

Article

A Domain-Independent Ontology Learning Method Based on Transfer Learning

Kai Xie ^{1,2}, Chao Wang ¹ and Peng Wang ^{1,*} 

¹ School of Computer Science and Engineering, Southeast University, Nanjing 211189, China; xiekai@nrec.com (K.X.); 220194862@seu.edu.cn (C.W.)

² NR Electric Co., Ltd., Nanjing 211102, China

* Correspondence: pwang@seu.edu.cn

Abstract: Ontology plays a critical role in knowledge engineering and knowledge graphs (KGs). However, building ontology is still a nontrivial task. Ontology learning aims at generating domain ontologies from various kinds of resources by natural language processing and machine learning techniques. One major challenge of ontology learning is reducing labeling work for new domains. This paper proposes an ontology learning method based on transfer learning, namely TF-Mnt, which aims at learning knowledge from new domains that have limited labeled data. This paper selects Web data as the learning source and defines various features, which utilizes abundant textual information and heterogeneous semi-structured information. Then, a new transfer learning model TF-Mnt is proposed, and the parameters' estimation is also addressed. Although there exist distribution differences of features between two domains, TF-Mnt can measure the relevance by calculating the correlation coefficient. Moreover, TF-Mnt can efficiently transfer knowledge from the source domain to the target domain and avoid negative transfer. Experiments in real-world datasets show that TF-Mnt achieves promising learning performance for new domains despite the small number of labels in it, by learning knowledge from a proper existing domain which can be automatically selected.



Citation: Xie, K.; Wang, C.; Wang, P. A Domain-Independent Ontology Learning Method Based on Transfer Learning. *Electronics* **2021**, *10*, 1911. <https://doi.org/10.3390/electronics10161911>

Academic Editor: Rui Pedro Lopes

Received: 3 June 2021

Accepted: 5 August 2021

Published: 9 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Ontology is a kind of formal normalization description for shared conceptual model [1]. It plays a very important role in semantic web [2] and knowledge graphs (KGs) [3]. However, constructing ontologies, especially for knowledge in Web pages, is still challenging, since information in Web pages is not only massive but also heterogeneous. Nevertheless, manual construction of ontologies is time-consuming as well extremely laborious and costly process. Ontology learning aims at reducing the time and effort in the ontology development process by machine learning techniques [4–6].

Web pages are written in HTML which is a kind of semi-structured data that has a large portion of free text. Learning ontologies from the free text and semi-structured data has been widely studied in past decades [7]. Most ontology learning methods are domain-independent, because they predefine some general lexico-syntactic patterns which can be applied to text in all domains, such as Hearst patterns [8]. However, this could lead to a poor learning performance because semi-structured information in the Web page is ignored. To utilize the semi-structured information, traditional machine learning, which is based on features such as conditional random fields (CRFs) and naïve Bayes, is also widely used in ontology learning. However, these methods are based on an assumption that training data and future data are in the same feature set and hold the same distributions. Unfortunately, this assumption is impossible to hold in a real dataset, especially for data from different domains. Namely, existing methods only can achieve good performance for a specific domain. Meanwhile, humans need to do a lot of time-consuming labeling work. In conclusion, the problems in existing ontology learning methods can be summarized

as: (1) text-based methods are domain-independent but with a low performance and (2) traditional learning-based methods achieve a reasonable result, but they are domain-dependent. To handle these two problems at the same time, this paper proposes a domain-independent ontology learning method based on transfer learning.

To overcome the labeling limitations in traditional machine learning, transfer learning aims at learning knowledge from out-of-domains [9]. In a transfer learning framework, domains are divided into the target domain and the source domain. There is only a little or even no labeled data in the target domains, meanwhile, there is large-scale of labeled data in source domains. The transfer process is using labeled data from the source domain to improve the learning of the target domain. However, the source domain must be selected carefully to avoid the negative transfer. Namely, the labeled data in the source domain could have no contribution or the negative contribution to the improvement in the learning model for the target domain. The reasons why transfer learning can solve these issues are: (1) transfer learning is feature-based, so it can utilize the various information in Web pages; (2) transfer learning can learn knowledge from out-of-domains. In order to introduce transfer learning into ontology learning, there still exists some challenges. First, the transferring information should be identified. Second, we need to avoid the negative transfer. Finally, applying transfer learning to ontology learning is a nontrivial process.

To this end, this paper proposes the transfer learning model TF-Mnt for ontology learning. It consists of three phases. First, a vision-based segmentation (VIPS) algorithm [10] is used to get basic units in the Web page, which consist of some text sections and images. A vision tree which reflects the visual relation between these units will be built by VIPS. Second, concepts and instances in the corresponding ontology are recognized by a transfer learning classifier. In TF-Mnt, for Web pages in a target domain A , it can automatically select a proper source domain to improve the learning of domain A based on the domain similarity measured by the correlation coefficient. Third, *is-a* and *subclass-of* relations between concepts and instances are captured by searching some typical substructures in the vision tree. With concepts, instances, and relations, ontology for a Web page can be constructed. Finally, experiments in real-world datasets show that with an auxiliary source domain, our domain-independent ontology learning method achieves a reasonable effect.

Our contributions can be summarized as follows: (1) We propose a transfer learning model called TF-Mnt. TF-Mnt model can obtain a good ontology learning performance for a new domain with only a small set of labeled data by learning knowledge from a proper previous domain, which can be selected automatically. (2) We apply transfer learning to solve the problems in ontology learning by defining various semi-structured features and textual features in the Web page.

The paper is organized as follows: In Section 2, we review the related work. In Section 3, we give a problem description of what is a domain-independent ontology learning method and give an overview of our method. In Section 4, we present our transfer learning model and show how to apply it to learn an ontology in Section 5. In Section 6, we conduct experiments to validate our method. Finally, we conclude the paper and make some further discussions in Section 7.

2. Related Work

In this section, we will introduce related work, which includes two parts: ontology learning and transfer learning.

2.1. Ontology Learning

Many approaches for learning an ontology automatically from Web have been proposed. Some methods are based on lexico-syntactic patterns. Paul et al. proposed a pattern-based method which defined patterns such as *NP such as NP, ..., and NP* to capture *subclass-of* relations and defined patterns such as *NP is part of NP* to capture *part-of* relations [11]. Wu et al. proposed an ontology learning method combining Hearst [8] and a

probabilistic model. Hearst was used to capture terms and *is-a* relations and a probabilistic model was used to learn *subclass-of* relations and to do ambiguousness elimination [12].

Also, statistical techniques [4], symbolic techniques [13], definition-based methods [14], and mining-based methods [15] have been used in ontology learning. AliMe KG is a domain knowledge graph in the field of E-commerce that captures user problems, points of interest (POI), item information, and relations. AliMe KG is constructed by semi-automated processes for mining structured knowledge from free text. The mining process first takes as input data source such as chatlog, item detail pages and item articles, then mines nodes and predicts links, and finally output structured knowledge [15]. Alibaba constructs a large-scale e-commerce cognitive concept net AliCoCo [16]. In order to align the taxonomy of each data source, rule-based matching algorithms together with human efforts are adopted. Moreover, mining new concepts from large-scale text corpus generated in the domain of e-commerce is formulated as sequence labeling task, where the input is a sequence of words and the output is a sequence of predefined labels. To construct a task-guided taxonomy from a domain-specific corpus and allow users to input a “seed” taxonomy, serving as the task guidance. Shen et al. propose an expansion-based taxonomy construction framework HiExpan, which automatically generates key term list from the corpus and iteratively grows the seed taxonomy [17]. HiExpan views all children under each taxonomy node forming a coherent set and builds the taxonomy by recursively expanding all these sets. Furthermore, HiExpan incorporates a weakly-supervised relation extraction module to extract the initial children of a newly expanded node and adjusts the taxonomy tree by optimizing its global structure. Huang et al. propose a method CoRel for seed-guided topical taxonomy construction, which takes a corpus and a seed taxonomy described by concept names as input and constructs a more complete taxonomy based on user’s interest, wherein each node is represented by a cluster of coherent terms [18]. CoRel has two modules: relation transferring and concept learning. A relation transferring module learns and transfers the user’s interested relation along multiple paths to expand the seed taxonomy structure in width and depth. A concept learning module enriches the semantics of each concept node by jointly embedding the taxonomy and text.

All these methods mentioned above are domain-independent but they only focused on free text sections in the Web page. They can get a high precision but a low recall. To utilize semi-structured data in the Web page, Du et al. proposed a six-phase domain-independent method to learn an ontology from HTML based on DOM tree, words’ term frequency—inverse document frequency (TF-IDF) in the Web pages and HTML links [19]. However, designers of the Web pages will highlight some important keywords in some special visual ways, such as increasing the size and the weight of words. Considering these visual features, words with low TF-IDF may also be very important.

Machine learning techniques are also widely used in Web understanding which is an important part of learning an ontology from Web. Wang et al. used VIPS [10] algorithm to segment a Web page to some sections and then removed the navigation section of the Web page before labeling the rest [20]. Zhu et al. proposed a method to understand a Web page [21,22]. They used a vision tree to represent a Web page and then labeled it by a CRFs model. These attempts could not create an ontology via learning based on these labels. Yao et al. proposed a method to learn an ontology from researcher profiles [23]. They first defined the schema for the researcher profile by extending the FOAF ontology [24], and then used a CRFs model to learn instances of the concepts in a predefined schema. Craven et al. proposed a method to construct knowledge bases from Web [25]. Their method required two inputs: one was an ontology which defined the concepts and relations; another was a set of training data consisting of labeled regions of hypertext that represented instances of these concepts. Learning algorithm was naïve Bayes. Both Yao and Craven learned just the entities of the predefined schema, while new concepts and relations in the Web page could not be learned. About new relation learning, Han and Elmasri proposed a method which captured conceptual structure on a Web page by learning generic patterns [26]. Mo et al. proposed a method which focused on learning domain ontology based on VIPS

and CRFs [27]. CRFs model was used to capture concepts and instances in the Web page. Learning of relations was based on the structure of the vision tree which is generated by VIPS. However, all methods mentioned above used traditional machine learning model, so they were hard to generalize. Gao et al. first introduced a new ontology optimization schema by means of representer theorem and kernel function, and the method was a kind of linear programming [28]. Then they presented a partial multi-dividing ontology algorithm with the aim of obtaining an efficient approach to optimize the partial multi-dividing ontology learning model [29].

2.2. Transfer Learning

Transfer learning aims at improving learning effect from other domains to avoid expensive data-labeling work. Current transfer learning approaches can be divided into three categories [9]: (1) inductive transfer learning, (2) transductive transfer learning, and (3) unsupervised transfer learning. We will focus on introducing inductive transfer learning because our method is an instance of it. In inductive transfer learning, labeled data is available in the target domain, but only in small sizes. Depending on labeled data in the source domain, inductive transfer learning can be divided into (1) self-taught learning, which can allow no labeled data in the source domain, and (2) multi-task learning, which requires that data in the source domain is labeled. Our work is multi-task learning, and its goal is to utilize labeled data in the source domain to improve the target domain's learning.

