column indicate the dimensionality of each layer of DBNs in the DCA model. The plus symbol means the concatenation of representation of predicate v and its filler words w. As shown in the table, these numbers are designed to decrease al-

[4] Note that during rules application, the filler words is not restricted (like LDA based models) to these 30K tokens, thanks to the pre-trained word embeddings with a vocabulary of 2.2M words.

[5] http://u.cs.biu.ac.il/~nlp/resources/downloads/annotation-of-rule-applications/

[6] We choose to keep the top 1500 (23% of all predicates) candidate predicates as V_R for each entailing predicate V_L, which follows the same settings from [12], where 500 of 2155 (23%) predicates were kept.

[7] https://nlp.stanford.edu/projects/glove/

Table 2
The structures of DBNs with various depth evaluated on the development set.

Depth	Structure
4	30000-3000-300(+300)-50
5	30000-4200-600(+300)-200-50
6	30000-7500-1900-500(+300)-200-50
7	30000-9000-2700-850-300(+300)-180-50
8	30000-10000-3300-1100-400(+300)-240-80-50
9	30000-12000-4800-1920-770-300(+300)-250-100-50
10	30000-13000-6000-2800-1300-600(+300)-400-180-85-50
11	30000-15000-7500-3800-1900-1000-500(+300)-400-200-100-50
12	30000-15500-8150-4200-2200-1200-600(+300)-480-260-135-75-50
13	30000-16500-11000-9200-5100-2800-1500-800-450(+300)-420-230-120-50

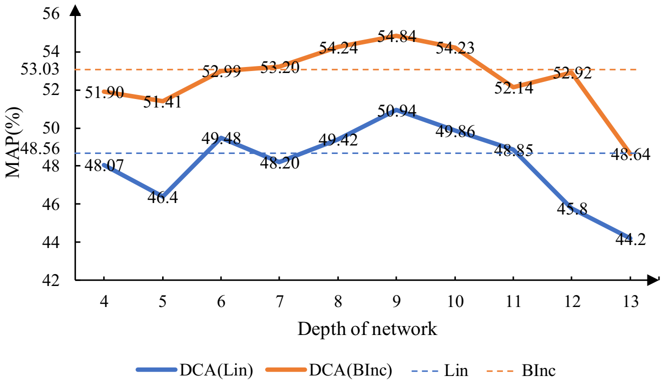


Fig. 8. DCA's performance on the development set with various depth.

most uniformly.[8] For example, in the 11-layer DBNs, the number of units in each layer is almost half the number in the previous layer.

The performance of DCAs with different depth is shown in Fig. 8, where the orange solid line indicates the *MAP* of DCAs with the extended BInc similarity at various depth, while the blue solid line indicates the performance of DCAs with the extended Lin similarity. The orange and blue dash lines show the performance of vanilla BInc [9] on the development set and original Lin similarity (i.e. DIRT [5]), respectively. We can observe that DCAs with BInc similarity perform better than those with Lin similarity which could not recognize the direction of entailment. DCAs with depth from 7 to 10 achieve higher *MAP* than BInc, where the one with depth 9 works best. Then, we fixed the depth to 9 and tuned the dimensionality of each layer, finally got the structure described in Fig. 7. DCA with that structure obtained 55.93 (with BInc similarity) and 52.13 (with Lin similarity) of *MAP* on the development set. As illustrated in the figure, the performance increases with the growth of depth before 9, and decreases after adding more layers. We suspect that DBN part of the model with lower depth might reduce the dimensionality too fast to capture the useful information from the representations of *w* and *v*, while if we add too many layers to the model, the effect of error cascading would cause a performance regression.

