Context-Sensitive Inference Rule Discovery: A Graph-Based Method

Xianpei Han Le Sun
State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences
Beijing, China
{xianpei, sunle}@nfs.iscas.ac.cn

Abstract

Inference rule discovery aims to identify entailment relations between predicates, e.g., '*X acquire Y* → *X purchase Y*' and '*X is author of Y* → *X write Y*'. Traditional methods discover inference rules by computing distributional similarities between predicates, with each predicate is represented as one or more feature vectors of its instantiations. These methods, however, have two main drawbacks. Firstly, these methods are mostly *context-insensitive*, cannot accurately measure the similarity between two predicates in a specific context. Secondly, traditional methods usually model predicates *independently*, ignore the rich inter-dependencies between predicates. To address the above two issues, this paper proposes a graph-based method, which can discover inference rules by effectively modelling and exploiting both the context and the inter-dependencies between predicates. Specifically, we propose a graph-based representation—*Predicate Graph*, which can capture the semantic relevance between predicates using both the predicate-feature co-occurrence statistics and the inter-dependencies between predicates. Based on the predicate graph, we propose a context-sensitive random walk algorithm, which can learn context-specific predicate representations by distinguishing context-relevant information from context-irrelevant information. Experimental results show that our method significantly outperforms traditional inference rule discovery methods.

1 Introduction

Inference rule discovery aims to identify entailment relations between predicates, such as '*X acquire Y* → *X purchase Y*' and '*X is author of Y* → *X write Y*', with each predicate is a textual pattern with (two) variable slots (*X* and *Y* in above). Inference rules are important in many fields such as Question Answering (Ravichandran and Hovy, 2002), Textual Entailment (Dagan et al., 2006) and Information Extraction (Hearst, 1992). For example, given the problem "*Which company purchases WhatsApp?*", a QA system can extract the answer "*Facebook*" from the sentence "*Facebook acquires WhatsApp for $19 billion*" based on the inference rule '*X acquire Y* → *X purchase Y*'.

Given a set of predicates and their instantiations in a large corpus, most traditional methods identify inference rules by computing distributional similarities between predicates, where each predicate is represented as one or more feature vectors of its variable instantiations. For example, given the predicates and their instantiations in Figure 1, we can represent '*X acquire Y*' as {*X='Google', Y='YouTube', X='children', Y='skill'*} and measure the similarity between '*X acquire Y*' and '*X purchase Y*' based on their common features {*X='Google', Y='YouTube'*}. To achieve the above goal, many similarity measures have been proposed for inference rule discovery, such as *DIRT* Similarity (Lin and Pantel, 2001), *Balanced-Inclusion* similarity (Szpektor and Dagan, 2008) and *Soft Set Inclusion* similarity (Nakashole et al., 2012), etc.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2902–2911, Osaka, Japan, December 11-17 2016.

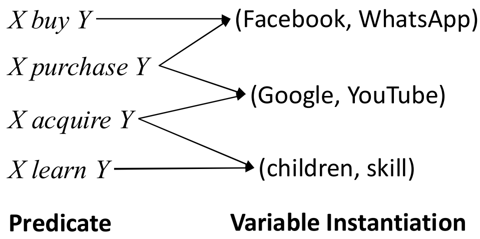

Figure 1. Some predicates and their variable instantiations

These distributional similarity based methods, however, have two main drawbacks:

Firstly, these methods are mostly *context-insensitive*, cannot accurately measure the similarity between two predicates in a specific context. Due to the ambiguity of predicates, a predicate may have different meanings under different contexts (In this paper, as the same as Melamud et al. (2013), the context of a predicate is specified by the predicate's given arguments). For example, the predicate '*X acquire Y*' should have different meanings under context (*Google, YouTube*) and context (*children, skill*), because it corresponds to two different senses of *acquire* in these two contexts. Unfortunately, traditional methods mostly use the same representation to represent a predicate in different contexts, therefore may learn invalid inference rules. For example, given two predicates '*X acquire Y*' and '*X purchase Y*', traditional context-insensitive methods will return the same similarity between them in contexts (*Google, YouTube*) and (*children, skill*). However, '*X acquire Y* → *X purchase Y*' is not a valid rule in context (*children, skill*). Based on the above discussion, we believe that context-specific predicate representation is critical to the success of inference rule discovery.