Transfer learning is a hot topic and many researchers have dedicated themselves to studying it in recent years. Zhuang et al. reviewed more than forty representative transfer learning approaches, especially homogeneous transfer learning approaches, from the perspectives of data and model [30]. Transfer learning has been highly useful in low-resource domains such as drug discovery [31], protein modeling [32], predicting reactions on carbohydrates [33], pneumonia detection in chest X-ray images [34], and natural language processing (NLP) [35]. Ruder et al. presented an overview of modern transfer learning methods in natural language processing, including how models are pretrained, what information the representations they learn capture, and how learning models can be integrated and adapted in downstream NLP tasks [35]. Daume et al. proposed a domain adaptation method called Mega Model based on probabilistic graph model [36]. In their method, knowledge in a domain is divided into in-domain and out-of-domain. Transfer process is to learn out-of-domain knowledge from the source domain and in-domain knowledge from the target domain. Raina et al. proposed a transfer learning model based on informative priors, which are measured by covariance matrix, a kind of similarity measurement between the source domain and the target domain [37]. Dai et al. proposed a boosting-based transfer learning model called TrAdBoost [38]. TrAdBoost tries to find instances that conflict with the same-distribution training data and then decreases its training weight to reduce its effect. Ling et al. proposed a spectral domain-transfer learning model which constructs a similarity matrix by spectral clustering method [39]. Knowledge could be correctly transferred via the similarity matrix and a constraint matrix. Besides, many other traditional machine learning models are extended to support the transfer learning framework, such as naïve Bayes [40], logistic regression [41], SVM [42]. For the application of the transfer learning, Hu et al. applied transfer learning to do cross-domain activity recognition [43]. He used an information retrieval method to get the similarity between activities and then used weighted SVM to recognize activities based on the similarity and pseudo training data generated from source domain. The key issue of transfer learning is the similarity measurement between the source domain and the target domain. If we select two domains with low similarity, it will lead to negative transfer.

In recent years, we notice that transfer learning methods can be used for many learning scenarios such as ontology learning, and the inadequate labeled data available for the target domain with various strategies can minimize human expertise during model development and training. Tan et al. surveyed current research on transfer learning using deep neural networks and its applications [44]. They defined deep transfer learning, category and

reviewed the recent research works based on the techniques used in deep transfer learning. Vedula et al. proposed a LSTM-based framework BOLT-K to learn an ontology for a target subject or domain [45]. BOLT-K first employs semantic and graphical features to recognize the entity or concept pairs likely to be related to each other and filters out spurious concept combinations. It is then jointly trained on knowledge from the target and source domains to learn relationships among the target concepts. The target concepts and their corresponding relationships are subsequently used to construct an ontology.

3. Problem Statement

A domain-independent ontology learning method contains two parts: (1) what is a domain-independent learning model; (2) what is ontology learning. For (1), hypothetically, we have unlabeled data in n ($n > 0$) domains $DM = (DM_0, \dots, DM_n)$ and a classifier model $P(y|x)$, which is learned from labeled data in a DM . Given $\forall x_i^j \in DM_i$ in DM , where $i \in [0, n]$, we can get a proper label $y^* = \max_y P(y|x_i^j)$ by the model. Given a learning task from a new domain DM_{n+1} , a domain-independent model can achieve a good learning effect despite only having access to a small set of labeled data (it is reasonable to label a small set of data for a new domain) in DM_{n+1} . Besides, DM_{n+1} can have its own special feature space which can be different but not totally from the previous n domains. If a model can achieve a good result in this new domain, then it is domain-independent. For (2), ontology learning is to apply an automatic or semi-automatic method to construct ontologies from structured data, semi-structured data or free text, including extracting the corresponding domain's terms and the relations between those terms and finally encoding them with an ontology language for easy retrieval [4].

From these two aspects mentioned above, text-based methods can solve ontology learning problems domain-independently. However, Web pages are a kind of semi-structured data consisting of not only free text, but also various semi-structured information such as DOM structures, page links and page visions. Here, a page vision is an image that a person seeing a Web page. Text-based methods can only learn ontology from a micro level. From a macro level, semi-structured data must be considered. Machine learning techniques can combine the micro and the macro because it is feature-based. However, traditional machine learning methods are domain-dependent. Therefore, it will learn an ontology learning in the domain-dependent way. To tackle this problem, we will apply transfer learning, which is based on features and also domain-independent, to ontology learning.

The framework of our domain-independent ontology learning method is illustrated in Figure 1. It mainly consists of three stages, namely, data preparation stage, terms recognition stage, and relation learning stage. (1) During data preparation, VIPS [10] algorithm is used to segment a Web page based on vision. A vision tree which represents the Web page will be built by VIPS. In the vision tree, features from both the micro level and the macro level will be extracted. Features from micro level are about the semantic meaning of text and from macro level are about semi-structured information in the vision tree and DOM tree. Moreover, the domain-independent and domain-dependent features are also extract. (2) In terms recognition, we first obtain the knowledge from the source domain and the target domain, and then calculate the correlation coefficient between domain-independent knowledge. Furthermore, a transfer learning model called TF-Mnt will be built and it will assign a label which is in {concept, instance, none} to each text in the vision tree. See Section 4.2 for detailed information about TF-Mnt model. We will describe what this model is and how it works. Moreover, we also address how to optimize the TF-Mnt model and choose optimized parameters. See Section 4.3 for detailed information about model optimization. See Section 5.1 for detailed information about terms recognition by TF-Mnt model. (3) In the learning of relations, our method will analyze the labeled vision tree to capture *is-a* and *subclass-of* relations before constructing the ontology. The basic idea is to search some substructures which reflect *is-a* relations and *subclass-of* relations from the labeled vision tree. After terms and relations are all recognized via our method, we will

encode them by resource description framework (RDF) for further use. See Section 5.1 for detailed information about the learning of relations.

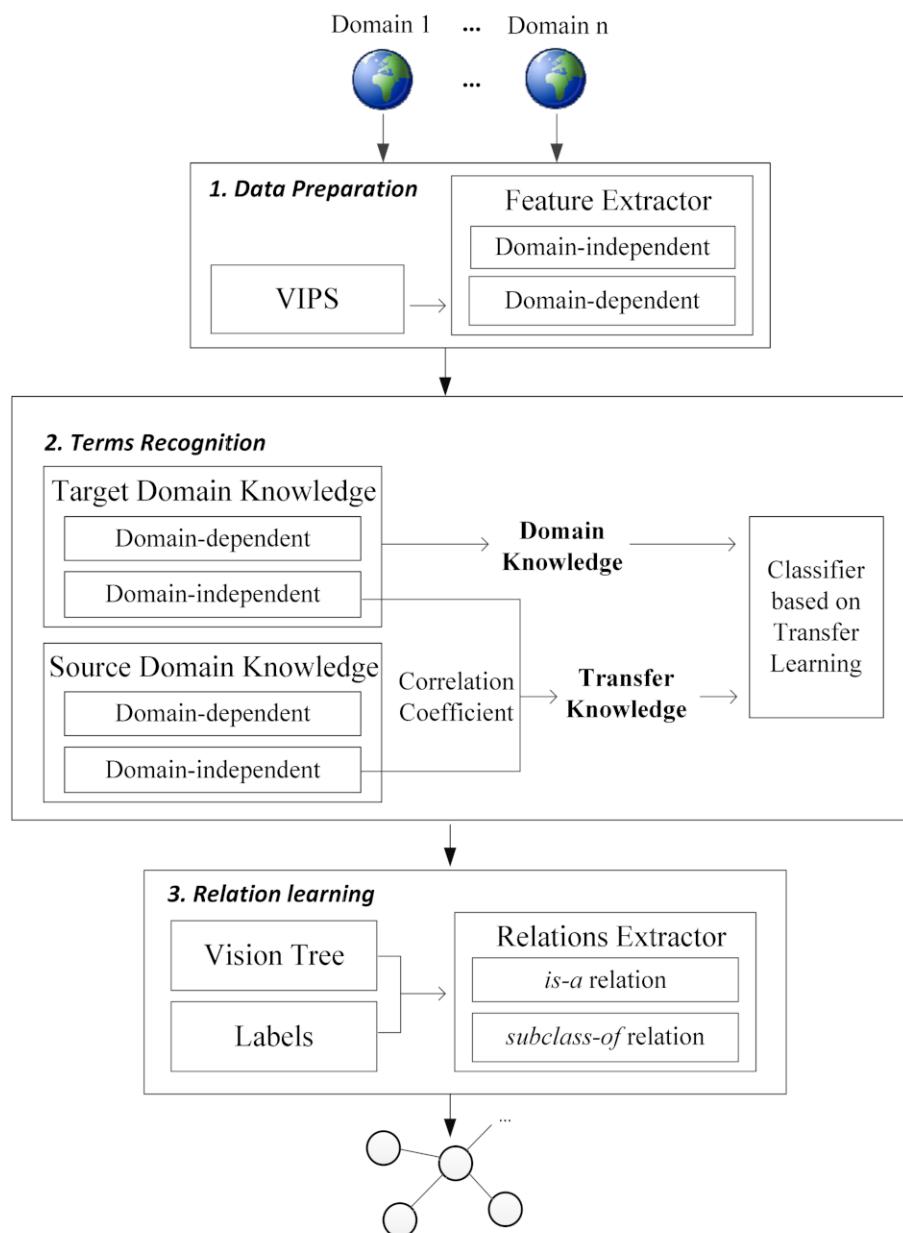


Figure 1. Overview of Our Approach.

4. Model

In this section, we will describe the details of our transfer learning model. We extend the maximum entropy (Mnt) model to make it support the transfer learning framework. The new model is called TF-Mnt, which can learn knowledge not only from in-domain, but also from out-of-domain. In Section 4.1 we will introduce transfer learning framework and the goal of it. In Section 4.2 we address our TF-Mnt model. Section 4.3 is parameter estimation of TF-Mnt.

4.1. Transfer Learning

In traditional machine learning, training data and testing data have the same input feature space and the same data distribution, therefore, knowledge of a domain A is learned

from its training data and can only be used to predict future data in domain A . If we apply traditional machine learning methods to a new domain, we need to label a lot of training data manually, that is a difficult and timing-consuming process. On the other hand, it is a waste that such expensive labels can only be used to predict data in specific domains.

