Learning Interpretable Relationships between Entities, Relations and Concepts via Bayesian Structure Learning on Open Domain Facts

Jingyuan Zhang, Mingming Sun, Yue Feng, Ping Li

Cognitive Computing Lab

Baidu Research

1195 Bordeaux Dr, Sunnyvale, CA 94089, USA

No. 10 Xibeiwang East Road, Beijing 100085, China

10900 NE 8th St, Bellevue, WA 98004, USA

{zhangjingyuan03,sunmingming01, v_fengyue, liping11}@baidu.com

Abstract

Concept graphs are created as universal taxonomies for text understanding in the open domain knowledge. The nodes in concept graphs include both entities and concepts. The edges are from entities to concepts, showing that an entity is an instance of a concept. In this paper, we propose the task of learning interpretable relationships from open domain facts to enrich and refine concept graphs. The Bayesian network structures are learned from open domain facts as the interpretable relationships between relations of facts and concepts of entities. We conduct extensive experiments on public English and Chinese datasets. Compared to the state-of-the-art methods, the learned network structures help improving the identification of concepts for entities based on the relations of entities on both English and Chinese datasets.

1 Introduction

Concept graphs are created as universal taxonomies for text understanding and reasoning in the open domain knowledge (Dagan et al., 2010; Bowman et al., 2015; Zamir et al., 2018; Huang et al., 2019; Hao et al., 2019; Jiang et al., 2019). The nodes in concept graphs include both entities and concepts. The edges are from entities to concepts, showing that an entity is an instance of a concept. The task of extracting and building concept graphs from user-generated texts has attracted a lot of research attentions for a couple of decades (Fellbaum, 1998; Wu et al., 2012; Shwartz et al., 2016; Chang et al., 2018; Le et al., 2019; Lewis, 2019). Most of these methods rely on high quality syntactic patterns to determine whether an entity belongs to a concept. For example, given the pattern "X is a Y" or "Y, including X" appearing in sentences, we can infer that the entity X is an instance of the concept Y. These pattern-based methods require that an entity and concept pair co-occurs in sentences. However, due to the different expressions of a certain concept, an entity and a concept may rarely appear in sentences together. We conduct a data analysis of millions of sentences extracted from Wikipedia and discover that only 10.61% of entity-concept pairs co-occur in sentences out of more than six million of pairs from the public Microsoft concept graph (https://concept.research.microsoft.com). We also analyze Baidu Baike (http://baike.baidu.com) and its corresponding concept graph. A similar phenomenon is observed that only 8.56% entity-concept pairs co-occur in sentences. Table 1 shows the statistics for Wikipedia and Baidu Baike. With such limitations, the existing approaches have difficulties in helping build a complete concept graph from open domain texts.

Dataset	# Pairs	# Sentences	# Co-occurrence	Percentage
Wikipedia	6,347,294	7,871,825	673,542	10.61%
Baike	3,229,301	9,523,183	276,485	8.56%

Table 1: Entity-concept pairs that co-occur in sentences from Wikipedia (English) and Baidu Baike (Chinese).

Nowadays, the task of open domain information extraction (OIE) has become more and more important (Christensen et al., 2011; Wu and Weld, 2010; Etzioni et al., 2011; Mausam et al., 2012; Sun et al., 2018b,a; Di et al., 2019; Rashed et al., 2019; Liu et al., 2020a,b). OIE aims to generate entity and relation level intermediate structures to express facts from open domain sentences. These open domain facts usually express natural languages as triples in the form of (subject, predicate, object). For example, given the sentence "Anderson, who hosted Whose Line, is a winner of a British Comedy Award in 1991.", two facts will be extracted. They are ("Anderson", "host", "Whose Line") and ("Anderson", "winner of a British Comedy Award", "1991"). The subject and object in a fact are both
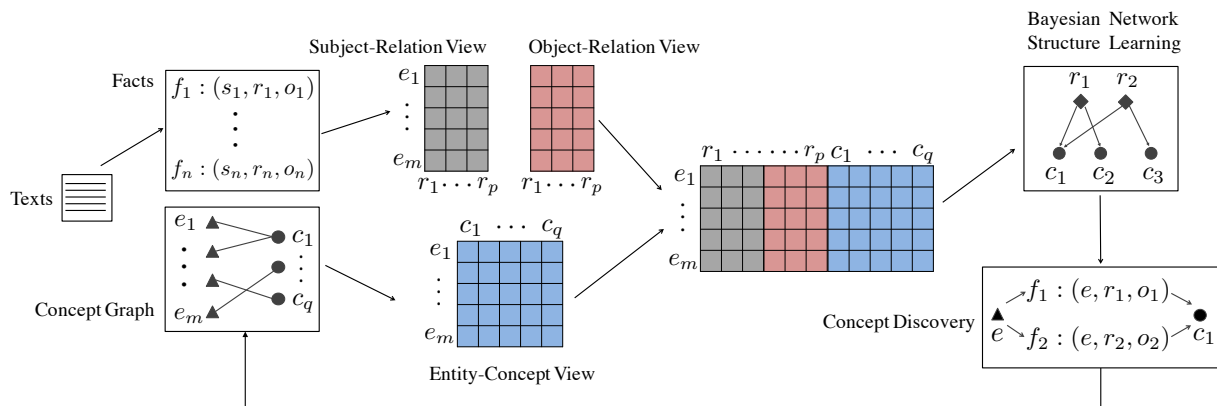
8045

Figure 1: The workflow of learning interpretable relationships from open domain facts for concept discovery. $f_i = (s_i, r_i, o_i)$ represents a fact, where s_i and o_i are both entities, and r_i is a relation. We use e_i to denote an entity and c_i to represent a concept.

entities. The open domain facts contain rich information about entities by representing the subject or object entities via different types of relations (i.e., groups of predicates).