Secondly, traditional methods usually model predicates *independently*, ignore the rich inter-dependencies between predicates. It is clear though, that there are rich inter-dependencies between predicates. For example, '*X buy Y*' is a synonym of '*X purchase Y*', and '*Y be acquired by X*' is the passive form of '*X acquire Y*'. These dependencies can be exploited to enhance inference rule discovery in many ways. For instance, we can collect richer instantiation co-occurrence statistics per predicate by combining the statistics of semantically similar predicates, or enforce global coherence between the representations of semantically similar predicates. Ignoring these useful inter-dependencies, traditional methods often suffer from the data sparsity problem. For example, if we represent predicates using only their instantiations, we cannot identify the inference rule '*X acquire Y* → *X buy Y*' in Figure 1, because '*X acquire Y*' and '*X buy Y*' don't share any common features.

To address the above two problems, this paper proposes a graph-based method, which can effectively exploit both the context of a predicate and the inter-dependencies between predicates for accurate inference rule discovery. Specifically, we propose a graph-based representation, called *Predicate Graph*, which can capture the semantic relevance between predicates and features by encoding both the predicate-feature co-occurrence statistics and the rich inter-dependencies between predicates. For example, the predicate graph will model the semantic relevance between the predicate '*X buy Y*' and the feature *X*='*Google*' in Figure 1 by taking advantage of the synonym relation between '*X buy Y*' and '*X purchase Y*'. Based on the predicate graph, we propose a context-sensitive random walk algorithm, which can learn context-specific predicate representations by distinguishing context-relevant information from context-irrelevant information. For example, to learn the representation of '*X acquire Y*' under context (*people, language*), our method will identify (*Google, YouTube*) and (*Facebook, WhatsApp*) in Figure 1 as context-irrelevant and will identify (*children, skill*) as context-relevant.

We have evaluated our method on a publicly available dataset. The experimental results show that, using context-specific predicate representations and taking advantage of inter-dependencies between predicates, our method can significantly outperform traditional inference rule discovery methods.

This paper is structured as follows. Section 2 briefly reviews related work. Section 3 describes the proposed method. Section 4 presents and discusses experimental results. Finally we conclude this paper in Section 5.

2 Related Work

Many approaches have been proposed for inference rule discovery, and most of them are distributional similarity based methods. Based on the distributional hypothesis, traditional methods differ in their feature representations and their similarity measures. For predicate representation, some methods represent predicates using one feature vector, where each feature is a pair of argument instantiations such as *X*=‘*children*’-*Y*=‘*skill*’(Szpektor et al., 2004; Sekine, 2005; Nakashole et al., 2012; Dutta et al., 2015); some methods represent predicates using two or more feature vectors, one for each argument slot (Lin and Pantel, 2001; Bhagat et al., 2007), e.g., one feature vector for slot *X* and one for slot *Y*. To compute the similarity between predicates, many similarity measures have been proposed, such as *DIRT* Similarity (Lin and Pantel, 2001), *Balanced-Inclusion* similarity (Szpektor and Dagan, 2008) and *Soften Set Inclusion* similarity (Nakashole et al., 2012), etc. Hashimoto et al. (2009) proposed a conditional probability based directional similarity measure to acquire verb entailment pairs on a large scale corpus. As discussed in above, the main drawbacks of these methods are that they are *context-insensitive* and model predicates *independently*.

Having observed that the meaning of a predicate is context-sensitive, several recent methods try to model the context of a predicate using class-based model or latent topic model. The class-based models represent the context of a predicate using ontological type signatures (Pantel et al., 2007; Nakashole et al., 2012), e.g., <*singer, song*> for ‘*X sing Y*’, based on the assumption that two predicates in a rule must have the same type signature. The shortcomings of the class-based context models are that they need a fine-grained ontology and it is often very challenging to determine the fine-grained types of arguments. The latent topic based model represents the context of a predicate as a vector in a low dimensional space, such as the LSA-based model (Szpektor et al., 2008) and the LDA based model (Ritter et al., 2010; Dinu and Lapata, 2010). Based on the context vector, the similarity between two predicates are computed by combining both the context vector similarity and the feature vector similarity (Szpektor et al., 2008), or by first learning predicate similarity per topic, then combining the per-topic similarities using context vector (Melamud et al., 2013). Currently, most of the context-sensitive methods focus on developing an extra context model, by contrast our method focuses on the learning of context-specific predicate representations, without the need of an extra context model.