4.4.2. Quantitative analysis

We implemented several previous entailment rules mining approaches introduced in Section 2 as our baselines. The performance on the test set of these methods lies in Figs. 9 (evaluated by *MAP*) and 10 (evaluated by *Coverage*), where DIRT represents the vanilla DIRT inference rule extraction method [5], whose similarity is computed by Eq. (4). DIRT could not identify the direction of entailment and is context-unaware. BInc represents the original BInc [9] method without the contextual extension, which is a directionality-aware but context-insensitive approach. DC (Double Conditioning) [10], SC (Single Conditioning) [11], LDA(Lin) and LDA(BInc) [12] are several LDA-driven context-sensitive methods. DC replaces Eq. (9) with Eq. (15), which is unable to identify the direction of entailment rules, either. While SC replaces Eq. (9) with Eq. (16), resulting in a directionality-aware method. LDA(BInc) [12] replaces Eq. (3) with an LDA topic model. Both DCA(Lin) and LDA(Lin) use $Lin_t(v_L, v_R)$ instead of $BInc_t(v_L, v_R)$in Eq. (9).

$$sim_{DC}(v_L, v_R, w) = \sum_t [P(t|v_L, w)P(t|v_R, w)]. \quad (15)$$

$$sim_{SC}(v_L, v_R, w) = \sum_t [P(t|v_L, w)P(t|v_R)]. \quad (16)$$

It can be observed from these figures that our DCA-based inference rules mining approach outperforms all baseline methods in the same dataset under both *MAP* and *Coverage* metrics.

We could draw the following conclusions from these results. Firstly, most directionality-aware approaches (e.g. BInc, SC, DCA(BInc)) perform better than their corresponding directionality-unaware methods (e.g. DIRT, DC, DCA(Lin)) because textual entailment is actually a uni-directional relationship. Secondly, context modeling using either LDA or DCA could improve the correctness of inference rules mining because inference rules are not universally applicable. Thirdly, DCA is about 2% *MAP* better than LDA to model contexts of predicates in this task.[9] Fourthly, rules extracted by DCA-driven approaches could cover more than double the amount of samples in the Zeichner's dataset by LDA based methods, which means our method has a wider range of applications.[10]

In addition to the comparison between DCA and those baseline methods, we also investigated the effectiveness of the word vectors based filler words representation. The results of this experiment are depicted in Fig. 11, where BInc and LDA are baselines, other DCA(·)s indicate the proposed method under various settings of word vectors.

We first removed all semantic knowledge from pre-trained word embeddings by feeding DCA with randomly initialized word vectors according to Gaussian distribution, to evaluate the context modeling capability of DBN itself, resulting in 53.72 (MAP%), which is higher than the original BInc approach and competitive with LDA, with considering the noise introduced. This phenomenon demonstrates the effectiveness of DBNs in the DCA model. Next, we explored the performance of different word vectors. By using the infrastructure of Indra [23], we obtained three word embed-

[8] The search space of structures of DBNs is too large to explore. In this paper, we followed the 'uniformly decreasing' principle to design them instead of randomly picking.

[9] Student's T test is performed between DCA models and their corresponding LDA equivalents with $p < 0.01$, indicating that the improvement is significant.

[10] DIRT and BInc also have high *Coverage* because they don't consider the contexts of rules during applying time, resulting in a lower correctness (*MAP*).

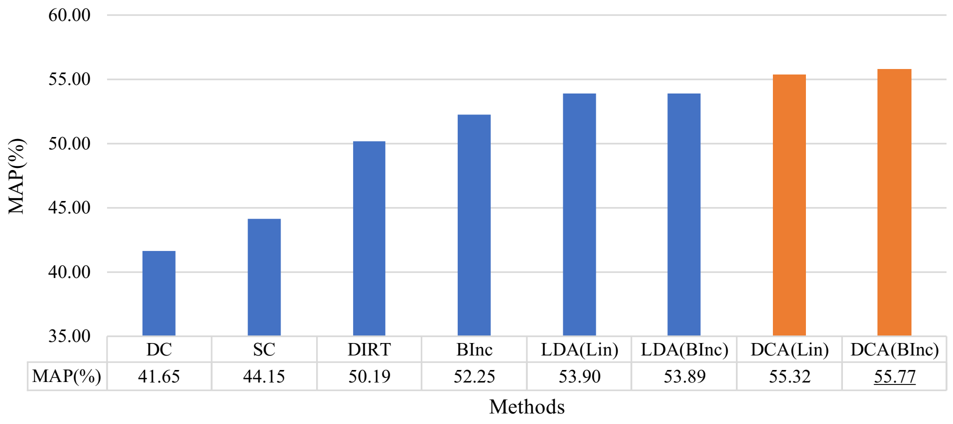


Fig. 9. *MAP*(%) of each method.