Inspired by human learning process, i.e., we can use previous knowledge when studying new knowledge, that is the motivation of transfer learning. Transfer learning is used to improve a learner from one domain by transferring information from a related domain. Consider an example of a people who want to learn French. The person with an extensive Spanish knowledge will be able to learn French in an efficient manner by transferring previously language knowledge to the task of learning French. It means that one person is able to take information from a previously learned task and use it in a beneficial way to learn a related task. From the perspective of machine learning, if two domains are related, transfer learning can be used to potentially improve the results of a target learner. In addition, transfer learning occurs when there is a limited supply of target training data. This could be due to the data being rare, the data being expensive to collect and label, or the data being inaccessible [46].

The definition of transfer learning is given in following form [9]: Given a source domain D_s and learning task T_s , a target domain D_t and learning task T_t , transfer learning aims to help improve the learning of the target predictive function f_t in D_t using the knowledge in D_s and T_s , where $D_s \neq D_t$ or $T_s \neq T_t$. Transfer learning is domain-independent because it can learn knowledge from other domains. However, data in the source domain and the target domain is drawn from different distributions; therefore, one of the important issues is how to transfer knowledge despite different distributions. On the other hand, to ensure a positive transfer effect, knowledge transfer cannot be arbitrary. If two domains do not overlap and we compulsively transfer knowledge between them, it will lead to a negative result. We need consider the above two issues in our model.

We first introduce some notations for our model. We use $D_s = \{(x_i, y_i) \in X_s \times Y : 1 < i \leq N\}$ to represent data in the source domain and use $D_t = \{(x_i, y_i) \in X_t \times Y : 1 < i \leq M\}$ to represent data in the target domain, where X_s , X_t are the input spaces in source domain and target domain respectively, and Y is a finite set, and $N \gg M$. In our model, we call $X_{di} = X_s \cap X_t$ domain-independent feature space, which is shared by the source domain and the target domain; in addition, $X_{di} \neq \emptyset$. In a specific domain D , the feature space of it is $X_D = X_{di} \cup X_{dd}$, where X_{dd} is called domain-dependent feature space, which means features in X_{dd} are unique in domain D .

4.2. TF-Mnt Model

Mnt, a kind of traditional machine learning model, is widely used in text classification. The model is based on maximum entropy principle, which states that, subject to precisely stated prior data, the probability distribution that best represents the current state of knowledge is the one with largest entropy [47,48]. Mnt model is shown in Equation (1), and Equation (2) is the normalizing factor. $g_i(x, y)$ is called feature function, which is usually defined as binary function shown in Equation (3). From Equation (1), each $g_i(x, y)$ is linear weighted by w_i , so Mnt is a linear model:

$$P_w(y|x) = \frac{\exp(\sum_{i=1}^m w_i g_i(x, y))}{Z_w(x)} \quad (1)$$

$$Z_w(x) = \sum_y \exp\left(\sum_{i=1}^m w_i g_i(x, y)\right) \quad (2)$$

$$g(x, y) = \begin{cases} 1, & x = x_0, y = y_0; \\ 0, & \text{others.} \end{cases} \quad (3)$$

To make Mnt model support the transfer learning framework, we extend it to a new model as Equation (4) shows, namely TF-Mnt. In TF-Mnt, x^t is the input feature vector of the target domain, and y^* is the label sent by TF-Mnt, where $(x^t, y^*) \in (X_t, Y)$. x^{t-di} is the

domain-independent feature vector, where $x^{t-di} \in X_{di}$ and $x^{t-di} \subseteq x^t$. $Z_{w^{s-di}}(x)$ and $Z_{w^t}(x)$ are the normalizing factors. Like Mnt model, TF-Mnt model is also a linear model. In TF-Mnt model, knowledge is divided into two parts: transfer knowledge weighted by a and domain knowledge weighted by b :

$$y^* = \operatorname{argmax}_y \left[a \frac{\exp\left(\left(w^{s-di}\right)^T \cdot \rho^+ \cdot f(x^{t-di}, y)\right)}{Z_{w^{s-di}}(x)} + b \frac{\exp\left(\left(w^t\right)^T \cdot f(x^t, y)\right)}{Z_{w^t}(x)} \right] \quad (4)$$

where $Z_{w^{s-di}}(x) = \sum_y \exp((w^{s-di})^T \cdot \rho^+ \cdot f(x^{t-di}, y))$, $Z_{w^t}(x) = \sum_y \exp((w^t)^T \cdot f(x^t, y))$, and $f()$ is the vector of the feature function

Transfer knowledge consists of two parts: w^{s-di} and ρ^+ . In our model, not all knowledge in the source domain can be transferred. Only knowledge learned from domain-independent features can be transferred and this part of knowledge is called domain-independent knowledge reflected by w^{s-di} . However, we cannot use it to the target domain directly because of the distribution difference between D_s and D_t . Thus we use a diagonal matrix ρ^+ to measure the correlation coefficients. Each diagonal element in ρ^+ is a correlation coefficient ρ that is an important statistic value reflecting the relevance between two distributions. If two distributions are positively linearly dependent, then ρ will be positive; if they are negatively linearly dependent, then ρ will be negative; if they are linearly independent, ρ will be 0. Besides, the more dependent the two distributions are, the larger $|\rho|$ will be. Since the negative correlation coefficient means that two distributions are negatively linearly dependent, in our model, we set all negative correlation coefficients in the original correlation coefficient matrix to zero to avoid negative transfer. That is the correlation coefficient matrix ρ^+ . We will discuss this property in Section 6.2.1. Besides, the product of w^{s-di} and ρ^+ is called transfer coefficient. It plays an important role in source domain selection, which will be further discussed in Section 6.2.2. Note that the dependencies of the source domain and target domains could not be linear. The correlation coefficient in such case cannot be calculated by the above way. Although TF-Mnt model only considers the linear dependency, it can be extended to non-linear dependency scenarios.

A target domain's domain knowledge (denoted by w^t) is learned from labeled data in the target domain. Although there is only a small set of labeled data in it, knowledge learned from these data will be helpful to the classification because these data reflect the target domain directly.

Figure 2 illustrates the discriminant process of TF-Mnt model. To classify data in the target domain, TF-Mnt needs to learn domain knowledge from target domain's feature X_t and source domain's domain-independence feature X_{s-di} . More generally, this can be also a framework to extend other linear discriminant models like logistic regression. More intuitively, because there is only a small set of labeled data in the target domain, training an original linear classification model can fit well the small dataset, but it will lead to an under-fitting phenomenon in the whole target domain. Knowledge learned from a source domain is like a correction factor to partly correct the wrong decisions made by the original model. This idea is similar to adding a penalty term to control the over-fitting phenomenon in traditional linear classification models [36], while in our framework, we use a correction factor to overcome under-fitting phenomenon due to the small set of training data. This correction factor not only can be learned from one source domain, but also can be learned from multiple sources domain.

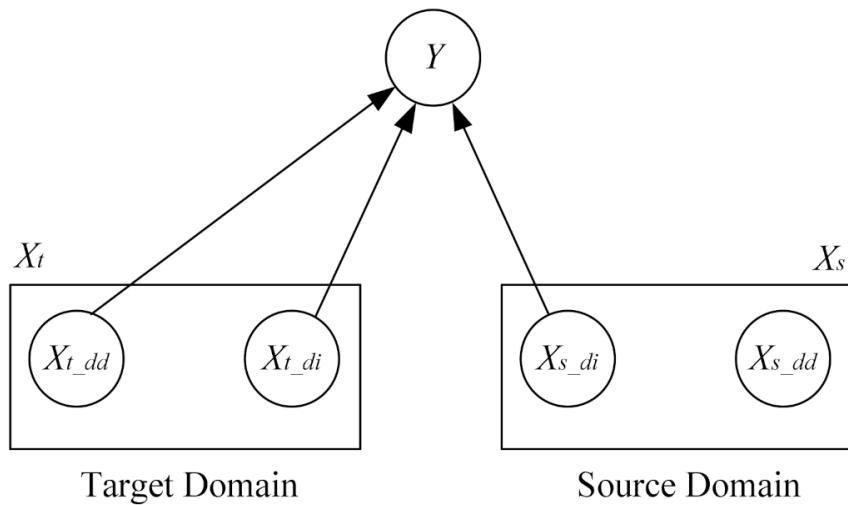


Figure 2. Discriminant Process of TF-Mnt Model.

4.3. Parameter Estimation

To implement TF-Mnt, three parameters in Equation (4) need to be estimated. They are source domain's domain-independent knowledge w^{s-di} , correlation coefficient vector ρ^+ , and target domain's domain knowledge w^t .

w^{s-di} Estimation. Our model is based on Mnt and improved iterative scaling (IIS) algorithm is widely used in Mnt parameter estimation. For D_s , we first use IIS algorithm to estimate w in formula (1). After getting parameters w in the source domain, we select domain-independent knowledge w^{s-di} , corresponding to domain-independent feature functions $f(X^{di}, Y)$, from w and then normalize them by Equation (5). Here, Y denotes a category such as concept or instance, and k is the total number of domain-independent feature functions. Finally, w^{s-di} consists of k parameters which are normalized:

$$\hat{W}_i = \frac{w_i}{|w|} \quad i = 1, 2, \dots, k \quad (5)$$

ρ^+ Estimation. ρ^+ is a $k \times k$ diagonal matrix and each diagonal element is a correlation coefficient ρ , which stands for the difference of a specific dimension between source domain's and target domain's domain-independent knowledge. ρ is a very important statistic value defined in Formula (6). $Cov(X, Y)$, the covariance between X and Y , reflects the relevance between them. $D(X)$ is the variance of X , and both $D(X)$ and $D(Y)$ are the normalizing factors. However, Equation (6) is the definition formula and cannot be used to calculate ρ directly, so we use Equation (7) to calculate it:

$$\rho_{X,Y} = \frac{Cov(X, Y)}{\sqrt{D(X)D(Y)}} \quad (6)$$

$$\rho_{X,Y} = \frac{\sum_{i=1}^n X_i Y_i - \frac{\sum_{i=1}^n X_i \sum_{i=1}^n Y_i}{n}}{\sqrt{\sum_{i=1}^n X_i^2 - \frac{(\sum_{i=1}^n X_i)^2}{n}} \sqrt{\sum_{i=1}^n Y_i^2 - \frac{(\sum_{i=1}^n Y_i)^2}{n}}} \quad (7)$$

Because ρ^+ is used to measure knowledge difference between two domains, corresponding to our model, X and Y in Equation (6) are knowledge learned from D_s and D_t . Calculating ρ^+ needs a set of samples, we thus use a bagging-based method by selecting m source domain's and target domain's labeled data randomly n times to construct n sets of training data. However, samples cannot be not fully random, because we must make sure the same latent factor in each set, otherwise ρ^+ is meaningless. Here we choose $P(Y)$ as the latent factor, that is, we must make sure $P(Y)$ in the source domain sample is equal to the target domain sample inner set. As for the inter set, $P(Y)$ must be different. Because Mnt

is a kind of discriminant model, unlike generative models (shown in Equation (8)) that depend on $P(y)$, sampling by $P(y)$ will not influence the model effect:

$$y^* = \operatorname{argmax}_y P(y|x) = \operatorname{argmax}_y \frac{P(x,y)}{P(x)} = \operatorname{argmax}_y P(x|y)P(y) \quad (8)$$

For each $y_i \in Y$, we generate $P(y_i)$ from a uniform distribution, and $P(Y)$ must satisfy $\sum_{i=1}^{|Y|} P(y_i) = 1$. Because the number of labeled data in the target domain is small, m will not be very large. Therefore, although we must build Mnt model many times, due to the small amount of training data, it will not be time-consuming. After $2n$ times (there are n sets of training data both in the source domain and the target domain) parameter estimation, we can get n groups of domain knowledge for both the source domain and the target domain. Then we select domain-independent knowledge and use Equation (5) to normalize them for both domains. Two sets of data will be constructed by above steps. One is n sets of normalized source domains' domain-independent knowledge and the other is n sets of target domains. Finally, as for ρ^+ , each diagonal element ρ can be calculated by Equation (7).