It would be helpful for concept graph completion if we can take advantage of the relations in open domain facts. We again take the above two facts of "Anderson" as an instance. If we have explored the connections between relations of facts and concepts, and learned that "host" and "winner of a British Comedy Award" are associated with an "English presenter" subject with a higher probability than a "Japanese presenter" subject, we can infer that "Anderson" belongs to the "English presenter" concept regardless of whether these two co-appear in a sentence or not. In real-world open domain corpus, however, the connections between relations and concepts are not available to us.

In this paper, we propose the task of learning interpretable relationships between entities, relations and concepts from open domain facts to help enriching and refining concept graphs. Learning Bayesian networks (BNs) from data has been studied extensively (Heckerman et al., 1995; Koivisto and Sood, 2004; Scanagatta et al., 2015; Niinimaki et al., 2016) in the last few decades. The BNs formally encode probabilistic connections in a certain domain, yielding a human-oriented qualitative structure that facilitates communication between a user and a system incorporating the probabilistic model. Specifically, we apply the Bayesian network structure learning (BNSL) (Chow and Liu, 1968; Yuan et al., 2011; Yuan and Malone, 2013) to discover meaningful relationships between entities, relations and concepts from open domain facts. The learned network encodes the dependen-

cies from the relations of entities in facts to the concepts of entities, leading to the identification of more entity-concept pairs from open domain facts for the completion of concept graphs. Figure 1 illustrates the proposed workflow of learning interpretable relationships from open domain facts.

We summarize our contributions as follows:

- We propose the task of learning interpretable relationships between entities, relations and concepts from open domain facts, which is important for enriching and refining concept graphs.

- We build the BNSL model to discover meaningful network structures that express the connections from relations of entities in open domain facts to concepts of entities in concept graphs.

- Experimental results on both English and Chinese datasets reveal that the learned interpretable relationships help identify concepts for entities based on the relations of entities, resulting in a more complete concept graph.

2 Related Work

Concept Graph Construction. Concept graph construction has been extensively studied in the literature (Fellbaum, 1998; Ponzetto and Strube, 2007; Banko et al., 2007; Suchanek et al., 2007; Wu et al., 2012; Shwartz et al., 2016; Chang et al., 2018; Le et al., 2019; Lewis, 2019). Notable works toward creating open domain concept graphs from scratch include YAGO (Suchanek et al., 2007) and Probase (Wu et al., 2012). In addition, a wide variety of methods (Nakashole et al., 2012; Weeds et al., 2014; Roller et al., 2014; Shwartz et al., 2016; Roller et al., 2018; Chang et al., 2018; Le et al., 2019; Lewis, 2019) are developed to detect the

hypernymy between entities and concepts for a more complete concept graph. Distributional representations of entities and concepts are learned for good hypernymy detection results (Weeds et al., 2014; Roller et al., 2014; Chang et al., 2018; Lewis, 2019). In contrast to distributional methods, path-based algorithms (Nakashole et al., 2012; Shwartz et al., 2016; Roller et al., 2018; Le et al., 2019) are proposed to take advantage of the lexico-syntactic paths connecting the joint occurrences of an entity and a concept in a corpus. Most of these methods require the co-occurrence of entity and concept pairs in sentences for the graph completion task. However, due to the different expressions of a certain concept, an entity and a concept may rarely appear in one sentence together. With such limitations, the existing methods in the literature cannot deal with those non co-occurring entity concept pairs, leading to an incomplete concept graph.

Open Domain Information Extraction. Open domain information extraction (OIE) has attracted a lot of attention in recent years (Wu and Weld, 2010; Christensen et al., 2011; Etzioni et al., 2011; Mausam et al., 2012; Pal and Mausam, 2016; Yahya et al., 2014; Sun et al., 2018b,a; Roy et al., 2019; Liu et al., 2020a,b). It extracts facts from open domain documents and expresses facts as triples of (subject, predicate, object). Recently, a neural-based OIE system Logician (Sun et al., 2018b,a; Liu et al., 2020a,b) is proposed. It introduces a unified knowledge expression format SAOKE (symbol aided open knowledge expression) and expresses the most majority information in natural language sentences into four types of facts (i.e., relation, attribute, description and concept). Logician is trained on a human labeled SAOKE dataset using a neural sequence to sequence model. It achieves a much better performance than traditional OIE systems in Chinese language and provides a set of open domain facts with much higher quality to support upper-level algorithms. Since the subject and object in a fact are both entities, the open domain facts contain rich information about entities by representing the subjects or objects via different types of relations (i.e., groups of predicates). It can help the task of concept graph completion by making full use of the relations in open domain facts. In this paper, we leverage the high-quality facts of Logician as one dataset in the experiment.

Bayesian Network Structure Learning. Learning a Bayesian network structure from real-world data is a well-motivated but computationally hard task (Heckerman et al., 1995; Koivisto and Sood, 2004; de Campos et al., 2009; Malone et al., 2011; Scanagatta et al., 2015; Niinimaki et al., 2016). A Bayesian network specifies a joint probability distribution of a set of random variables in a structured fashion. A key component in this model is the network structure, a directed acyclic graph on the variables, encoding a set of conditional independence assertions. Several exact and approximate algorithms are developed to learn optimal Bayesian networks (Chow and Liu, 1968; Koivisto and Sood, 2004; Singh and Moore, 2005; Silander and Myllymäki, 2006; Yuan et al., 2011; Yuan and Malone, 2013). Some exact algorithms (Koivisto and Sood, 2004; Singh and Moore, 2005; Silander and Myllymäki, 2006) are based on dynamic programming to find the best Bayesian network. In 2011, an A* search algorithm is introduced (Yuan et al., 2011) to formulate the learning process as a shortest path finding problem. However, these exact algorithms are inefficient due to the full evaluation of an exponential solution space. In this paper, we consider the Chow-Liu tree building algorithm (Chow and Liu, 1968) to approximate the underlying relationships between entities, relations and concepts as a dependency tree. This method is very efficient when there are large numbers of variables.

3 Finding Interpretable Relationships

We formulate the relationships between entities, relations, and concepts as follows:

- Entities are associated with a set of relations that represent the behaviors and attributes of entities;

- A concept is defined by a set of relations. The instances of a concept are those entities that associate with the corresponding set of relations.