Recent research has also investigated the jointly learning of inference rules. Kok and Domingos (2008) and Yates and Etzioni (2009) learned inference rules by clustering predicates using relational clustering algorithms. Berant et al.(2010) and Berant et al.(2011) proposed two global learning methods, which first classify each pair of predicates using a local classifier, then these local results are globally rescored using *Integer Linear Programming(ILP)* algorithm. Nakashole et al. (2012) proposed a prefix-tree mining algorithm, which can arrange predicates into a semantic taxonomy. The current joint learning methods mostly employ a meta-classification schema, i.e., the inter-dependencies between predicates are used to adjust the pair-wise predicate similarities, therefore their predicate representations still suffer from the data sparsity problem. In contrast our method exploits the inter-dependencies for better predicate representation, which can effectively resolve the data sparsity problem.

3 Graph-Based Context-Sensitive Inference Rule Discovery

This section describes our graph-based method for context-sensitive inference rule discovery. We first construct a graph, which can effectively capture the semantic relevance between predicates and features. Then we propose a context-sensitive random walk algorithm, which can learn accurate, context-sensitive predicate representations. Finally, we discover inference rules by computing similarities between context-sensitive predicate representations.

3.1 The Predicate Graph Representation

Generally, there are two kinds of information which can be exploited to represent a predicate: 1) its variable instantiations in a corpus, such as the instantiations (*Google, YouTube*) and (*children, skill*) in Figure 1 for representing predicate ‘*X acquire Y*’; 2) the information from semantically similar predicates, for example, the instantiation (*Google, YouTube*) of ‘*X purchase Y*’ can be used to enrich the representation of ‘*X buy Y*’. In this paper, we uniformly encode the above two kinds of information using a graph representation, named *Predicate Graph*, which is defined as follows:

*A **Predicate Graph** is a weighted graph **G=(V, E)**, where the node set **V** contains all predicates and all features of these predicates; each edge between a predicate and a feature represents a co-occurrence relation between them; each edge between two predicates represents a semantic-dependent relation between them.*

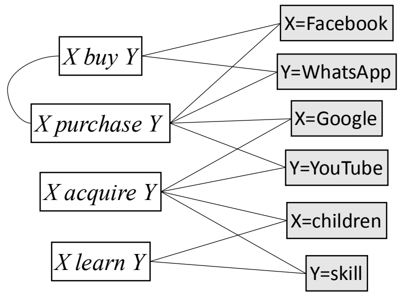

Figure 2. A predicate graph demo

Figure 2 demonstrates a predicate graph, which is constructed using the information in Figure 1. We can see that, the instantiation information of predicates is modelled by co-occurrence edges between (*predicate, feature*), such as the edges between ('*X buy Y*', *Y*='*WhatsApp*') and between ('*X buy Y*', *X*='*Facebook*'). The inter-dependencies between predicates are modelled by semantic-dependent edges between predicates, e.g., the edge between ('*X buy Y*', '*X purchase Y*'). Based on the co-occurrence and the semantic-dependent edges, both the explicit and the implicit semantic relevance between predicates and features can be captured using the paths between them. For example, the implicit semantic relevance between the feature *X*='*Google*' and the predicate '*X buy Y*' can be modelled through the path *X*='*Google*'–'*X purchase Y*'–'*X buy Y*'.

The Construction of Predicate Graph. Given a set of predicates and their instantiations in a large corpus, we construct predicate graph by first adding all predicates and all features as nodes, then we link these nodes using the following two types of edges:

- *Co-occurrence Edge*. We take each argument instantiation of a predicate p as a feature f of p and add a co-occurrence edge between them, the pointwise mutual information (PMI) between p and f is used as the edge's weight;
- *Semantic-Dependent Edge*. To encode inter-dependencies between predicates, we add a semantic-dependent edge between a predicate p and each of its semantically similar predicates. We use the same edge weight α for all semantic-dependent edges, and which will be empirically tuned. Specifically, given a predicate p, we find its semantically similar predicates as follows: 1) we identify its active/passive verb form as its semantically similar predicate, e.g., '*Y be acquired by X*' will be identified as a semantically similar predicate of '*X acquire Y*'; 2) we generate semantically similar predicate candidates by replacing each verb/noun in the predicate p with its synonyms/hypernyms in WordNet 3.0. If a predicate candidate is a valid predicate (i.e., it is one of the given predicates), we take it as a semantically similar predicate of p. For example, ('*X buy Y*', '*X purchase Y*') and ('*X be maker of Y*', '*X be creator of Y*') will be identified semantically similar using the synonym relations between (*buy*, *purchase*) and between (*maker*, *creator*).