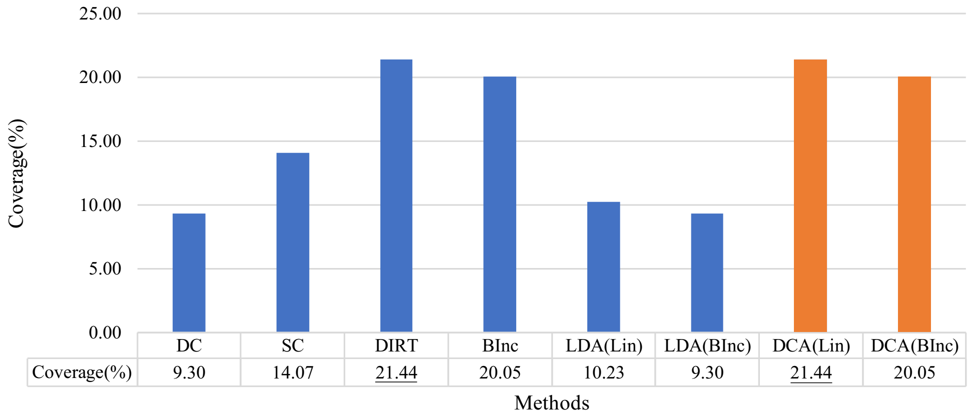


Fig. 10. *Coverage*(%) of each method.

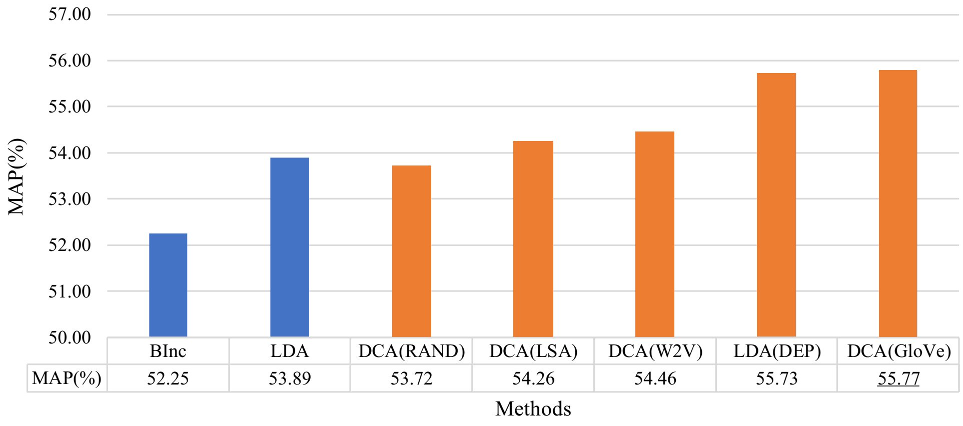


Fig. 11. *MAP*(%) of DCAs under different settings of word vectors representations.

dings trained on Wikipedia, namely, LSA (Latent Semantic Analysis [24], which is obtained using singular value decomposition on word-document matrix), W2V (Word2Vec [25], which is trained using continuous bag-of-words and skip-gram architectures), and DEP (Dependency-based word embeddings [26], which is similar with W2V, but the contexts are defined according to dependency parsing trees). Although LSA gets the lowest MAP score among these pre-trained word vectors, it still performs better than both LDA and RAND, indicating the usefulness of semantic knowledge from word vectors. Compared with W2V, DEP involves additional dependency information and gets better performance which is even comparable to GloVe, which proved capable to catch entailment relation in natural language inference tasks [20–22].

In conclusion, the superiority of our DCA based approach comes from three aspects: Firstly, the network structure of DCA, where DBNs learn the latent low dimensional topic representation of predicates and their contexts, is carefully designed to take inputs from both statistical information from dataset and semantics knowledge stored in pre-trained word vectors. Secondly, thanks to the word embeddings based filler words representation, our DCA based method could overcome the restriction of LDA driven approaches that all filler words must have appeared in the training

Table 3
Examples of correct predictions and false positive and false negative errors.