In concept graphs, a concept is associated with a set of entities which share some common behaviors or attributes. However, the essence of a concept is a set of relations, and entities which associate with these relations automatically become the instance of the concept. So our formulation of the relationships between entities, relations and concepts can be illustrated by Figure 2.

In the closed domain, a knowledge base has a predefined ontology and the relationships in Figure 2 are already known. For example, DBPedia (Auer et al., 2007) builds a knowledge graph
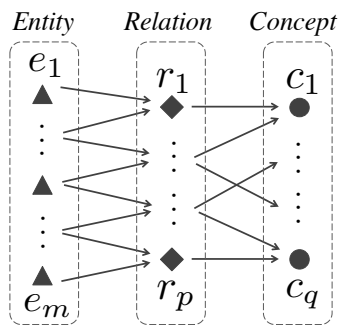
Figure 2: Relationships of entities, relations and concepts.

from Wikipedia to encode the relationships between entities and relations in the forms of facts. The relationships between relations and concepts are represented in the ontology structure of DBPedia, where each concept is associated with a group of relations.

However, in the open domain, a predefined ontology does not exist, and hence the components in Figure 2 may not be associated with each other. For instance, given an open domain concept graph, we can discover the relationships between entities and concepts. Given the open domain corpus/facts, we can find the relationships between entities and relations. But the relationships between open domain concepts and relations are not available, to our knowledge. In this paper, we aim to find the connection between open domain relations and concepts, so that we can provide interpretations to the question "why the entity is associated with those concepts in open domain".

3.1 Problem Formulation

Suppose we have a set of entities $E = \{e_1, \cdots, e_m\}$, a set of relations $R = \{r_1, \cdots, r_p\}$, a set of concepts $C = \{c_1, \cdots, c_q\}$, and a set of observed triplets $O = \{(e, r, c)\}$. Here E and C are from a concept graph G. R is from a set of facts $F = \{f_1, \cdots, f_n\}$ extracted from a text corpus D. A triplet (e, r, c) is observed means that the entity e with relation r and concept of c is found in above data sources. Given a set of observations O with **N** samples, the Bayesian network can be learned by maximizing the joint probability $p(O)$:

$$
\begin{aligned}
p(O) &= \prod_{(e,r,c)\in O} p((e,r,c)) \\
&= \prod_{(e,r,c)\in O} p(c|(e,r)) \cdot p(r|e) \cdot p(e) \\
&= \prod_{(e,r,c)\in O} p(c|r) \cdot p(r|e) \cdot p(e)
\end{aligned}
$$

where $p(c|(e,r)) = p(c|r)$ is due to our Bayesian network assumption (see Figure 2). By learning with the observed triplets with above model, we can infer the missing triplets, especially give interpretable relationship between entities and concepts.

Since $p(r|e)$ can be approximated by the information from OIE corpus, the core of the above problem becomes to learn the part of the network of $p(c|e)$. The difficulty of learning $p(c|e)$ is the unknown structure of the Bayesian network. Due to sparsity of real-world knowledge base, the target network would be sparse. But the sparse structure must be known beforehand for probability learning.

In this paper, we employ the Bayesian Network Structure Learning (BNSL) technique to explore the connections between relations and concepts. Due to the large number of variables (i.e., entities, relations and concepts) in open domain facts and concept graphs, we develop an approximate algorithm to learn the network structure.

3.2 The Proposed Approximate Algorithm

Due to the sparsity of the relationships between relations and concepts, we decompose the problem into several sub-problems, with each sub-problem containing only one concept variable. Then for each concept variable, we identify possible related relations and apply a BNSL algorithm to discover the network structure between them. Finally, we use the learned network for concept discovery. The procedure is shown in Algorithm 1. We will state the key steps in detail in the next sub-sections.

3.2.1 Sub-problem Construction

Given a concept $c \in C$, we first collect all its entities $E_c \subset E$ from the concept graph. Then we can obtain a set of facts F_c that contain these entities. Since an entity can appear in a fact as a subject or an object, we split the facts F_c into subject-view facts $F_{c,s}$ and object-view facts $F_{c,o}$. If we make use of all the relations under the subject or object view, it would be inefficient or event impossible to learn the sparse network structure with a large number of relation variables. Hence, based on the facts, we select possible related relations to the concept c to reduce the complexity of the problem.

3.2.2 Relation Selection

There are various strategies which can be applied for the relation selection. We can assume that a relation is highly related to the concept if it appears many times in the fact set F_c. In this way, we can

Algorithm 1: BNSL for concept discovery

Input: Texts D and a concept graph G.
Output: Valid entity-concept pairs.
```
/* OIE step:                    */
```
1   Extract open domain facts F from D;
```
/* Concept discovery step:      */
```
2   **for** *each concept $c \in C$* **do**
3   Get entities E_c of this concept;
4   Select facts F_c including E_c;
```
        /* Subject view step:    */
```
5   Split F_c into subject-view facts $F_{c,s}$;
6   Select top K relations $R_{c,s}$ from $F_{c,s}$;
7   Get entity-relation data $X_{c,s}$;
```
        /* Object view step:     */
```
8   Repeat step 5 to get object-view $F_{c,o}$;
9   Repeat step 6 to get $R_{c,o}$ from $F_{c,o}$;
10   Repeat step 7 to get $X_{c,o}$;
```
        /* BNSL training step:   */
```
11   Feed $X_{c,s}$ and $X_{c,o}$ into BNSL;
12   Get a network structure S_c for c;
13   **end for**
```
/* BNSL prediction step:        */
```
14   Predict on new entities;
15   Return valid entity-concept pairs;

---

count the frequencies of relations for each view and select the top K as the most relevant ones with a concept. We call it TF selection since we measure the relevance of a relation according to its frequency. We can also select relations according to the TFIDF measurement (Wu et al., 2008). For each view, we select the most relevant K relations for the concept c. We denote them as $R_{c,s} \subset R$ for the subject-view facts and $R_{c,o} \subset R$ for the object-view facts. In summary, for each concept, we construct two sub-problems for the BNSL task. One is from the subject view and the other is from the object view. Under each view, the sub-problem contains one concept and at most K relations. The goal is to learn a network structure from the concept and corresponding relations.