3.2 Context-Sensitive Random Walk Algorithm

In this section, we describe how to accurately represent a predicate in a specific context. Specifically, given a predicate p, its context c and all features $\{f_1, f_2, \ldots, f_n\}$, we represent predicate p as a vector:

$$\vec{\boldsymbol{v}}_p^c = (w_{p1}^c, w_{p2}^c, \ldots, w_{pn}^c)$$

where w_{pi}^c is the relevance score between predicate p and feature f_i under context c. In following we first develop a context-insensitive random walk algorithm which can estimate context-insensitive relevance score between a predicate p and a feature f, then we extend the algorithm by taking context into consideration. For simplicity, we assign each node in predicate graph $G=(V, E)$ an integer index from 1 to $|V|$, and use it to represent the node.

Context-Insensitive Random Walk. Given a predicate graph $G=(V, E)$, the relevance score between a predicate p and a feature f can be naturally modelled as the relevance score between the two nodes in G corresponding to p and f. Estimating relevance score between two nodes in a graph is one of the fundamental tasks in graph mining, and many algorithms have been developed. In this paper we estimate the context-insensitive relevance score between two nodes using one of the most widely used algorithm – *Random Walk with Restart (RWR)* (Tong et al., 2006), which can be fast computed and has been successfully used in many applications, like personalized PageRank (Haveliwala, 2003), image retrieval (He et al., 2004), etc.

Specifically, *RWR* models the relevance score between node *i* and node *j* in a graph *G* as the steady-state probability $r_{i,j}$, i.e., the probability of a random walk starts from node *i* will end at node *j*. For example, the relevance between ('*X acquire Y*', *X*='*Facebook*') in Figure 2 will be computed by starting random walks from the predicate node '*X acquire Y*', then estimate the probability of these random walks ending at the feature node *X*='*Facebook*'.

The random walk used in *RWR* is specified as follows: consider a random particle that starts from node *s* that indicates predicate *p*, at each step the particle iteratively transmits to its neighbourhood with probability that is proportional to their edge weights, and it also has a restart probability $\lambda \in [0, 1]$ to return to the start node *s*:

$$\mathrm{P}(\mathrm{i} \rightarrow \mathrm{j}) = \begin{cases} (1-\lambda)\dfrac{w_{ij}}{\sum_k w_{ik}} & \text{transmit to neighorhood } j \\ \lambda & \text{restart to start node } s \end{cases}$$

where $\mathrm{P}(\mathrm{i} \rightarrow \mathrm{j})$ is the probability of transmit from node *i* to node *j* at each step, and w_{ij} is the edge weight between node *i* and node *j*. *RWR* can also be written in matrix form:

$$\vec{r}_s = (1-\lambda)\mathbf{M}\vec{r}_s + \lambda\vec{e}_s$$

where $\vec{r}_s$ is the n×1 relevance score vector, with $r_{s,j}$ is the relevance score of node *j* with respect to start node *s*, and $\vec{e}_s$ is n×1 starting vector with the s^{th} element 1 and 0 for others; **M** is the neighbourhood transition matrix with $\mathrm{M}_{\mathrm{ij}} = w_{ji}/\sum_k w_{jk}$.

Using *RWR*, the relevance score between a predicate *p* and a feature *f* can effectively summarize the semantic relevance information between them by exploiting the global structure of predicate graph. For example, in Figure 2 all the paths between '*X buy Y*' and *X='Facebook'* will be used to estimate the relevance score between them, such as the direct edge *'X buy Y'— X='Facebook'* and the indirect path *'X buy Y'—'X purchase Y'— X='Facebook'*. To demonstrate the effect of *RWR*, Table 1 shows the state-steady probability of the random walk starting from '*X acquire Y*'. We can see that *RWR* can effectively exploit both the inter-dependencies between predicates and the predicate-feature co-occurrence information. For example, although '*X acquire Y*' doesn't co-occur with *X='Facebook'* in Figure 2, *RWR* can still estimate the relevance score between them as *0.045*.