Entailment rules	Corresponding applications	Comments
X defeat Y → *X beat Y*	Richard defeat Marcus. → Richard beat Marcus.	True positive (paraphrase)
X join Y → *X be a member of Y*	Anderson join the U.S. Air Force. → Anderson be a member of the U.S. Air Force.	True positive (inference)
X shoot Y ↛ *X throw Y*	Jenna shoot Frank. ↛ Jenna throw Frank.	True negative (irrelevance)
X influence Y ↛ *X interfere with Y*	The judge influence the outcome. ↛ The judge interfere with the outcome.	True negative (backward entailment)
X speak Y ↛ *X don't speak Y*	The staff speak English. ↛ The staff don't speak English.	False positive (negation)
X be better for Y ↛ *X be bad for Y*	hemp be better for our planet. ↛ hemp be bad for our planet.	False positive (antonym)
X need to keep Y ↛ *X get to keep Y*	Microsoft need to keep developers. ↛ Microsoft get to keep developers.	False positive (preprocessing)
X paint Y → *X present Y*	A picture paint a thousand words. → A picture present a thousand words.	False negative (metaphor)

set. Thirdly, the directionality-aware method driven by distributional hypothesis, i.e., BInc, captures the entailment phenomenon from texts. In summary, the proposed method retains the high coverage advantage of context-unaware methods as a context-aware approach with high accuracy.

4.4.3. Qualitative analysis and error analysis

Besides the above quantitative analysis, we also investigated the extracted rules, especially the errors our method made. Some of them are listed in Table 3, where false positive means that examples are labeled as incorrect by Zeichner but got a rather high confidence score, and false negative represents examples labeled as correct with a very low confidence score, while true positive and true negative indicate the confidence reported by our method is consistent with Zeichner's annotation.

Most of the extracted entailment rules are paraphrase, e.g., '*X defeat Y* → *X beat Y*', and inference, e.g., '*X join Y*→ *X be a member of Y*', which are denoted as 'true positive' in Table 3, whilst true negative samples consist of irrelevance, e.g., '*X shoot Y* ↛ *X throw Y*', and backward entailment, which means the two predicates are relevant but entailment relation only exists in the backward direction, e.g., '*X influence Y*↛ *X interfere with Y*' but '*X influence Y*← *X interfere with Y*' under the context <*the judge, the outcome*>. These samples show that our method could capture the entailment relation and detect its direction effectively.

Many of the false positive errors come from antonyms or predicates with the opposite meanings but sharing common contexts. For example, 'speak' and 'don't speak' (the first example in Table 3) have almost the same contexts and similar statistical representation. Therefore, they have similar $P(t|v, w)$ and thus high confidence score F. This is inevitable for all methods based on the distributional hypothesis which asserts that similar contexts lead to entailment relationship. We plan to resolve this in the future work.

Furthermore, the predicate normalization introduced in Section 4.1 caused another rare but existing false positive error, e.g. the seventh example in Table 3. Both 'need to keep' and 'get to keep' were normalized as 'keep' to enrich the diversity of contexts for better confidence computation. In fact, we have tried to get rid of the predicate normalization but it caused a performance regression due to lack of diversity of contexts. This problem might be fixed by complementing more training corpora.

The false negative error might be caused by metaphor or extended usage of a predicate. For example, '*X paint Y* → *X present Y*' is not true in most contexts except <*a picture, a thousand words*>. Thus, the contexts of these predicates might vary a lot, resulting in a low confidence score.

5. Conclusion and future work

In this paper, we proposed a novel DBN-based model named Deep Contextual Architecture (DCA) to mine entailment rules from text corpora. On the one hand, our method combined semantic meanings from pre-trained word vectors and statistical information from text corpora, which highly improved inference rules mining. On the other hand, by leveraging the low dimensional data representation ability of DBNs, our neural networks based method achieved state-of-the-art performance on this task by both *MAP* and *Coverage* metrics. Evaluation on publicly available datasets demonstrated the advantage of our approach on entailment rules mining.