Algorithm 1 is used to calculate the correlation coefficient between the source domain and the target domain. D_s and D_t are labeled data in the source domain and the target domain respectively. We construct n samples to calculate ρ from line 2 to line 7. Line 3 is the distribution generator which generates $P(Y)$ in iteration i . Line 4 is to sample m training data based on $P(Y)$ from both the source domain and the target domain. Line 5 is Mnt trainer. After training, we obtain both domain-independent and domain-dependent parameters, which are not normalized. In line 6, parameters are normalized and domain-independent parameters are selected. Then we construct one group of sample ($w_{nor}^{t,di}$, $w_{nor}^{s,di}$) in line 7. Finally, in line 8, we calculate correlation coefficient ρ by Equation (7). After getting ρ , we will set those negative value in ρ to 0 to avoid negative transfer (line 9) to construct final ρ^+ .

Algorithm 1 Calculate Correlation Coefficient between D_s and D_t .

	Input: D_s, D_t, m, n
	Output: ρ^+
1	$\text{samples} = [(), ()]$
2	for $i=0$ to n :
3	$P_Y = \text{distributionGen}()$
4	$\text{trainingSet} = \text{randomSelect}(D_s, D_t, m, P_Y)$
5	$w^t, w^s = \text{MntTrainer}(\text{trainingSet})$
6	$w_{nor}^{t,di}, w_{nor}^{s,di} = \text{selectGeneralW}(\text{normalize}(w^t, w^s))$
7	$\text{samples.insert}(w_{nor}^{t,di}, w_{nor}^{s,di})$
8	$\rho = \text{correlationCoefficient}(\text{samples})$
9	$\rho^+ = \text{setNegativeToZero}(\rho)$
10	return ρ^+

In order to decrease the time of model training, for each previous domain, we will save all the parameters and corresponding data samples as $p = \{[P_1(Y), w_1], \dots, [P_m(Y), w_m]\}$. Then for a new domain A , when calculating ρ^+ with a previous domain B , we will sample labeled data in A by each $P_i(Y)$ in p_B . w^t Estimation. After getting w^{s-di} and ρ^+ , they are constants in Equation (4). We then use the IIS algorithm to estimate w^t . The core of IIS algorithm is trying to find a vector δ to make log likelihood in next iteration $L(w^t + \delta)$ increased until converging. For one iteration, log likelihood will increase:

$$(Lw^t + \delta) - L(w^t) = \\ \sum_{x,y} \tilde{P}(x,y)P_{w^t + \delta}(y|x) - \sum_{x,y} \tilde{P}(x,y)P_{w^t}(y|x) =$$

$$\sum_{x,y} \tilde{P}(x,y) \log \left[a \frac{\exp\left((w^{s-di})^T \cdot \rho \cdot f(x^{t-di}, y)\right)}{Z_{w^{s-di}}(x)} + b \frac{\exp\left((w^t + \delta)^T \cdot f(x^t, y)\right)}{Z_{w^t + \delta}(x)} \right] - \sum_{x,y} \tilde{P}(x,y) \log \left[a \frac{\exp\left((w^{s-di})^T \cdot \rho \cdot f(x^{t-di}, y)\right)}{Z_{w^{s-di}}(x)} + b \frac{\exp\left((w^t)^T \cdot f(x^t, y)\right)}{Z_{w^t}(x)} \right] \quad (9)$$

First, we apply $\log(a+b) \geq \log(2\sqrt{ab})$ where $a > 0, b > 0$ to Equation (9):

$$L(w^d + \delta) - L(w^d) \geq$$

$$\sum_{x,y} \tilde{P}(x,y) \log \left[2 \sqrt{a \frac{\exp\left((w^{s-di})^T \cdot \rho \cdot f(x^{t-di}, y)\right)}{Z_{w^{s-di}}(x)} * b \frac{\exp\left((w^t + \delta)^T \cdot f(x^t, y)\right)}{Z_{w^t + \delta}(x)}} \right] - \sum_{x,y} \tilde{P}(x,y) P_{w^t}(y|x) \quad (10)$$

Then we apply $-\log x \geq 1-x$ where $x > 0$ and Jensen Inequation to scale Equation (10) like the inference of Mnt IIS algorithm. Finally we can get that optimizing our model is the same as optimizing the original Mnt model. Therefore, w^t is easy to estimate by training target domain's labeled data via Mnt IIS algorithm.

5. Domain-Independent Ontology Learning

To construct the ontology, firstly, we apply the VIPS algorithm [10] to segment a Web page. The VIPS algorithm is based on the page layout features, and it first extracts all the suitable blocks from the HTML DOM tree, then tries to find the separators between these extracted blocks. Here, separators denote the horizontal or vertical lines in a Web page that visually cross with no blocks. Finally, based on these separators, the vision tree for the Web page is constructed. In the vision tree, information, which are the text (here we ignore images because an ontology does not contain images) in the Web page, is stored separately in leaf nodes while inner nodes in the vision tree reflect structure information among their children. Children of an inner node may have some similarities in the visions or semantics. Secondly, the ontology for the Web page will be constructed after the process of terms recognition and the learning of relations.

5.1. Terms Recognition by TF-Mnt

We will build a classifier to label each text, storing on each leaf node in the vision tree, to $y \in \{C, I, N\}$. Text labeled by C represent the concepts in the ontology, and I stand for instances and N means text that have no contribution to construct the ontology. This classifier is required to be domain-independent to ensure the ontology learning method is domain-independent. Therefore, we choose a transfer learning model TF-Mnt that we describe in Section 4 to construct this classifier.

In TF-Mnt, features are divided into domain-independent features and domain-dependent features. TF-Mnt model can only transfer knowledge learned from domain-independent features, so here we list all defined ones:

- Features from text, nine features total: the number of words, the font size, the font weight, the ratio of capital words, the location of colon, the ratio of domain concept keywords, the ratio of country keywords, the ratio of month keywords, and the ratio of all keywords.
- Features from DOM, four features total: contain URL, the location in the Web page, in `` and in `<h>`.
- Features from vision tree, seven features total: the depth of the text node, the type of sibling, the type of the next node, the type of the last node, next node in ``, last

node end with colon, label of the last node. (1) Node type is inner node or leaf. (2) We use depth first search (DFS) algorithm to construct a sequence and the last or the next node means the relative location in this sequence.

5.2. Learning of Relations by Patterns

Our method focuses on learning the hierarchical ontology, in which there are only hierarchical relations (subclass-of relations) between two concepts. Besides, for concepts in the ontology, if they have instances in the Web page, our method is also able to capture them (is-a relations). In terms recognition step, transfer learning is used to capture concepts and instances domain-independently. To ensure that the whole ontology learning method is domain-independent, learning of relations need to be domain-independent too.

To achieve this goal, we define three basic structural patterns to capture *is-a* relations and three to capture *subclass-of* relations. Our method will first search the vision tree to find the substructures that satisfy patterns reflecting *is-a* relations, and then clean all nodes labeled by *I* before capturing *subclass-of* relations. For instance, we list the *is-a* relation pattern in Figure 3a, in which the left part is a fragment of a Web page and the right part is the corresponding structure in the vision tree. The labels *C* and *I* come from the recognition phase terms. If a concept *C* and instances $I = \{I_1, \dots, I_n\}$ have the relative location as Figure 3a shows, we say “each $I_i \in I$ is-a *C*”. According to the process of VIPS algorithm, both *C* and *I* come from the same vision block A, and because of the different visual features between *C* and *I*, they are further separated to C and B.

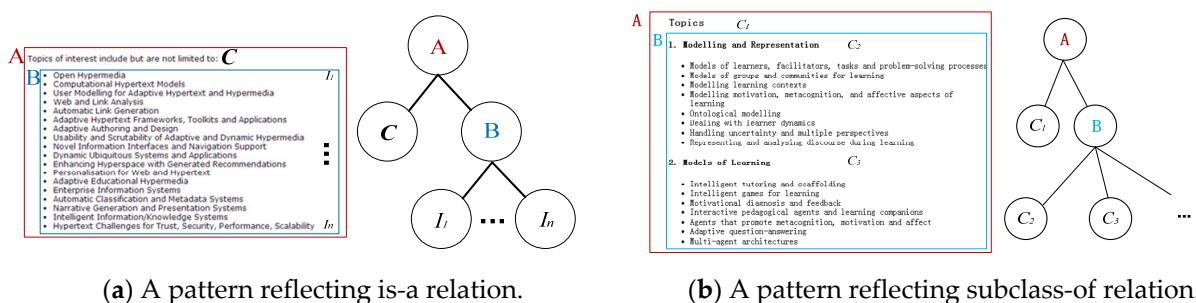


Figure 3. Patterns reflecting *is-a* relation and *subclass-of* relation.

After first search, some nodes labeled by *I* can find their concepts, while some cannot, we then clean all of them, so the only semantic nodes in the vision tree are concepts. Our method will search this tree and capture *subclass-of* relations by finding some substructures which satisfy the *subclass-of* patterns we predefine. Figure 3b illustrates an example of a pattern reflecting *subclass-of* relation. The process of learning of relations is similar to the method in [27].

6. Evaluation

This section is the experiment section about transfer learning and ontology learning. Section 6.1 introduces dataset. In Section 6.2, we conduct five experiments in total. Four of them are about the TF-Mnt model, they are how the sign of the correlation coefficient affects the model, how the transfer coefficient affects the model, how the transfer weight affects the model and how well our model works compared with other models. The last one is about learning of relations.

6.1. Dataset and Settings

We select four domains and each of them contains 50 websites. They are from wiki pages of the Fortune Global 500 companies (shortened as CM), computer science researchers’ profiles pages (shortened as P), famous computer science conferences’ pages (shortened as CF) and famous computer science journals’ pages (shortened as J). Each domain has a total of 200 pages, and we manually labeled more than 16,000 data. Table 1

shows the detailed information of our dataset. The reason why we select these four domains is that they are typical. For journal and conference pages, they have high similarity and they have a simple design and a lot of information. Because they are very formal, their developers will abide by a strict standard. For wiki pages, they have abundant information. Constructing ontologies from such pages is difficult because there are too many terms and relations in each page. For profile pages, there is a lack of standard compared with those three mentioned above as they are designed for personal use.