Feature \ Context	No Context	X=Microsoft Y=Nokia	X=people Y=language
X=Facebook	*0.045*	*0.055*	*0.003*
Y=WhatsApp	*0.045*	*0.055*	*0.003*
X=Google	*0.064*	*0.092*	*0.073*
Y=YouTube	*0.064*	*0.092*	*0.073*
X=children	*0.119*	*0.080*	*0.163*
Y=skill	*0.119*	*0.080*	*0.163*

Table 1. The representations of 'X acquire Y' in different contexts (λ=0.1 and semantic-dependent edge weight = 0.5)

Context-Sensitive Random Walk. The main problem of the above random walk algorithm is that it is context-insensitive, cannot accurately represent a predicate in different contexts. For example, the above algorithm will return the same representation for '*X acquire Y*' in contexts (*Microsoft, Nokia*) and (*people, language*), although it corresponds to different senses of *acquire*.

To learn context-specific predicate representations, we extend *RWR* algorithm by also taking context into consideration. Specifically, the start point of our algorithm is to distinguish context relevant information from context irrelevant information. For example, to represent '*X acquire Y*' in the context (*peo-*

ple, language), the features *X*='*Facebook*', *X*='*Google*', *Y*='*WhatsApp*' and *Y*='*YouTube*' will be identified as context-irrelevant and their relevance scores will be reduced, meanwhile the features *X*='*children*' and *Y*='*skill*' will be identified as context-relevant and their relevance scores will be increased. To achieve the above goal, we revise the transition probability of *RWR* using a context-sensitive node-dependent restart probability:

$$P(i \rightarrow j|c) = \begin{cases} (1-\lambda_{c,i})\dfrac{w_{ij}}{\sum_i w_{ik}} & \text{transmit to neighorhood } j \\ \lambda_{c,i} & \text{restart to start node } s \end{cases}$$

where $\lambda_{c,i}$ is the restart probability at node *i* in context *c*, which depends on the context relevance between node *i* and context *c*. For instance, in Figure 1, to learn the representation of '*X acquire Y*' in context (*people, language*), our method will set a high restart probability to context-irrelevant nodes *X='Facebook', X='Google', Y='WhatsApp'* and *Y='YouTube'*, in contrast our method will set a low restart probability to context-relevant nodes *X='children'* and *Y='skill'*. Based on the context-sensitive random walk, we can easily identify context-relevant information: once a random walk hits a context-irrelevant node, it will jump to the start node, then the relevance scores of all nodes which are semantically similar to the context-irrelevant node will be reduced. The context-sensitive random walk algorithm can also be written in matrix form:

$$\vec{r}_s^{\,c} = \mathbf{M}(\mathbf{I}-\mathbf{\Lambda})\vec{r}_s^{\,c} + (\vec{1}\,\mathbf{\Lambda}\,\vec{r}_s^{\,c})\vec{e}_s$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_{c,1}, \lambda_{c,2}, \dots, \lambda_{c,n})$ is the diagonal matrix of node-dependent restart probabilities, **I** is the identity matrix and $\vec{1}$ is a 1×n vector with all entries 1.

To compute the context-sensitive node-dependent restart probability $\lambda_{c,i}$, we first measure the context relevance between a feature *f* and context *c*. In this paper, the context of a predicate *p* is its variable instantiation (*X=x, Y=y*), such as (*X*='*Microsoft*', *Y*='*Nokia*') for '*X acquire Y*'. Then we measure the context relevance using the word similarity between feature *f* and the corresponding argument of context *c*:

$$\mathrm{CR}(f,c) = \mathrm{Sim}(f_w, c_{fs})$$

where f_w is the word content of feature *f* (e.g., *people* for *X*='*people*'), *fs* is the slot signature of feature *f* (e.g., *X* for *X*='*people*'), and c_{fs} is the word in the slot *fs* of context *c*. In this paper, the similarity between two words is the cosine similarity between their word vectors (Pennington et al., 2014), using a publicly available pre-trained word vectors[1].

Finally, the context-sensitive node-dependent restart probability of node *i* is computed as:

$$\lambda_{i,c} = \begin{cases} \lambda + \beta\,(1-\lambda)\big(1.0 - \mathrm{CR}(i,c)\big) & \text{if i is a feature} \\ \lambda & \text{if i is a predicate} \end{cases}$$

where λ is the global restart probability used for smoothing, β is used to control the impact of context relevance in context-sensitive random walk, which will be empirically tuned.