3.2.3   Data Observations

Given a sub-problem for a concept c, we first obtain the corresponding data observations and then feed them as the input of BNSL for interpretable relationship discoveries. For each concept, we can learn a Bayesian network structure from its top subject-view or object view relations. The data observations $X_{c,s}$ with TF relation selection for the subject-view of the concept c are generated as

follows: for each entity $e \in E_c$, we use 1 to be the concept observation, meaning that the entity e is an instance of concept c. We use the times of the subject e and a top relation $r \in R_{c,s}$ appearing together in facts $F_{c,s}$ as a relation observation for e and r. The K relation observations and the concept observation together become the positive data observations for c. In order to learn meaningful network structures, we generate an equal number of negative data observations for c. We first randomly sample the same number of entities from $E_{c'} = \{e_i : e_i \in E \setminus E_c\}$ as negative entities of c. We use 0 as the concept observation for negative entities. Then for each negative entity e', we count the times of the subject e' and a relation $r \in R_{c,s}$ appearing in all the collected facts as a relation observation for e' and r. The K relation observations and the concept observation together become the negative data observations for c. $X_{c,s}$ consists of both the positive and negative data observations. Similarly, we can obtain the data observations $X_{c,o}$ for the object view.

3.2.4   Network Structure Learning

In this paper, we employ the widely-used Chow-Liu tree building algorithm (Chow and Liu, 1968) as the BNSL method. This algorithm approximates the underlying distributions of variables as a dependency tree, which is a graph where each node only has one parent and cycles are not allowed. It will first calculate the mutual information between each pair of nodes (i.e., variables), and then take the maximum spanning tree of that matrix as the approximation. While this will only provide a rough approximation of the underlying data, it provides good results for many applications (Suzuki, 2010; Tavassolipour et al., 2014; Hassan-Moghaddam and Jovanovic, 2018; Ding et al., 2019), especially when you need to know the most important influencer on each variable. In addition, this algorithm becomes extremely efficient when it deals with to a large number of variables.

Since both the subject and object views reflect some properties of entities, we can concatenate the subject-view relations and object-view relations together for a more complete representation of entities. The concatenated data can be forwarded into BNSL for a more comprehensive result of interpretable relationship discovery. Given q concept variables and K relevant relations for each concept, the number of parameters in BNSL is at most $q \times K$.

3.2.5 Prediction

After we learn a network structure for each concept, we can learn the concept of a new entity e easily. We first identify the open domain facts with e as its subject or object, and then feed the observation of relations for a concept c into the network to calculate the probability of $p(c|e)$. We still use the open domain entity "Anderson" and its two facts introduced in Section 1 as an example to show how BNSL works. Assume we have two open domain concepts, "English presenter" and "Japanese presenter". Given the entity "Anderson" and its open domain relations "host" and "winner of a British Comedy Award" as input of BNSL, the output is the probabilities that "Anderson" belongs to each concept. BNSL will predict a higher probability for "Anderson" having the concept "English presenter" than having "Japanese presenter".

4 Experiments

With the learned relationship between relations and concepts from BNSL, we indirectly associate entities with their concepts and give interpretations to the question "why the entity is associated with those concepts in open domain". The hypernymy detection task aims to identify concepts for entities in open domain. It is helpful for us to evaluate the quality of the learned relationships from BNSL. In this section, we conduct extensive experiments to evaluate the performance of BNSL.

4.1 Data Description

We test the performance of our proposed method on two public datasets, one is in English and the other is in Chinese. For the English dataset, we use 15 million high-precision OIE facts[1], the Microsoft concept graph[2] and 7.87 million Wikipedia sentences[3] for our experiments. Since there are more than 5 million concepts in the English dataset and most of them have few entities, we focus on those concepts with more than 50 entities in the experiments. For the Chinese dataset, we use sentences and the corresponding facts[4] in (Sun et al., 2018b). The concept graph is also built by Baidu Baike. Table 2 shows the statistics of the concept

---

[1] http://reverb.cs.washington.edu
[2] https://concept.research.microsoft.com/Home/Download
[3] https://www.kaggle.com/mikeortman/wikipedia-sentences
[4] https://ai.baidu.com/broad/download?dataset=saoke

Concept Graphs	Dataset	# entities	# concepts	# overlaps	% overlaps
	English	12,501,527	5,376,526	613,454	27.10%
	Chinese	9,230,727	3,245	475,507	48.14%
Facts	Dataset	# facts	# subjects	# objects	# predicates
	English	14,728,268	1,396,793	1,698,028	664,746
	Chinese	37,309,458	624,632	550,404	10,145

Table 2: Statistics of concept graphs and facts.

graphs and open domain facts.

In open domain facts, each mention of a subject or object is considered as an open domain entity. So we naturally map an entity in open domain facts and concept graphs by the same mention. In Table 2, the column "# of overlap" is about the number of fact entities appearing in the concept graph and the last column is the percentage of fact entities in the concept graph. With the predicates as relations for the open domain facts, we build the Bayesian network structure learning method to bridge the gap between relations in open domain facts and concepts in the concept graph.

4.2 Experimental Setting

In the experiment, we compare with the state-of-the-art model HypeNet (Shwartz et al., 2016) for hypernymy detection. HypeNet improves the detection of entity-concept pairs with an integrated path-based and distributional method. An entity and a concept must appear together in a sentence so that HypeNet can extract lexico-syntactic dependency paths for training and prediction. However, only less than 11% of entity-concept pairs co-occur in Wikipedia sentences in reality (Table 1). Therefore, we compare BNSL with HypeNet only on the entity-concept pairs that co-appear in sentences.