Table 1. The statistics of the dataset.

Domain	C	N	I	Total
CF	578	417	1261	2256
J	437	186	1796	2419
P	562	329	3503	4394
CM	1557	512	5636	7705
Total	3134	1444	12,196	16,774

We implemented our TF-Mnt model based on Python. The VIPS algorithm is retrieved from the original paper [10], while the transfer learning algorithm and the relation learning algorithm based on the maximum entropy model are developed independently. We also use the Numpy library to improve the efficiency of linear algebra computation, and employ the BeautifulSoup library to parse XML files, utilize the NLTK library for NLP tasks such as Lemmatization, and use RdfLib library for ontology generation. Finally, our model is trained on a computer with Intel(R) Xeon(R) Silver 4110 CPU 2.10 GHz and 128 G memory.

6.2. Results and Discussion

We use precision, recall, F-score and error rate as the criteria to measure the performance of our domain-independent ontology learning model. To measure the transfer learning effect, we use $TE = \frac{E_{No-TF} - E_{TF}}{E_{No-TF}}$. If $TE < 0$, then the transfer knowledge's contribution is negative and vice versa. The higher TE is, the more contributions transfer knowledge donates and vice versa. Because we use transfer learning framework, it requires that there is only a small set of labeled data in the target domain. Therefore, we split 20% of the labeled data as training set and 80% of the labeled data as testing set in a specific target domain.

6.2.1. Effects of the Correlation Coefficient

According to Section 4.3, when calculating correlation coefficients between domains, TF-Mnt has an implicit factor: the proportion of labels. Namely, the difference in the proportion of labels in the training set reflects the different distribution of knowledge that can be provided to the model. For the TF-Mnt model, the weights of the feature functions reflect the knowledge learned by the model.

We first conducted experiments on the distribution of label proportions and the weights of feature functions. The experimental results on four domains are shown in Figure 4, in which $f()$ is a feature function of word length, y is the category, C stands for concepts, and I denotes instances. Therefore, Figure 4 contains eight visualization of the distributions. The horizontal coordinate is the weight of the feature function (WOF) and the vertical coordinate is the proportion of labels in the training set (POT) corresponding to the feature function. The results show that there is a distribution between the label proportions and the weights of the feature functions, which is not scattered and unbounded. There are a few outliers, but most of them are small probability events. Further analysis shows that the weights that classified as concept are basically distributed on the positive semi-axis, i.e., it means that the text with concept labels have the similar feature of the word length. In addition, we can observe that in the company domain, the feature function weights are evenly distributed in the positive and negative halves, while the feature function weights in the other domains are basically in the positive half. This is because some of the lengths

of company names are long while concept names in other domains are short, so some of the feature weights in the company domain are in the negative half-axis.

Then we analyzed the relationship between weight distributions and correlation coefficients by experiments. As Tables 2 and 3 show, the correlation coefficients between domains with similar distribution of feature function weights are relatively large and positive, i.e., they have positive correlation; the correlation coefficients between domains with some differences in the distribution of feature function weights are relatively small and negative, i.e., they have negative correlation. Thus, it can be seen that it is feasible to use correlation coefficients to measure the similarity of feature function weights between domains. This provides a basis for using the correlation coefficient to filter the feature function weights in the source domain and select the feature function weights with higher similarity to those in the target domain for knowledge transferring.

Specifically, first, from Table 2, we can see that the correlation coefficient values between domains with large differences in distribution (profiles, journals, conferences, and companies) are negative, while between domains with small differences in distribution, the correlation coefficient is positive. The correlation coefficient between journal domain and conference domain is positive and larger, while the correlation coefficient with company domain is positive but smaller. The correlation coefficient between the conference field and the company field becomes negative. Second, from Table 3, the correlation coefficients between the profiles, conferences and company domains are negative. It is not difficult to find that the distribution of journal domain and profile domain is relatively close. The distributions of company domain and conference domain are closer. Once again, it is verified that the correlation coefficient is positive between domains if the distributions of feature function weights and training set label ratios are relatively close. If the distributions are somewhat different, the correlation coefficient is negative.

Table 2. The correlation coefficient matrix of the feature function $f(x = (1 < \# \text{OfWords} \leq 10) \&\& y = C)$.

Domain	CS-Journals	CS-Conferences	Wiki-Companies	CS-Profiles
CS-Journals	-	0.2550	0.0252	-0.0174
CS-Conferences	0.2550	-	-0.0860	-0.1042
Wiki-Companies	0.0252	-0.0860	-	-0.0252
CS-Profiles	-0.0174	-0.1042	-0.0252	-

Table 3. The correlation coefficient matrix of the feature function $f(x = (1 < \# \text{OfWords} \leq 10) \&\& y = I)$.

Domain	CS-Journals	CS-Conferences	Wiki-Companies	CS-Profiles
CS-Journals	-	0.2875	-0.0579	-0.0315
CS-Conferences	0.2875	-	-0.0753	-0.0781
Wiki-Companies	-0.0579	-0.0753	-	0.1319
CS-Profiles	-0.0315	-0.0781	0.1319	-

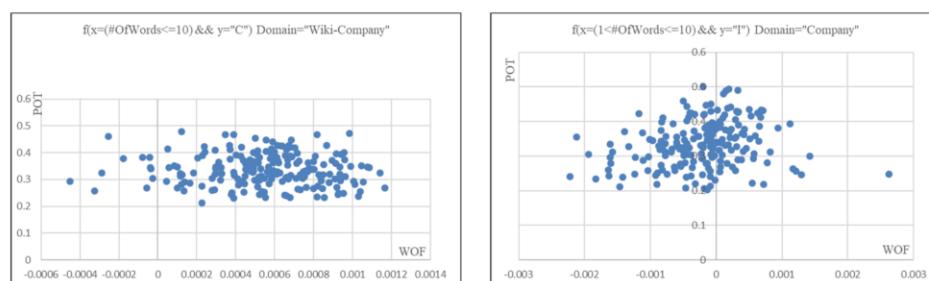


Figure 4. Cont.

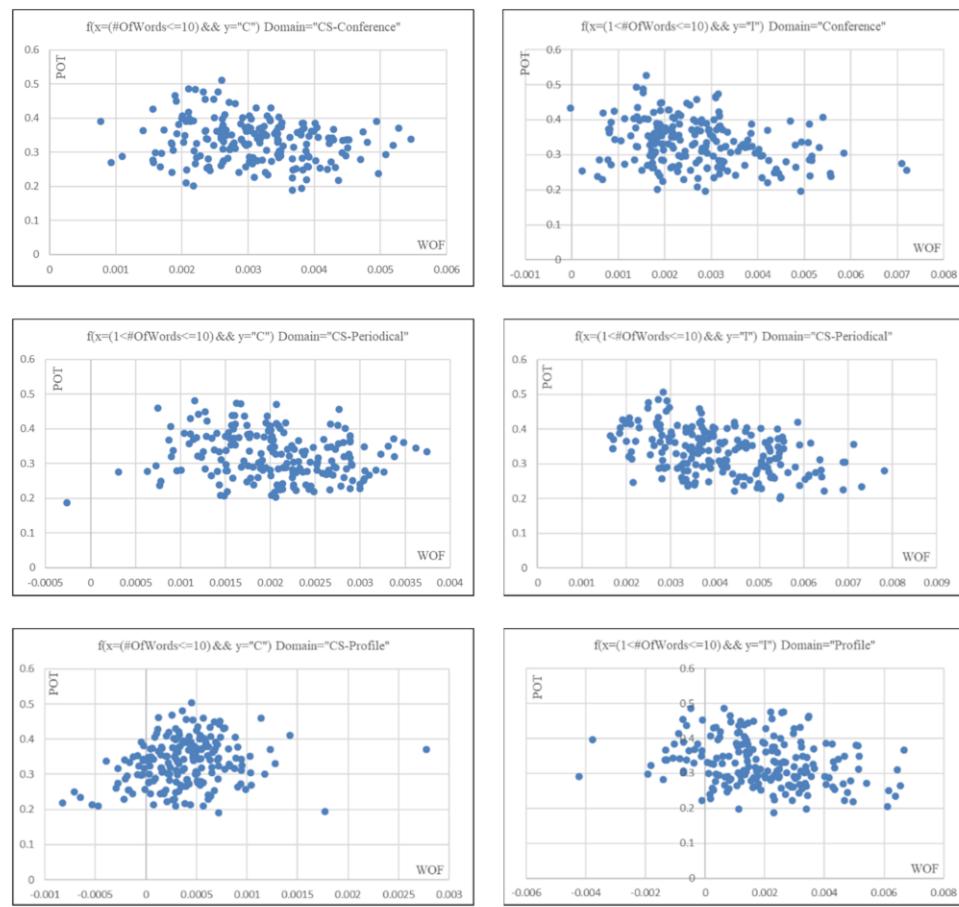


Figure 4. The distribution of label proportions and feature function weights.

We then conduct the experiment to verify how the correlation coefficient affects transfer effect. Figure 5 shows the results of three models with using original ρ (contains both + and -), only positive ρ^+ and only negative ρ^- . X-axis illustrates domain selection represented by A-B form which means the source domain is A and the target domain is B . Y-axis shows the transfer effect. From Figure 5, in most situations, using ρ^+ , the model will show a positive transfer. Once negative correlation coefficients are added into the model (using original ρ), it will decrease the transfer effect. The model with ρ^- has the worst result, showing a negative transfer phenomenon all the time. Therefore, we can conclude that positive correlation coefficient has positive contribution in classification and vice versa. We can explain this phenomenon with the reasoning that because correlation coefficient reflects linear dependence between two distributions, and if it is negative, the two distributions are negatively related, which means domain-independent knowledge of a specific dimension is important in one domain but less important in the other. Therefore, negative correlation coefficient is a factor leading to negative transfer. In following experiments, if we do not emphasize, we will set all the negative correlation coefficients to 0.