Table 2 shows the learned context-specific representations of '*X acquire Y*' in different contexts. We can see that our algorithm can effectively learn context-specific representations: the most important features are *X='Google'* and *Y='Youtube'* in context (*X*='*Microsoft, Y*='*Nokia*'), by contrast the most important features are *X='children'* and *Y='skill'* in context (*X*='*people*', *Y*='*language*').

3.3 Context-Sensitive Inference Rule Discovery

Based on the above algorithm, each predicate in a specific context is represented as the context-specific steady-state probability vector $\vec{r}_s^{\,c}$. To discover inference rules, we first compute similarities between predicates, then two predicates *p* and *q* in context *c* will form an inference rule if their similarity is above a threshold. Specifically, because each representation $\vec{r}_s^{\,c}$ can be viewed as a distribution over nodes, we measure the similarity between two predicates using the Kullback–Leibler divergence between $\vec{r}_p^{\,c}$ and $\vec{r}_q^{\,c}$ (Kullback & Leibler, 1951):

[1] http://www-nlp.stanford.edu/data/glove.840B.300d.txt.gz

$$\mathrm{KL}(\vec{r}_p^c | \vec{r}_q^c) = \sum_{\mathrm{i}} r_{p,i}^c \times \ln(\frac{r_{p,i}^c}{r_{q,i}^c})$$

Notice that KL divergence is a distance measure: the smaller the KL divergence between $\vec{r}_p^c$ and $\vec{r}_q^c$, the more similar the two predicates p and q.

4 Experiments

In this section, we evaluate the performance of our method and compare it with traditional methods.

4.1 Experimental Settings

Corpus. In this paper, we use the ReVerb corpus (Fader et al., 2011) as the inference rule discovery corpus, which contains about 15 million publicly available unique open extractions. Each extraction in ReVerb is an instantiation of a predicate in the form (x, predicate, y), such as (*Facebook, acquire, Instagram*) and (*Paris, is capital of, France*). Before inference rule discovery, we apply some clean-up preprocessing to the ReVerb extractions: we remove all predicates occurring in less than 50 times and all arguments occurring in less than 10 times.

Evaluation. For evaluation, we use the publicly available dataset constructed by Zeichner et al. (2015)[2]. The dataset contains 6567 instantiated inference rules, where each one is manually labeled as correct or incorrect. For example, '*X be crucial to Y* → *X be important in Y*' is labeled as correct with instantiation (*oil prices, decisions*), and '*X own Y* → *X purchase Y*' is labeled as incorrect with instantiation (*we, these items*). For evaluation, we remove all inference rules whose predicates are not within the *ReVerb* corpus. Finally the evaluation dataset contains 5688 inference rules (2213 are correct and 3475 are incorrect). We split the dataset randomly in 2 subsets: 80% for testing and 20% for validating.

To assess the performance of different methods, we compute similarity scores for all annotated testing inference rules using different methods, and outputted the ranked inference rules of different methods using their similarity scores.

As the same as Melamud et al. (2013), we compare different methods by measuring Mean Average Precision (MAP) (Manning et al., 2008) of the inference rule ranking outputted by different methods. To compute MAP values and corresponding statistical significance, we randomly split test set into 30 subsets and computed Average Precision on every subset, the average over all subsets are used as the final MAP value.

Baselines. We compare our method with three types of inference rule discovery methods:

1) We evaluate two distributional similarity based context-insensitive baselines. One follows the DIRT similarity in (Lin and Pantel, 2001), we denote it as *DIRT*. The other uses the *Balanced-Inclusion* similarity in (Szpektor and Dagan, 2008), we denote it as *BINC*.

2) We evaluate a latent topic model based context-sensitive method. We follow the method described in Melamud et al. (2013), a two level model which computes context-sensitive similarity using two predicates' word-level vectors biased by topic-level context representations. We apply their method on two base word-level similarities, the *LIN* similarity and the *BINC* similarity, correspondingly denoted as *WT-LIN* and *WT-BINC*.

3) We evaluate the global learning method proposed in Berant et al. (2011), which use ILP solvers to performance global optimization over local classification results—We denote it as *ILP*. For comparison, we directly use the inference rule resource[3] released by Berant et al. (2011), which was also learned from the *ReVerb* corpus.