In addition, we compare BNSL with recurrent neural networks (RNNs). We apply attention-based Bi-LSTM (Zhou et al., 2016) and derive three versions of RNNs as baseline methods, RNN(f), RNN(sen) and RNN(e). RNN(f) determines the concepts of an entity according to the facts containing the entity, while RNN(sen) by the sentences containing the co-appearance of an entity and a concept. Specifically, each entity in RNN(f) is represented by its associated facts. Each fact is a sequence of subject, predict and object. Each subject, predict and object vector is fed in sequence into RNN(f), resulting a fact embedding vector. The averaged fact vector becomes the entitys feature for concept classification.

Similar to HypeNet, RNN(sen) requires the entity-concept pairs co-appearing in sentences. Dif-

ferent from RNN(sen), RNN(e) focuses on sentences containing the entity only. Based on the sentences, RNN(e) aims to learn which concept an entity belongs to. We follow HypeNet and RNN to use pre-trained GloVe embeddings (Pennington et al., 2014) for initialization. Besides, we compare BNSL with traditional support vector machines (SVM) with linear kernel. The input features for SVM and BNSL are the same, i.e., the top K relations for each concept. Here we set $K = 5$.

During testing, all methods are evaluated on the same testing entities. we calculate the accuracy, precision, recall and F1-score over the prediction results for evaluation. We split the data into 80% of training and 20% of testing. For English, the total numbers of training and testing data are 504,731 and 123,880, respectively; whereas for Chinese, the numbers are 5,169,220 and 1,289,382, respectively.

4.3 Performance Evaluation

In this section, we show the evaluation performance on the task of concept discovery with the learned interpretable relationships from open domain fact. Table 3 and Table 4 list the results for co-occurred and non co-occurred entity-concept pairs in sentences respectively. In the tables, (s) and (o) mean the performance only under the subject and the object view, respectively. RNN(f), BNSL and SVM present the prediction performance with the concatenation of both the subject and object views. As is mentioned in the previous section, we can use TF or TFIDF for the most relevant relation selection. We test both strategies for BNSL and SVM. For the English dataset, TFIDF performs much better than TF while the result is the opposite for the Chinese dataset. In this section, we analyze the results of BNSL and SVM with TFIDF for the English dataset. For the Chinese dataset, we report the performance of BNSL and SVM with TF. We will show more results for the relation selection in the next section.

For the co-occurred entity-concept pairs in sentences, BNSL(s) performs the best for both datasets. Surprisingly, SVM performs much better than HypeNet with an improvement of around 10% on accuracy for both datasets as is shown in Table 3. In addition, SVM achieves better results compared to RNN(sen). The reason that HypeNet or RNN(sen) cannot perform well may be that the information expressed from the sentences are too diverse. HypeNet or RNN(sen) cannot capture meaningful pat-

terns from sentences for the task of concept discovery. Since RNN(e) further ignores the concept information during the sentence collection step, it cannot perform well compared with RNN(sen). In contrast, information extracted from open domain facts are much more concentrated about concepts. Furthermore, the most relevant relations associated with entities help filtering out noise. Therefore, SVM can achieve a much better result than sentence-based baselines.

Though SVM does well on the co-occurred data, BNSL outperforms SVM with all the four evaluation metrics. By learning interpretable relationships between relations and concepts, BNSL captures the most important knowledge about concepts and further exploits their dependencies to help improve the concept discovery task. However, the concatenation of subject and object views for BNSL cannot help improve the performance for both datasets. Similar phenomena can be observed for RNN(f) and SVM. Specifically, the results under the subject view are usually better than those of the object view, implying that when people narrate facts, they may pay more attention to selecting suitable predicate for subjects, rather for objects. Table 4 lists the performances of RNN(e), RNN(f), SVM and BNSL on non co-occurred data. We can observe a similar trend compared to the results on co-occurred data.

Since HypeNet and BNSL make use of different information sources (natural language sentences for HypeNet and open domain facts for BNSL), we try to ensemble them to improve the performance further. We first train HypeNet and BNSL independently. Then we can obtain prediction probabilities of entity-concept pairs from HypeNet and BNSL separately. We select the probabilities with higher values as the final predictions. The last row in Table 3 shows the performance of ensembling HypeNet and BNSL. We denote it as B + H. It can be seen that B + H achieves the best accuracy, recall and F1-scores on the co-occurred data. It reveals that interpretable relationships extracted from open domain facts are complementary to natural language sentences in helping concept discovery. Studying meaningful knowledge from open domain facts provides an alternative perspective to build concept graphs and this paper starts the first trial.

4.4 Analysis on the Relation Selection

Relation selection helps reducing the complexity of BNSL. In this section, we first evaluate how differ-

Dataset	English				Chinese			
Method	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
HypeNet	69.64%	75.09%	69.74%	72.31%	76.57%	87.17%	71.22%	78.39%
RNN(sen)	77.18%	80.74%	78.62%	79.67%	71.90%	72.85%	84.35%	78.18%
RNN(e)	67.77%	77.09%	61.62%	68.49%	57.67%	61.19%	79.53%	69.16%
RNN(s)	73.38%	80.35%	70.39%	75.04%	64.93%	64.02%	94.13%	76.21%
RNN(o)	70.95%	79.81%	65.46%	71.93%	64.97%	64.08%	94.01%	76.21%
RNN(f)	70.01%	79.08%	64.25%	70.90%	49.55%	61.23%	42.81%	49.95%
SVM(s)	76.68%	74.82%	88.93%	81.26%	85.06%	90.01%	84.33%	87.07%
SVM(o)	74.81%	72.72%	89.14%	80.10%	51.86%	57.54%	73.87%	64.69%
SVM	77.43%	74.38%	92.00%	82.25%	86.07%	90. 86%	85.22%	87.95%
BNSL(s)	86.03%	82.89%	**95.07%**	88.56%	87.54%	**92.40%**	86.21%	89.20%
BNSL(o)	86.22%	84.52%	92.76%	88.45%	49.03%	56.79%	61.10%	58.86%
BNSL	84.79%	81.87%	94.08%	87.55%	87.37%	92.32%	86.00%	89.05%
B + H	**91.27%**	**91.15%**	93.75%	**92.43%**	**87.88%**	86.01%	**95.18%**	**90.36%**

Table 3: Performance on the co-occurred data. The best results are **in bold**.