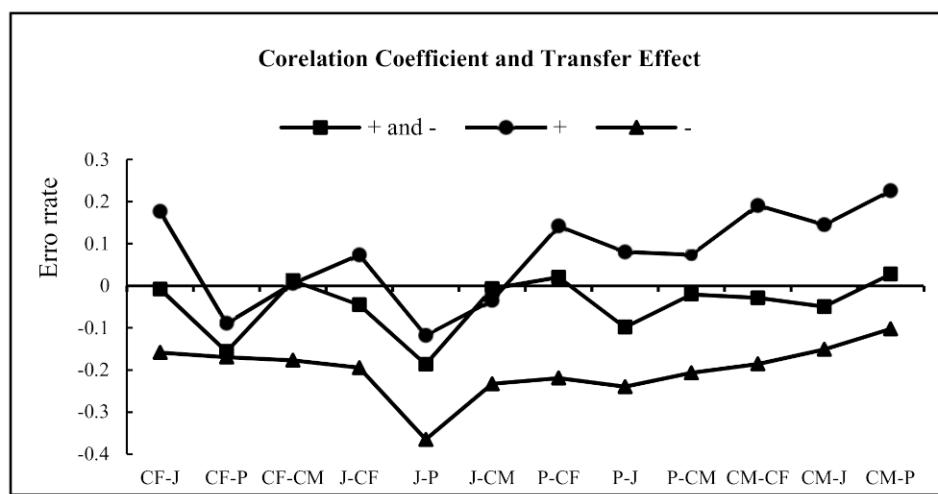


Figure 5. The error rate of correlation coefficient selection.

However, there exist three exceptions which are CF-P, J-P and J-CM. Even when using ρ^+ , their transfer effects are all negative and these situations are all negative transfers. J and CF are negative sources of P; J is a negative source of CM. Besides negative correlation coefficients, we must find other factors that lead to negative transfer.

The purpose of transfer learning is to use a large number of known samples from the source domain and combine them with a small number of known samples within the target domain to enhance the classification within the target domain. In traditional machine learning, the division of the training set and the test set is generally done according to 2:1, i.e., 66% of the training set 33% of the test set. However, since labels are very valuable, in transfer learning, the training set of the target domain is very small, and the domain knowledge it provides is very limited, which is when the importance of transfer knowledge can be reflected; while if the training set of the target domain is large, it can provide a large amount of domain knowledge, and the transfer knowledge is weakened to some extent. This experiment investigates the effect of the proportion of training sets in the target domain on the transfer effect in this case.

Figure 6 illustrates the relationship between the curves of training set partitioning and error rate under positive transfer phenomenon. Figure 7 illustrates the relationship between the curves of training set partitioning and error rate under negative transfer situations. The horizontal axis represents the proportion of training set partitioning in the target domain (POT in target), and the vertical axis has different meanings for different curves. For the curves of non-transfer and transfer, it represents the error rate of both; for the curve of transfer effect, it represents the magnitude of the transfer effect. The green curve represents the error rate of non-transfer learning model, the blue curve represents the error rate of transfer learning model, and the red curve represents the transfer effect. Once the domain is selected appropriately and positive transfer occurs, the transfer learning effect is better when the training set division is smaller than the transfer threshold; when the training set division is larger than the transfer threshold, the difference between the transfer learning and non-transfer learning models is not large. If negative transfer occurs, the effect of non-migratory learning is always better than migratory learning.

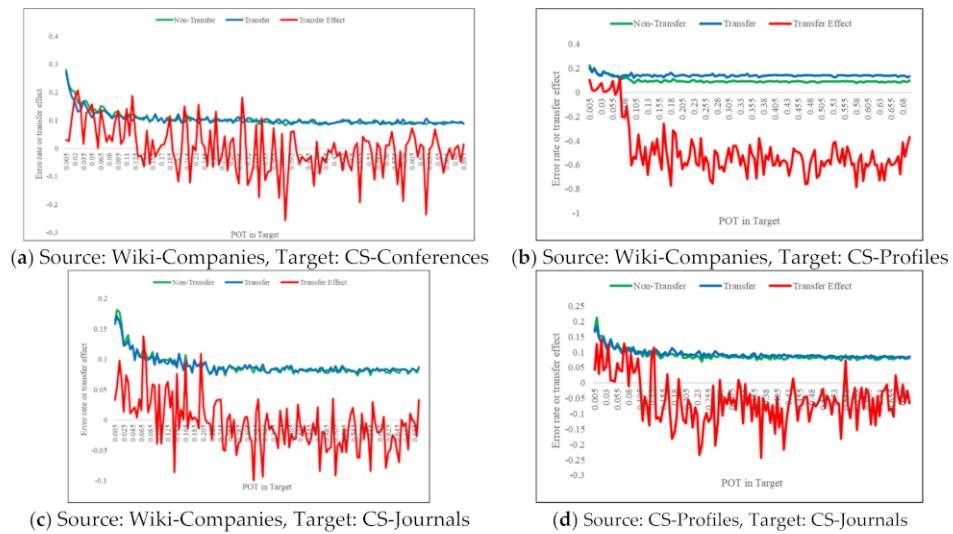


Figure 6. The relationship between the training set division and transfer effect in positive transfer.

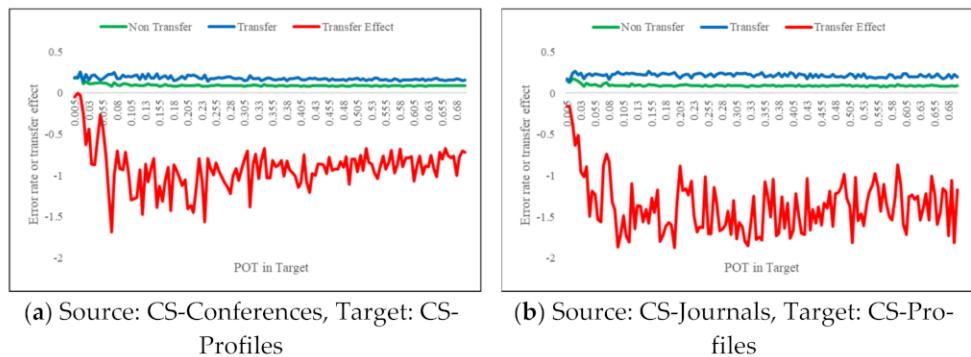


Figure 7. The relationship between the training set division and transfer effect in negative transfer.

Figure 6 shows the relationship between the training set division and the transfer effect in the positive transfer phenomenon. From Figure 6a, it can be seen that when the training set is divided at a ratio less than 0.12, the transfer learning model has better results, while when the training set division is greater than 0.12, the error rate of both the transfer learning model and the non-transfer learning model does not change much with the gradual increase of the training set division, and sometimes the positive transfer phenomenon is generated, and sometimes the negative transfer phenomenon is generated. This is because the model has reached a limit for learning domain knowledge, and after reaching this limit, it is impossible to learn new knowledge by increasing the training samples. We call this point the transfer threshold of the model, i.e., the transfer threshold is the point where the transfer effect curve intersects with the horizontal axis for the first time.

Figure 7 shows the relationship between training set partitioning and transfer effect in the negative transfer phenomenon. It can be seen from Figure 7 that if the negative transfer phenomenon is generated, the error rate of transfer learning is greater than the error rate of non-transfer learning from beginning to end, and the transfer effect of the model is less than 0.

6.2.2. Effects of the Transfer Coefficient

In TF-Mnt, as we define in Section 4.2, $(w^{s-di})^T \cdot p^+$ is called the transfer coefficient. It is the product of source domain's domain-independent parameters and correlation coefficient vector. In this experiment, we will discuss how this value affects transfer learning. Table 4 illustrates the result of the experiment, where the first column is domain selection like

the form in Section 6.2.1, the second column is the value of transfer coefficient and the third column is the transfer effect. Because there are some negative w_i in w^{s-di} , the transfer coefficient may be negative. In Table 4, for a specific target domain, we sort the source domains by the value of transfer coefficient by an ascending order. In Section 6.2.1, CF-P, J-P and J-CM appear negative transfer. From the Table 4, we can explain the reason by that the transfer coefficients of them are too low. We cannot find that transfer coefficient and transfer effect have a monotone increasing relation and cannot find a critical value to distinguish negative sources from positive ones for a specific target domain either. However, what we can find is that if we select source domains with the largest transfer coefficient, the transfer effect will be good. Therefore, for a target domain, our strategy is to choose the source domain with the largest transfer coefficient to improve target domain's learning.

Table 4. The effect of the transfer coefficient.

Domain Selection	Transfer Coefficient	Transfer Effect
J-CM	-0.0680	-0.0347
CF-CM	-0.0468	0.0056
P-CM	-0.0413	0.0992
J-P	-0.1017	-0.1179
CF-P	-0.0760	-0.0892
CM-P	-0.0399	0.2258
P-CF	-0.1167	0.1419
J-CF	-0.0958	0.0733
CM-CF	-0.0645	0.1906
CF-J	-0.0808	0.1766
P-J	-0.0700	0.0802
CM-J	-0.0645	0.1447

6.2.3. Effects of the Transfer Weight

In TF-Mnt, transfer knowledge and domain knowledge are weighted by a and b . This experiment will analyze how these weights affect transfer performance. Here we fix $b = 1$. Because a and b are linear weights, we can scale them to let $b = 1$ except the situation that $b = 0$. For $b = 0$, it means only use transfer knowledge to construct model and it is equivalence to $b = 1$ and $a \rightarrow \infty$. For $a = 0$, this situation means only use domain knowledge to construct model. Figure 8 shows the result of the experiment, x -axis shows the value of a and y -axis shows the error rate of the corresponding model. It is clear from the figure that these curves have different tendency in positive transfer Figure 8a–d and negative transfer Figure 8e–f.

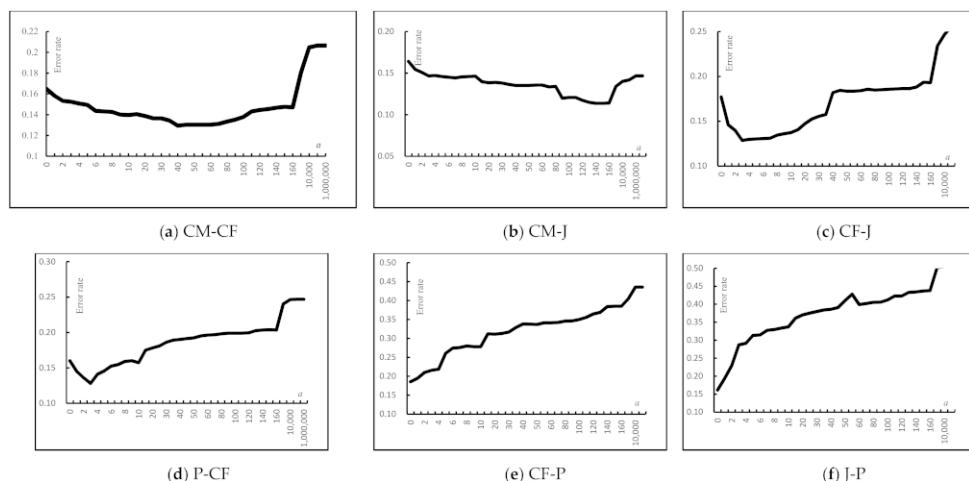


Figure 8. The effect of the transfer weight.