For our graph-based method, we tune its parameters on the validating dataset, and the final parameters used in our method are as follows: the global restart probability λ=0.1, the weight of the semantic dependent edge α = 4.0, and the context relevance restart weight β=0.7.

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

[3] http://www-nlp.stanford.edu/joberant/homepage_files/resources/ACL2011Resource.zip

4.2 Experimental Results and Discussions

We conduct experiments on the test dataset using all baselines. For our method, we use two different settings: one uses context-insensitive random walk – we denote it as *RWR-CI*, and the other uses context-sensitive random walk—we denote it as *RWR-CS*. The overall results are presented in Table 2.

System	MAP
DIRT	0.401
BINC	0.424
WT-LIN	0.482
WT-BINC	0.500
ILP	0.513
RWR-CI	0.511
RWR-CS	**0.576**

Table 2. The overall results of different methods

From Table 2, we can see that:

1) By taking both the context and the inter-dependencies between predicates into consideration, our method can achieve significant performance improvement over traditional methods. Compared with the distributional similarity based baselines *DIRT* and *BINC*, *RWR-CS* achieved 44% and 36% MAP improvements. Compared with the latent topic model based context-sensitive baselines *WT-LIN* and *WT-BINC*, *RWR-CS* achieved 20% and 15% MAP improvements. Compared with the global learning baseline *ILP*, *RWR-CS* achieved 12% MAP improvement.
2) Context-sensitive similarity is critical for inference rule discovery. By taking the context into consideration, *WT-LIN*, *WT-BINC* and *RWR-CS* correspondingly achieved 20%, 18% and 13% MAP improvements over their context-insensitive counterparts—*DIRT*, *BINC* and *RWR-CI*.
3) The predicate inter-dependency can enhance the performance of inference rule discovery. By taking advantage of the rich inter-dependencies, both *ILP* and *RWR-CI* achieve performance improvements over the two baselines which model predicates independently: *DIRT* and *BINC*.

To better understand the reasons why and how the graph-based method works well, we evaluate our method using different settings. The results are presented in Table 3.

	Context-Insensitive Random Walk	**Context-Sensitive Random Walk**
Co-occurrence Edges	0.506	0.547
+ Semantic-Dependent Edges	0.511	0.576

Table 3. The results of the different settings of our method

From Table 3, we can see that:

1) The context-sensitive random walk algorithm can effectively capture the semantics of a predicate in a specific context: Using context-sensitive random walk algorithm, our method achieves MAP improvements on both predicate graph settings (co-occurrence edges only and all edges).
2) The predicate inter-dependency and the context-sensitive random walk can reinforce each other: our method can achieve a 14% MAP improvement by both adding semantic-dependent edges and performing context-sensitive random walk, which is larger than the sum of the performance improvements by only adding semantic-dependent edges (1% improvement) and by only performing context-sensitive random walk (8% improvement). We believe this is because although the inter-dependencies between predicates can enrich predicate representation with more information, it may also introduce some irrelevant information. As a complement, the context-sensitive random walk can filter out irrelevant information and retain only relevant information.

5 Conclusions and Future Work

This paper proposes a graph-based method for context-sensitive inference rule discovery. The advantages of our method are: 1) our method is context-sensitive, it can accurately represent the semantics of a predicate in a specific context; 2) our method can take advantage of the inter-dependencies between predicates for better predicate representation. Experiments verified the effectiveness of our method.

In future work, we aim to jointly model inference rule discovery and knowledge base completion, so that inference rules can be exploited to complete a knowledge base and the semantic knowledge in the given knowledge base can be used to enhance inference rule discovery. Furthermore, we also want to learn the distributed representations of predicates using deep neural networks.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants no. 61572477, 61433015 and 61272324, and the National High Technology Development 863 Program of China under Grants no. 2015AA015405. Moreover, we sincerely thank the reviewers for their valuable comments.

Reference

Berant, J., Dagan, I. and Goldberger, J. 2010. *Global learning of focused entailment graphs.* In: Proceedings of ACL 2010.

Berant, J., Dagan, I. and Goldberger, J. 2011. *Global learning of typed entailment rules*. In: Proceedings of ACL 2011.