Dataset	English				Chinese			
Method	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
RNN(e)	63.94%	67.38%	52.09%	58.75%	53.82%	51.84%	95.06%	67.09%
RNN(s)	73.83%	74.61%	71.12%	72.82%	55.18%	52.55%	**97.49%**	68.29%
RNN(o)	73.74%	77.05%	66.56%	71.42%	55.34%	52.64%	97.47%	68.36%
RNN(f)	72.36%	75.53%	65.02%	69.88%	51.82%	51.63%	42.45%	46.59%
SVM(s)	71.94%	66.48%	86.91%	75.34%	90.03%	86.73%	94.30%	90.36%
SVM(o)	65.82%	61.55%	81.70%	70.21%	51.14%	50.39%	85.37%	63.37%
SVM	71.62%	65.62%	89.16%	75.60%	90.91%	88.11%	94.37%	91.14%
BNSL(s)	**85.97%**	**82.15%**	**91.42%**	**86.54%**	**92.47%**	**90.12%**	95.23%	**92.60%**
BNSL(o)	82.27%	78.36%	88.48%	83.11%	51.52%	50.70%	74.63%	60.38%
BNSL	84.78%	80.77%	90.74%	85.47%	92.39%	90.05%	95.15%	92.53%

Table 4: Performance on the non co-occurred data. The best results are **in bold**.

ent relation selection strategies will influence the performance of BNSL and SVM methods. Table 5 is the performance of TF and TFIDF relation selection on the entire data for both English and Chinese. We observe that TFIDF selection performs better on English while TF is better on Chinese. However, BNSL always outperforms SVM regardless of the views or the relation selections. In addition, since SVM performs much better than the neural network based HypeNet and RNN, we try to ensemble it with BNSL to improve the performance further. We consider the prediction probabilities of SVM as a new variable and incorporate it into BNSL for network structure learning. We denote the model as BNSL + SVM. For comparison, we ensemble SVM with BNSL by taking the results of BNSL as one new feature dimension to SVM. We name it as SVM + BNSL. It can be seen from Table 5 that the ensemble of BNSL and SVM outperforms single models on both datasets. Especially, BNSL +

SVM does better than SVM + BNSL, revealing that BNSL has a better capability of exploring meaningful knowledge from other sources.

Furthermore, we evaluate how BNSL performs with different numbers of relations. Figure 3 shows the results of BNSL(s) by setting relation numbers from 1 to 20. TFIDF relation selection is used for the English dataset and TF for Chinese. We can observe that BNSL performs best when we select the top 5 relations and the results become stable with more than 5 relations.
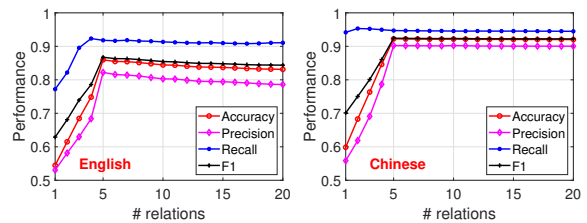

Figure 3: BNSL(s) with different numbers of relations.

Relation Selection		TF Selection				TFIDF Selection			
Dataset	Method	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score
English	SVM(s)	58.19% (10)	55.17% (10)	87.43% (6)	67.65% (11)	72.38% (10)	67.28% (10)	87.12% (10)	75.93% (11)
	BNSL(s)	71.57% (5)	67.93% (5)	81.70% (10)	74.19% (6)	86.00% (2)	82.24% (2)	91.82% (2)	86.77% (2)
	SVM + BNSL(s)	71.62% (4)	68.36% (4)	80.48% (11)	73.93% (7)	82.04% (7)	78.31% (6)	88.63% (7)	83.15% (7)
	BNSL + SVM(s)	78.46% (1)	80.55% (1)	75.04% (12)	77.70% (3)	88.36% (1)	86.48% (1)	90.94% (4)	88.65% (1)
	SVM(o)	55.07% (12)	52.91% (12)	92.29% (1)	67.26% (12)	66.65% (12)	62.64% (12)	82.48% (12)	71.21% (12)
	BNSL(o)	71.14% (7)	65.68% (7)	88.54% (5)	75.42% (4)	82.64% (5)	78.99% (5)	88.95% (6)	83.67% (6)
	SVM + BNSL(o)	66.84% (9)	61.65% (9)	89.07% (3)	72.87% (8)	78.27% (9)	74.79% (8)	85.28% (11)	79.70% (9)
	BNSL + SVM(o)	77.02% (2)	73.10% (2)	85.50% (7)	78.81% (1)	84.16% (4)	81.49% (3)	88.40% (9)	84.80% (4)
	SVM	57.38% (11)	54.36% (11)	92.05% (2)	68.35% (10)	72.15% (11)	66.46% (11)	89.45% (5)	76.26% (10)
	BNSL	71.26% (6)	66.77% (6)	84.63% (9)	74.65% (5)	84.78% (3)	80.89% (4)	91.09% (3)	85.69% (3)
	SVM + BNSL	68.31% (8)	63.71% (8)	85.09% (8)	72.86% (9)	78.70% (8)	73.99% (9)	88.50% (8)	80.60% (8)
	BNSL + SVM	75.84% (3)	70.60% (3)	88.58% (4)	78.57% (2)	82.22% (6)	76.50% (7)	93.03% (1)	83.96% (5)
Chinese	SVM(s)	89.80% (8)	86.91% (8)	93.73% (5)	90.19% (8)	74.58% (8)	67.98% (6)	92.95% (8)	78.53% (8)
	BNSL(s)	92.23% (5)	90.24% (5)	94.71% (1)	92.42% (5)	75.01% (6)	67.90% (8)	94.88% (1)	79.16% (6)
	SVM + BNSL(s)	93.31% (4)	93.13% (4)	93.52% (8)	93.32% (4)	76.37% (3)	69.62% (3)	93.55% (6)	79.83% (3)
	BNSL + SVM(s)	95.56% (1)	97.36% (1)	93.65% (7)	95.47% (1)	77.54% (2)	70.64% (2)	94.27% (4)	80.76% (2)
	SVM(o)	51.16% (12)	50.71% (12)	82.58% (9)	62.84% (10)	50.55% (12)	50.33% (12)	84.65% (10)	63.12% (10)
	BNSL(o)	51.39% (10)	50.96% (10)	73.85% (11)	60.31% (12)	50.79% (10)	50.55% (10)	72.37% (12)	59.53% (12)
	SVM + BNSL(o)	51.33% (11)	50.82% (11)	82.41% (10)	62.87% (9)	50.66% (11)	50.39% (11)	84.73% (9)	63.20% (9)
	BNSL + SVM(o)	51.72% (9)	51.18% (9)	74.54% (12)	60.69% (11)	50.97% (9)	50.68% (9)	72.98% (11)	59.82% (11)
	SVM	90.35% (7)	87.69% (7)	93.88% (4)	90.68% (7)	74.68% (7)	67.95% (7)	93.45% (7)	78.68% (7)
	BNSL	92.15% (6)	90.16% (6)	94.62% (2)	92.34% (6)	75.12% (5)	68.08% (5)	94.61% (2)	79.18% (5)
	SVM + BNSL	93.61% (3)	93.55% (3)	93.68% (6)	93.61% (3)	76.33% (4)	69.57% (4)	93.60% (5)	79.82% (4)
	BNSL + SVM	95.46% (2)	96.59% (2)	94.25% (3)	95.40% (2)	77.68% (1)	70.77% (1)	94.32% (3)	80.87% (1)