Figure 8a–d show four positive transfer situations. In positive transfer, the error rate shows a decreased tendency at first (smaller than the error rate at $a = 0$, so it is positive transfer) and then increasing as a becomes larger. Finally, it will approach to an asymptote which represents the error rate of only using transfer knowledge. In most situations, effect of only using domain knowledge is better than only using transfer knowledge because domain knowledge reflects the target domain directly although it learns from a few labels. However, Figure 8b is an exception, in which effect of the model with only transfer knowledge is better than the effect of the model with only domain knowledge. And from Figure 8a–d, we can find when the model reaches the best performance, a is in $(0, +\infty)$, which means only with domain knowledge or transfer knowledge is not better than combining them together by some weights. Therefore, transfer knowledge and domain knowledge are all fully used by TF-Mnt model.

Figure 8e,f show two negative transfer situations. In negative transfer, the error rate increases all the time before approaching to an asymptote which also represents the error rate of the model with only transfer knowledge. Because they are negative transfer, error rates are larger than only transferring with domain knowledge all the time. While no matter in situations of positive transfer or negative transfer, curves will finally approach to an asymptote.

6.2.4. Model Comparison

Figure 9 is the result of model comparison. We choose four famous traditional machine learning models, which are decision tree (C4.5), naïve Bayes, support vector machine (SVM) and Mnt. In this experiment, for a specific target domain, the TF-Mnt model will select the source domain with the largest transfer coefficient. Five models were used to compare the precision, recall, F-score and error rate for the four domains. The horizontal coordinates represent each of the four domains. From Figure 9, it can be concluded that the proposed model TF-Mnt has the highest F-score and the lowest error rate in the four domains.

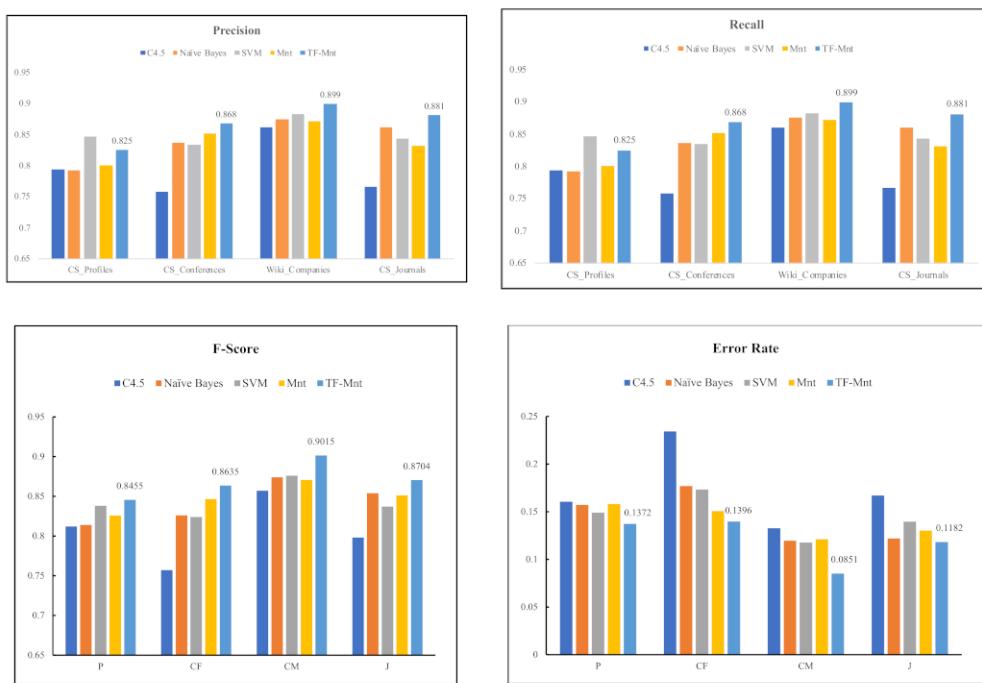


Figure 9. Comparison of performances between models.

According to the precision in Figure 9, it can be observed that in the profile domain, the highest precision is obtained using the SVM model, while in the remaining three domains,

the highest precision is obtained using the transfer learning model proposed in this paper. In all domains, the lowest precision is obtained using the decision tree model.

According to the recall in Figure 9, it can be observed that in the journal domain, the highest recall is obtained using the naïve Bayesian model, the next highest recall is obtained by the Mnt model, and the transfer learning model and the SVM model are tied for third place. In the remaining three domains, it is the transfer learning model that has the highest recalls.

According to the F-score in Figure 9, our TF-Mnt model gets the highest F-Score in all the four domains while decision tree gets the lowest F-Score. In domain P and C, SVM model also achieves a good result, and it is almost close to TF-Mnt and the naïve Bayes model represents well in CM and J. The original Mnt model also has a good performance, and although it is not the best, it is very stable.

According to the error rate in Figure 9, our model also has the best performance, getting the lowest error rate in all the situations, while decision tree also performs the worst. Naïve Bayes, SVM and Mnt are very close with some fluctuations.

In conclusion, TF-Mnt models do learn some knowledge from auxiliary data to improve learning and represent it well in these four domains according to the result of F-score and error rate but it will cost some extra time to learn the auxiliary knowledge. Decision tree is not suitable to do this classification as it gets the lowest F-score and the highest error rate in all the situations. As for naïve Bayes, SVM and Mnt, they can also achieve good results, but TF-Mnt is better than them.

6.2.5. Ontology Learning

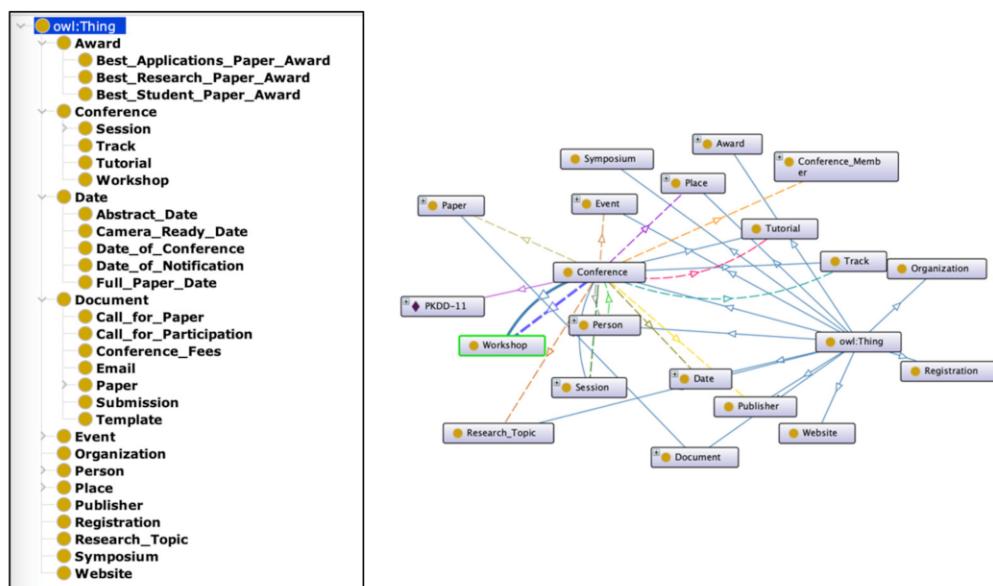
This experiment focuses on learning of relations. However, unlike terms recognition, relation is transitive. Namely, suppose in an ontology, concept A is a subclass of B and B is a subclass of C . It is insufficient, if our method only finds A is a subclass of C . Due to this property, we cannot use binary value to judge the effect of relation learning. To validate our result more precisely, we introduce two distinct values α and β , where α is the value for *subclass-of* relation and β for *is-a* relation. More specifically, if in an ontology there is a relation path $C_1 \rightarrow C_2 \rightarrow \dots \rightarrow C_k \rightarrow I_k$ (here $C_1 \rightarrow C_2$ means C_2 is a subclass of C_1), the score between (C_i, C_j) , where $0 < i < j \leq k$, is α^{j-i-1} and the score between (C_i, I_k) is β^{k-i-1} . An ontology can be represented by triple $(C, C/I, \text{subclass-of/is-a})$, while in ontology validation, we must construct tuple $(C, C/I, \text{subclass-of/is-a}, S)$, with an extra element S which reflects the score of the relation. We set $\alpha = \beta = 0.8$ in this experiment.

Table 5 shows the precision, recall, and F-score of ontology learning, which contains the results of *is-a* relation learning (the second column), *subclass-of* relation learning (the third column) and ontology learning which is the average of terms recognition and learning of relations (the fourth column). From Table 5, results of *is-a* relation learning are relatively good because most instances appear in lists in HTML, and they are easy to capture. Our methods reach 0.902 F-score in average. However, *subclass-of* relation learning is not so good, with only reaching 0.722 F-score in average. Unlike *is-a* relations, most *subclass-of* relations have less structured information. Combining terms recognition and learning of relations, our domain-independent ontology learning method can learn ontology for a Web page in a new domain with only a small set of labeled data (5 percentage of dataset in that domain) with 0.859 F-score in average.

Figure 10 illustrates a real-world example of an ontology about academic conferences, which is learned automatically by TF-Mnt model. The source domain is computer science researchers, and the target domain is computer science conferences. In Figure 10, the left part is the concept hierarchy, and the right part is the relations between concepts, which contains hierarchy and other relations. We can see that our model can extract ontology knowledge efficiently and correctly.

Table 5. Performance of ontology learning.

Domain	is-a Relation			subclass-of Relation			Ontology Learning		
	P	R	F	P	R	F	P	R	F
CF	0.894	0.911	0.902	0.754	0.760	0.757	0.873	0.866	0.869
J	0.835	0.918	0.874	0.743	0.768	0.755	0.850	0.875	0.862
P	0.918	0.907	0.912	0.890	0.689	0.745	0.895	0.843	0.868
CM	0.943	0.899	0.920	0.785	0.531	0.633	0.896	0.791	0.837
Avg.	0.897	0.909	0.902	0.773	0.687	0.722	0.879	0.844	0.859

**Figure 10.** An example of a conference ontology learned by TF-Mnt.

7. Conclusions and Future Work

This paper proposes a domain-independent ontology learning method based on the Web page vision segmentation and the transfer learning. Our goal is to solve poor learning effect problem and difficult to generalize problem in traditional ontology learning methods. We propose a new transfer learning model called TF-Mnt which divides features in different domains into domain-independent features and domain-dependent features. For a specific domain, model can transfer knowledge from other domains based on domain-independent features. The core of transfer knowledge is the correlation coefficient which is a measure of similarity between two domains and a criterion of the source domain selection. To apply TF-Mnt in ontology learning, we define semi-structured information in the Web page as domain-independent features and domain semantic meaning as domain-dependent features. With TF-Mnt model and features, a classifier can be built and then capture concepts and instances from Web pages. About learning of relations, the basic idea is to search some substructures which reflect *is-a* and *subclass-of* relations from the vision tree. With terms and relations, ontologies can be built and stored.