Meanwhile, there is still room to improve in this model. Firstly, DCA employs an AverageNN (Fig. 6) to represent the filler words containing multiple tokens, which could be replaced by pre-trained phrase or sentence level LSTM encoders like [27]. Secondly, it is possible to involve attention mechanism to this model, e.g., self-attention could be useful to model predicates or filler words containing multiple tokens, while inter-attention between predicates and their contexts could better model interaction between them. Thirdly, it is rational and possible to merge individual inference rules mined by DCA into a knowledge graph and use a navigation mechanism similar with [28] to deduce more rules according to it. Lastly, there are remaining challenges like how to distinguish antonyms from entailment when the contexts of predicates are similar. We plan to research along these directions in the future.

Acknowledgments

We are grateful to Rekia Kadari, Qinyu Yin, Le Qi, Qi shi and the anonymous reviewers for their valuable comments and insightful feedbacks.

This work was supported by the National Basic Research Program of China (973 Program) [Grant No. 2014CB340503] and the National Natural Science Foundation of China [Grant Nos. 61472105 and 61502120].

References

[1] B. Wang, D. Zheng, X. Wang, S. Zhao, T. Zhao, Multiple-choice question answering based on textual entailment, Acta Sci. Nat. Univ. Pekinensis 52 (1) (2016) 134–140, doi:10.13209/j.0479-8023.2016.017.

[2] N. Dhruva, O. Ferschke, I. Gurevych, Solving open-domain multiple choice questions with textual entailment and text similarity measures., in: Proceedings of the CLEF (Working Notes), 2014, pp. 1375–1385. http://ceur-ws.org/Vol-1180/CLEF2014wn-QA-DhruvaEt2014.pdf.

[3] R. Pasunuru, H. Guo, M. Bansal, Towards improving abstractive summarization via entailment generation, in: Proceedings of the EMNLP 2017, 2017, p. 27. http://www.aclweb.org/anthology/W/W17/W17-45.pdf#page=39.

[4] P. Xu, R. Frank, J. Kasai, O. Rambow, Tag parser evaluation using textual entailments, in: Proceedings of the Thirteenth International Workshop on Tree Adjoining Grammars and Related Formalisms, 2017, pp. 132–141.

[5] D. Lin, P. Pantel, DIRT@ SBT@ discovery of inference rules from text, in: Proceedings of the seventh ACM SIGKDD International Conference on Knowledge discovery and data mining, ACM, 2001, pp. 323–328. http://dl.acm.org/citation.cfm?id=502559.

[6] I. Szpektor, E. Shnarch, I. Dagan, Instance-based evaluation of entailment rule acquisition (2007). http://eprints.pascal-network.org/archive/00002988/.

[7] I. Szpektor, H. Tanev, D. Dagan, B. Coppola, et al., Scaling web-based acquisition of entailment relations (2004). http://eprints.pascal-network.org/archive/00000797/.

[8] R. Bhagat, P. Pantel, E.H. Hovy, M. Rey, LEDIR: an unsupervised algorithm for learning directionality of inference rules., in: Proceedings of the EMNLP-CoNLL, Citeseer, 2007, pp. 161–170.

[9] I. Szpektor, I. Dagan, Learning entailment rules for unary templates, in: Proceedings of the Twenty-second International Conference on Computational Linguistics-Volume 1, Association for Computational Linguistics, 2008, pp. 849–856. http://dl.acm.org/citation.cfm?id=1599188.

[10] A. Ritter, O. Etzioni, et al., A latent Dirichlet allocation method for selectional preferences, in: Proceedings of the Forty-eighth Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2010, pp. 424–434. http://dl.acm.org/citation.cfm?id=1858725.

[11] G. Dinu, M. Lapata, Topic models for meaning similarity in context, in: Proceedings of the Twenty-third International Conference on Computational Linguistics: Posters, Association for Computational Linguistics, 2010, pp. 250–258. http://dl.acm.org/citation.cfm?id=1944595.

[12] O. Melamud, J. Berant, I. Dagan, J. Goldberger, I. Szpektor, A two level model for context sensitive inference rules., in: Proceedings of the ACL (1), 2013, pp. 1331–1340. http://anthology.aclweb.org/P/P13/P13-1131.pdf.