Table 5: Performance of relation selections on the entire data. The results are reported as "value + (rank)".

4.5 Analysis with missing information

In reality, the open domain facts or co-occurring sentences associated with entity-concept pairs are usually missing, making the input information for concept discovery extremely sparse. In this section, we study how BNSL performs with the sparse input. Given a set of entities, we first extract the corresponding facts (or sentences) under each concept. For both datasets, we get around 30 million entity-concept pairs for testing and more than 97% do not have the corresponding fact information with the top K relations, making the prediction of BNSL very challenging. Furthermore, both datasets have a large number of fine-grained concepts, making the task more difficult. For the missing data, we feed an empty fact or sentence into BNSL and other models for training and testing. Also, we observe that RNN does not performs as well compared with other methods and in particular RNN(sen) performs the worst when the input is extremely sparse.

In Figure 4, we report the improvement of F1-score over RNN(sen). We can observe that HypeNet, SVM and BNSL can achieve much better performance, showing their robustness with missing values. In addition, B + H can still achieve the best result. It further confirms that open domain facts and natural language sentences are comple-
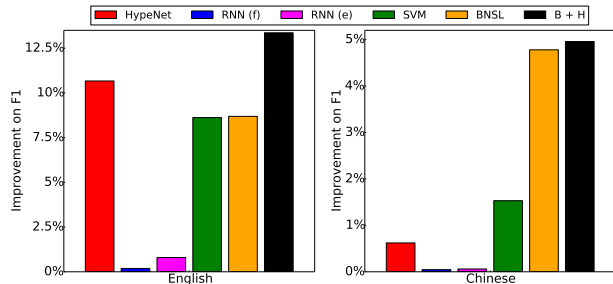

Figure 4: F1-score improvement on RNN(sen).

mentary to each other even when there is a large portion of missing information.

5 Conclusion

In this paper, we investigate the task of learning interpretable relationships between entities, relations and concepts from open domain facts to help enriching and refining concept graphs. The Bayesian network structures are learned from open domain facts as the discovered meaningful dependencies between relations of facts and concepts of entities. Experimental results on an English dataset and a Chinese dataset reveal that the learned network structures can better identify concepts for entities based on the relations of entities from open domain facts, which will further help building a more complete concept graph.

References

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary G. Ives. 2007. Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC+ASWC)*, pages 722–735, Busan, Korea.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2670–2676, Hyderabad, India.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 632–642, Lisbon, Portugal.

Cassio Polpo de Campos, Zhi Zeng, and Qiang Ji. 2009. Structure learning of bayesian networks using constraints. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 113–120, Montreal, Canada.

Haw-Shiuan Chang, Ziyun Wang, Luke Vilnis, and Andrew McCallum. 2018. Distributional inclusion vector embedding for unsupervised hypernymy detection. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 485–495, New Orleans, LA.

C. K. Chow and C. N. Liu. 1968. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Theory*, 14(3):462–467.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2011. An analysis of open information extraction based on semantic role labeling. In *Proceedings of the 6th International Conference on Knowledge Capture (K-CAP)*, pages 113–120, Banff, Canada.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2010. Recognizing textual entailment: Rational, evaluation and approaches - erratum. *Nat. Lang. Eng.*, 16(1):105.

Shimin Di, Yanyan Shen, and Lei Chen. 2019. Relation extraction via domain-aware transfer learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1348–1357, Anchorage, AK.

Jie Ding, A. Robert Calderbank, and Vahid Tarokh. 2019. Gradient information for representation and modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2393–2402, Vancouver, Canada.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam. 2011. Open information extraction: The second generation. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3–10, Barcelona, Spain.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*.

Junheng Hao, Muhao Chen, Wenchao Yu, Yizhou Sun, and Wei Wang. 2019. Universal representation learning of knowledge bases by jointly embedding instances and ontological concepts. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1709–1719, Anchorage, AK.

Sepideh Hassan-Moghaddam and Mihailo R. Jovanovic. 2018. Topology identification via growing a chow-liu tree network. In *Proceedings of the 57th IEEE Conference on Decision and Control (CDC)*, pages 5421–5426, Miami, FL.