Bhagat, R., Pantel, P., Hovy, E. and Rey, M. 2007. *LEDIR: An unsupervised algorithm for learning directionality of inference rules*. In: Proceedings of EMNLP-CoNLL 2007.

Dagan, I., Glickman, O. and Magnini, B. 2006. *The pascal recognizing textual entailment challenge*. In: Lecture Notes in Computer Science, 3944:177-190.

Dinu, G. and Lapata, M. 2010. *Measuring distributional similarity in context*. In: Proceedings of EMNLP 2010.

Dutta, A., Meilicke, C. and Stuckenschmidt, H. 2015. *Enriching Structured Knowledge with Open Information*. In: Proceedings of WWW 2015.

Fader, A., Soderland, S. and Etzioni, O. 2011. *Identifying relations for open information extraction*. In: Proceedings of EMNLP 2011.

Hashimoto, C. and Torisawa, K. and Kuroda, K. and De Saeger, S. and Murata, M. and Kazama, J. 2009. *Large-Scale Verb Entailment Acquisition from the Web*. In: Proceedings of EMNLP 2009.

Haveliwala, T. H. 2003. *Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search*. In: IEEE Transactions on Knowledge and Data Engineering.

He, J., Li, M., Zhang, H., Tong, H. and Zhang, C. 2004. *Manifoldranking based image retrieval*. In: Proceedings of ACM Multimedia 2004.

Hearst, M. A. 1992. *Automatic acquisition of hyponyms from large text corpora*. In: Proceedings of COLING 1992.

Kullback, S. and Leibler, R.A. 1951. On information and sufficiency. In: Annals of Mathematical Statistics 22 (1): 79–86.

Lin, D. and Pantel, P. 2001. *DIRT—discovery of inference rules from text*. In: Proceedings of ACM SIGKDD 2001.

Kok, S. and Domingos, P. 2008. *Extracting semantic networks from text via relational clustering*. In: Machine Learning and Knowledge Discovery in Databases, pp. 624-639. Springer Berlin Heidelberg, 2008.

Manning, C., Raghavan, P. and Sch¨utze, H. 2008. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge.

Melamud, O., Berant, J., Dagan, I., Goldberger, J. and Szpektor, I. 2013. *A Two Level Model for Context Sensitive Inference Rules*. In: Proceedings of ACL 2013.

Nakashole, N., Weikum, G. and Suchanek, F. 2012. *Patty: A taxonomy of relational patterns with semantic types*. In: Proceedings of EMNLP 2012.

Pantel, P., Bhagat, R., Coppola, B., Chklovski, T. and Hovy, E. 2007. *ISP: Learning inferential selectional preferences*. In: Proceedings of NAACL-HLT 2007.

Pennington, J., Socher, R. and Manning C. 2014. *Glove: Global Vectors for Word Representation*. In: Proceedings of EMNLP 2014.

Ritter, A. and Etzioni, O. 2010. *A latent dirichlet allocation method for selectional preferences*. In: Proceedings of ACL 2010.

Ravichandran, D. and Hovy, E. 2002. *Learning surface text patterns for a question answering system*. In: Proceedings of ACL 2002.

Sekine, S. 2005. *Automatic paraphrase discovery based on context and keywords between NE pairs*. In: Proceedings of IWP 2005.

Szpektor, I., and Dagan, I. 2008. *Learning entailment rules for unary templates*. In: Proceedings of COLING 2008.

Szpektor, I., Dagan, I., Bar-Haim, R. and Goldberger, J. 2008. *Contextual preferences*. In: Proceedings of ACL-HLT 2008.

Szpektor, I., Tanev, H., Dagan, I. and Coppola, B. 2004. *Scaling Web-based Acquisition of Entailment Relations*. In: Proceedings of EMNLP 2004.

Szpektor, I., Shnarch, E. and Dagan, I. 2007. *Instance-based evaluation of entailment rule acquisition*. In: Proceedings of ACL 2007.

Tong, H., Faloutsos, C. and Pan, J.Y. 2006. *Fast random walk with restart and its applications*. In: Proceedings of ICDM 2006.

Yates, A. and Etzioni, O. 2009. *Unsupervised methods for determining object and relation synonyms on the web*. In: Journal of Artificial Intelligence Research 34, no. 1 (2009): 255.

Zeichner, N., Berant, J. and Dagan, I. 2012. *Crowdsourcing inference-rule evaluation*. In Proceedings of ACL 2012.