[13] T. Mikolov, W.-T. Yih, G. Zweig, Linguistic regularities in continuous space word representations., in: Proceedings of the hlt-Naacl, 13, 2013, pp. 746–751. http://www.aclweb.org/anthology/N13-1#page=784.

[14] J. Pennington, R. Socher, C.D. Manning, Glove: global vectors for word representation, in: Proceedings of the EMNLP, 14, 2014, pp. 1532–1543, doi:10.3115/v1/d14-1162. http://llcao.net/cu-deeplearning15/presentation/nn-pres.pdf.

[15] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[16] L. Maaloe, M. Arngren, O. Winther, Deep belief nets for topic modeling, (2015). arXiv:1501.04325 [cs, stat], http://arxiv.org/abs/1501.04325.

[17] Z.S. Harris, Distributional structure, Word 10 (2–3) (1954) 146–162.

[18] A. Fader, S. Soderland, O. Etzioni, Identifying relations for open information extraction, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2011, pp. 1535–1545. http://dl.acm.org/citation.cfm?id=2145596.

[19] N. Zeichner, J. Berant, I. Dagan, Crowdsourcing inference-rule evaluation, in: Proceedings of the Fiftieth Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2, Association for Computational Linguistics, 2012, pp. 156–160. http://dl.acm.org/citation.cfm?id=2390704.

[20] A.P. Parikh, O. Tckstrm, D. Das, J. Uszkoreit, A decomposable attention model for natural language inference, (2016). arXiv:1606.01933 [cs], http://arxiv.org/abs/1606.01933.

[21] Q. Chen, X. Zhu, Z. Ling, S. Wei, H. Jiang, Enhancing and combining sequential and tree LSTM for natural language inference, (2017). arXiv:1609.06038 [cs], http://arxiv.org/abs/1609.06038.

[22] R. Ghaeini, S.A. Hasan, V. Datla, J. Liu, K. Lee, A. Qadir, Y. Ling, A. Prakash, X.Z. Fern, O. Farri, DR-BiLSTM: dependent reading bidirectional LSTM for natural language inference, arXiv:1802.05577 [cs] (2018). http://arxiv.org/abs/1802.05577.

[23] J.E. Sales, L. Souza, S. Barzegar, B. Davis, A. Freitas, S. Handschuh, Indra: a word embedding and semantic relatedness server, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), 2018. Miyazaki, Japan, May

[24] T.K. Landauer, P.W. Foltz, D. Laham, An introduction to latent semantic analysis, Discour. Process. 25 (2–3) (1998) 259–284.

[25] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger (Eds.), Proceedings of the Advances in Neural Information Processing Systems 26, Curran Associates, Inc., 2013, pp. 3111–3119. http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

[26] O. Levy, Y. Goldberg, Dependency-based word embeddings, in: Proceedings of the Fifty-second Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2, 2014, pp. 302–308.

[27] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, A. Bordes, Supervised learning of universal sentence representations from natural language inference data, arXiv:1705.02364 [cs] (2017). http://arxiv.org/abs/1705.02364.

[28] V.S. Silva, A. Freitas, S. Handschuh, Recognizing and justifying text entailment through distributional navigation on definition graphs, in: Proceedings of the AAAI, 2018.

Maosheng Guo is a Ph.D. Candidate in Harbin Institute of Technology, P.R. China. He is a member of the Research Center for Social Computing and Information Retrieval (HIT-SCIR) in Harbin Institute of Technology, China. Textual Entailment and Natural Language Inference are his main topics of interest.

Yu Zhang is a Professor in Harbin Institute of Technology, P.R. China. He is a member of the Research Center for Social Computing and Information Retrieval (HIT-SCIR) in Harbin Institute of Technology, China. Question Answering Systems and Information Retrieval are his main topics of interest.

Dezhi Zhao is a member of the Research Center for Social Computing and Information Retrieval (HIT-SCIR) in Harbin Institute of Technology, China.Question Answering Systems and Textual Entailment are his main topics of interest.

Ting Liu is a vice dean and full professor of the School of Computer Science, Harbin Institute of Technology, P. R. China. He is director of Research Center for Social Computing and Information Retrieval (HIT-SCIR). Information Retrieval and Natural Language Processing are his main topics of interest.