David Heckerman, Dan Geiger, and David Maxwell Chickering. 1995. Learning bayesian networks: The combination of knowledge and statistical data. *Mach. Learn.*, 20(3):197–243.

Silu Huang, Jialu Liu, Flip Korn, Xuezhi Wang, You Wu, Dale Markowitz, and Cong Yu. 2019. Contextual fact ranking and its applications in table synthesis and compression. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 285–293, Anchorage, AK.

Tianwen Jiang, Tong Zhao, Bing Qin, Ting Liu, Nitesh V Chawla, and Meng Jiang. 2019. The role of "condition": A novel scientific knowledge graph representation and construction model. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1634–1642, Anchorage, AK.

Mikko Koivisto and Kismat Sood. 2004. Exact bayesian structure discovery in bayesian networks. *J. Mach. Learn. Res.*, 5:549–573.

Matthew Le, Stephen Roller, Laetitia Papaxanthos, Douwe Kiela, and Maximilian Nickel. 2019. Inferring concept hierarchies from text corpora via hyperbolic embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 3231–3241, Florence, Italy.

Martha Lewis. 2019. Compositional hyponymy with positive operators. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 638–647, Varna, Bulgaria.

Guiliang Liu, Xu Li, Miningming Sun, and Ping Li. 2020a. An advantage actor-critic algorithm with

confidence exploration for open information extraction. In *Proceedings of the 2020 SIAM International Conference on Data Mining (SDM)*, pages 217–225.

Guiliang Liu, Xu Li, Jiakang Wang, Mingming Sun, and Ping Li. 2020b. Large scale semantic indexing with deep level-wise extreme multi-label learning. In *Proceedings of the World Wide Web Conference (WWW)*, pages 2585–2591, Taipei.

Brandon M. Malone, Changhe Yuan, and Eric A. Hansen. 2011. Memory-efficient dynamic programming for learning optimal bayesian networks. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, San Francisco, CA.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 523–534, Jeju Island, Korea.

Ndapandula Nakashole, Gerhard Weikum, and Fabian M. Suchanek. 2012. PATTY: A taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1135–1145, Jeju Island, Korea.

Teppo Niinimaki, Pekka Parviainen, and Mikko Koivisto. 2016. Structure discovery in bayesian networks by sampling partial orders. *J. Mach. Learn. Res.*, 17:57:1–57:47.

Harinder Pal and Mausam. 2016. Demonyms and compound relational nouns in nominal open IE. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction (AKBC@NAACL-HLT)*, pages 35–39, San Diego, CA.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

Simone Paolo Ponzetto and Michael Strube. 2007. Deriving a large-scale taxonomy from wikipedia. In *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 1440–1445, Vancouver,Canada.

Ahmed Rashed, Josif Grabocka, and Lars Schmidt-Thieme. 2019. Multi-relational classification via bayesian ranked non-linear embeddings. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pages 1132–1140, Anchorage, AK.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pages 1025–1036, Dublin, Ireland.

Stephen Roller, Douwe Kiela, and Maximilian Nickel. 2018. Hearst patterns revisited: Automatic hypernym detection from large text corpora. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 358–363, Melbourne, Australia.

Arpita Roy, Youngja Park, Taesung Lee, and Shimei Pan. 2019. Supervising unsupervised open information extraction models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 728–737, Hong Kong, China.

Mauro Scanagatta, Cassio Polpo de Campos, Giorgio Corani, and Marco Zaffalon. 2015. Learning bayesian networks with thousands of variables. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1864–1872, Montreal, Canada.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 2389–2398, Berlin, Germany.

Tomi Silander and Petri Myllymäki. 2006. A simple approach for finding the globally optimal bayesian network structure. In *Proceedings of the 22nd Conference in Uncertainty in Artificial Intelligence (IJCAI)*, Cambridge, MA.

Ajit P Singh and Andrew W Moore. 2005. *Finding optimal Bayesian networks by dynamic programming*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web (WWW)*, pages 697–706, Banff, Canada.

Mingming Sun, Xu Li, and Ping Li. 2018a. Logician and Orator: Learning from the duality between language and knowledge in open domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2119–2130, Brussels, Belgium.

Mingming Sun, Xu Li, Xin Wang, Miao Fan, Yue Feng, and Ping Li. 2018b. Logician: a unified end-to-end neural approach for open-domain information extraction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining (WSDM)*, pages 556–564, Marina Del Rey, CA.

Joe Suzuki. 2010. A generalization of the chow-liu algorithm and its application to statistical learning. Technical report, arXiv:1002.2240.

Mostafa Tavassolipour, Mahmood Karimian, and Shohreh Kasaei. 2014. Event detection and summarization in soccer videos using bayesian network and copula. *IEEE Trans. Circuits Syst. Video Techn.*, 24(2):291–304.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David J. Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pages 2249–2259, Dublin, Ireland.

Fei Wu and Daniel S. Weld. 2010. Open information extraction using wikipedia. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 118–127, Uppsala, Sweden.

Ho Chung Wu, Robert Wing Pong Luk, Kam-Fai Wong, and Kui-Lam Kwok. 2008. Interpreting TF-IDF term weights as making relevance decisions. *ACM Trans. Inf. Syst.*, 26(3):13:1–13:37.

Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. 2012. Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 481–492, Scottsdale, AZ.

Mohamed Yahya, Steven Whang, Rahul Gupta, and Alon Halevy. 2014. Renoun: Fact extraction for nominal attributes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 325–335, Doha, Qatar.

Changhe Yuan and Brandon M. Malone. 2013. Learning optimal bayesian networks: A shortest path perspective. *J. Artif. Intell. Res.*, 48:23–65.

Changhe Yuan, Brandon M. Malone, and XiaoJian Wu. 2011. Learning optimal bayesian networks using A* search. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2186–2191, Barcelona, Spain.

Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3712–3722, Salt Lake City, UT.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, Berlin, Germany.