However, there are also some directions that we will study in the future. First, in ontology learning, because that we use VIPS algorithm to get units for classification and VIPS algorithm segments a Web page only based on the Web page vision but not by semantic, some of these units classified to concepts or instances consist of sentences. We will use some semantic methods and pattern-based methods to deal with these units by extracting core words in these sentences. For a domain, our approach is not able to generate a domain ontology, it just constructs ontologies of each Web page. For further use, these ontologies must be merged together and some inconsistencies and conflicts must be eliminated and resolved. Second, our transfer learning model is a general framework, so

we can use other linear models to do the extension such as logistic regression and so on. We also can use other values to measure the similarity between source domain and target domain. Also, TF-Mnt can be further extended to support multi sources transfer learning.

Author Contributions: Conceptualization, K.X. and P.W.; methodology, K.X. and P.W.; validation, C.W.; writing—original draft preparation, K.X., C.W. and P.W.; writing—review and editing, K.X. and P.W. All authors have read and agreed to the published version of the manuscript.

Funding: The work is supported by National Key R&D Program of China (2018YFD1100302), National Natural Science Foundation of China (No.61972082), and All-Army Common Information System Equipment Pre-Research Project (No. 31514020501, No. 31514020503).

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

References

1. McGuinness, D.L.; Van Harmelen, F. OWL Web ontology language overview. *W3C Recomm.* **2004**, *10*, 2004.
2. Berners-Lee, T.; Hendler, J.; Lassila, O. The semantic Web. *Sci. Am.* **2001**, *284*, 28–37. [CrossRef]
3. Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; Zhang, W. Knowledge vault: A Web-scale approach to probabilistic knowledge fusion. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; 2014.
4. Maedche, A.; Steffen, S. Ontology learning for the semantic Web. *IEEE Intell. Syst.* **2001**, *16*, 72–79. [CrossRef]
5. Maedche, A.; Staab, S. Ontology learning. In *Handbook on Ontologies*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 173–190.
6. Asim, M.N.; Wasim, M.; Khan, M.U.G.; Mahmood, W.; Abbasi, H.M. A survey of ontology learning techniques and applications. *Database* **2018**, *2018*, bay101. [CrossRef] [PubMed]
7. Wong, W.; Liu, W.; Bennamoun, M. Ontology learning from Text: A Look Back and into the Future. *ACM Comput. Surv.* **2012**, *44*, 20–36. [CrossRef]
8. Hearst, M.A. Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 15th International Conference on Computational Linguistics, Nantes, France, 23–28 August 1992.
9. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
10. Cai, D.; Yu, S.; Wen, J.R.; Ma, W.Y. VIPS: A Vision-Based Page Segmentation Algorithm. Microsoft Technical Report MSR-TR-2003-79. 2003. Available online: https://www.researchgate.net/publication/24347339_VIPS_a_Vision-based_Page_Segmentation_Algorithm (accessed on 9 August 2021).
11. Buitelaar, P.; Cimiano, P.; Magnini, B. (Eds.) Ontology learning from text: An overview. In *Ontology Learning from Text: Methods, Evaluation and Applications*; IOS Press: Amsterdam, The Netherlands, 2004.
12. Wu, W.; Li, H.; Wang, H.; Zhu, K. Towards a Probabilistic Taxonomy of Many Concepts. Microsoft Technical Report MSR-TR-2011-25. 2011. Available online: https://www.researchgate.net/publication/241623566_Probase_A_probabilistic_taxonomy_for_text_understanding (accessed on 9 August 2021).
13. Navigli, R.; Velardi, P. Learning domain ontologies from document warehouses and dedicated Web sites. *Comput. Linguist.* **2004**, *30*, 151–179. [CrossRef]
14. Navigli, R.; Velardi, P. Learning word-class lattices for definition and hypernym extraction. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Uppsala, Sweden, 11–16 July 2010.
15. Li, F.L.; Chen, H.; Xu, G.; Qiu, T.; Ji, F.; Zhang, J.; Chen, H. AliMeKG: Domain knowledge graph construction and application in e-commerce. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Ireland, 19–23 October 2020.
16. Luo, X.; Liu, L.; Yang, Y.; Bo, L.; Cao, Y.; Wu, J.; Li, Q.; Yang, K.; Zhu, K.Q. AliCoCo: Alibaba e-commerce cognitive concept net. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, Portland, OR, USA, 14–19 June 2020.
17. Shen, J.; Wu, Z.; Lei, D.; Zhang, C.; Ren, X.; Vanni, M.T.; Sadler, B.M.; Han, J. Hiexpan: Task-guided taxonomy construction by hierarchical tree expansion. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018.
18. Huang, J.; Xie, Y.; Meng, Y.; Zhang, Y.; Han, J. Corel: Seed-guided topical taxonomy construction by concept learning and relation transferring. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, San Diego, CA, USA, 23–27 August 2020.
19. Du, T.C.; Li, F.; King, I. Managing knowledge on the Web—Extracting ontology from HTML Web. *Decis. Support Syst.* **2009**, *47*, 319–331. [CrossRef]
20. Wang, P.; You, Y.; Xu, B.; Zhao, J. Extracting Academic Information from Conference Web Pages. In Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence, Boca Raton, FL, USA, 7–9 November 2011.
21. Zhu, J.; Zhang, B.; Nie, Z.; Wen, J.R.; Hon, H.W. Webpage Understanding: An Integrated Approach. In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, CA, USA, 12–15 August 2007.
22. Nie, Z.; Wen, J.R.; Ma, W.Y. Webpage Understanding: Beyond Page-level Search. *ACM SIGMOD Rec.* **2009**, *37*, 48–54. [CrossRef]

23. Yao, L.; Tang, J.; Li, J. A Unified Approach to Researcher Profiling. In Proceedings of the Web Intelligence, IEEE/WIC/ACM International Conference on Web Intelligence, Fremont, CA, USA, 2–5 November 2007.
24. Brickley, D.; Miller, L. FOAF Vocabulary Specification, Namespace Document. Available online: <http://xmlns.com/foaf/0.1/> (accessed on 15 January 2021).
25. Craven, M.; DiPasquo, D.; Freitag, D.; McCallum, A.; Mitchell, T.; Nigam, K.; Slattery, S. Learning to construct knowledge bases from the World Wide Web. *Artif. Intell.* **2000**, *118*, 69–113. [CrossRef]
26. Hyoil, H.; Elmasri, R. Learning rules for conceptual structure on the Web. *J. Intell. Inf. Syst.* **2004**, *22*, 237–256.
27. Mo, W.; Wang, P.; Song, H.; Zhao, J.; Zhang, X. *Learning Domain-Specific Ontologies from the Web. Linked Data and Knowledge Graph*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 132–146.
28. Gao, W.; Zhu, L.; Guo, Y.; Wang, K. Ontology learning algorithm for similarity measuring and ontology mapping using linear programming. *J. Intell. Fuzzy Syst.* **2017**, *33*, 3153–3163. [CrossRef]
29. Gao, W.; Guirao, J.L.; Basavanagoud, B.; Wu, J. Partial multi-dividing ontology learning algorithm. *Inf. Sci.* **2018**, *467*, 35–58. [CrossRef]
30. Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; He, Q. A comprehensive survey on transfer learning. In *Proceedings of the IEEE*; IEEE: Piscataway, NJ, USA, 2020; Volume 109, pp. 43–76.
31. Cai, C.; Wang, S.; Xu, Y.; Zhang, W.; Tang, K.; Ouyang, Q.; Lai, L.; Pei, J. Transfer learning for drug discovery. *J. Med. Chem.* **2020**, *63*, 8683–8694. [CrossRef]
32. Rao, R.; Bhattacharya, N.; Thomas, N.; Duan, Y.; Chen, X.; Canny, J.; Abbeel, P.; Song, Y.S. Evaluating protein transfer learning with TAPE. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 9689. [PubMed]
33. Pesciullesi, G.; Schwaller, P.; Laino, T.; Reymond, J.-L. Transfer learning enables the molecular transformer to predict regio-and stereoselective reactions on carbohydrates. *Nat. Commun.* **2020**, *11*, 1–8. [CrossRef] [PubMed]
34. Chouhan, V.; Singh, S.K.; Khamparia, A.; Gupta, D.; Tiwari, P.; Moreira, C.; Damaševičius, R.; De Albuquerque, V.H.C. A novel transfer learning based approach for pneumonia detection in chest X-ray images. *Appl. Sci.* **2020**, *10*, 559. [CrossRef]
35. Ruder, S.; Peters, M.E.; Swayamdipta, S.; Wolf, T. Transfer learning in natural language processing tutorial. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics, Minneapolis, MN, USA, 2–7 June 2019.
36. Daume, H., III; Marcu, D. Domain Adaptation for Statistical Classifiers. *J. Artif. Intell. Res.* **2006**, *26*, 101–126. [CrossRef]
37. Raina, R.; Ng, A.Y.; Koller, D. Constructing informative priors using transfer learning. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006.
38. Dai, W.; Yang, Q.; Xue, G.-R.; Yu, Y. Boosting for transfer learning. In Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, USA, 20–24 June 2007.
39. Ling, X.; Dai, W.; Xue, G.R.; Yang, Q.; Yu, Y. Spectral domain-transfer learning. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008.
40. Dai, W.; Xue, G.R.; Yang, Q.; Yu, Y. Transferring naive bayes classifiers for text classification. In Proceedings of the National Conference on Artificial Intelligence, Vancouver, BC, Canada, 22–26 June 2007.
41. Liao, X.; Xue, Y.; Carin, L. Logistic regression with an auxiliary data source. In Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 7–11 August 2005.
42. Wu, P.; Dietterich, T.G. Improving SVM accuracy by training on auxiliary data sources. In Proceedings of the 21st International Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004.
43. Hu, D.H.; Zheng, V.W.; Yang, Q. Cross-domain activity recognition via transfer learning. *Pervasive Mob. Comput.* **2011**, *7*, 344–358. [CrossRef]
44. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A survey on deep transfer learning. In *International Conference on Artificial Neural Networks*; Springer: Cham, Switzerland, 2018.
45. Vedula, N.; Maneriker, P.; Parthasarathy, S. Bolt-k: Bootstrapping ontology learning via transfer of knowledge. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019.
46. Weiss, K.; Khoshgoftaar, T.M.; Wang, D. A survey of transfer learning. *J. Big Data* **2016**, *3*, 1–40. [CrossRef]
47. Ratnaparkhi, A. A maximum entropy model for part-of-speech tagging. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Philadelphia, PA, USA, 17–18 May 1996.
48. Nigam, K.; Lafferty, J.; McCallum, A. Using maximum entropy for text classification. In Proceedings of the IJCAI-99 Workshop on Machine Learning for Information Filtering, Stockholm, Sweden, 31 July–6 August 1999.