Domain-Adaptation Technique for Semantic Role Labeling with Structural Learning

Soojong Lim, Changki Lee, Pum-Mo Ryu, Hyunki Kim, Sang Kyu Park, and Dongyul Ra

Semantic role labeling (SRL) is a task in natural-language processing with the aim of detecting predicates in the text, choosing their correct senses, identifying their associated arguments, and predicting the semantic roles of the arguments. Developing a high-performance SRL system for a domain requires manually annotated training data of large size in the same domain. However, such SRL training data of sufficient size is available only for a few domains. Constructing SRL training data for a new domain is very expensive. Therefore, domain adaptation in SRL can be regarded as an important problem. In this paper, we show that domain adaptation for SRL systems can achieve state-of-the-art performance when based on structural learning and exploiting a *prior* model approach. We provide experimental results with three different target domains showing that our method is effective even if training data of small size is available for the target domains. According to experimentations, our proposed method outperforms those of other research works by about 2% to 5% in F-score.

Keywords: Domain adaptation, semantic role labeling, natural language, semantic analysis, structured learning, prior model.

I. Introduction

Big data explosion has led to an exponential growth in the amount of valuable textual data in many fields. Thus, automatic information retrieval (IR) and information extraction (IE) methods have become more important in helping researchers and analysts to keep track of the latest developments in their fields. Current IR is still mostly limited to keyword search and unable to infer relationships between entities in a text. A system that is able to understand how words in a sentence are related semantically can greatly improve the quality of IE and would allow IR to handle more complex user queries.

Semantic role labeling (SRL) is a task for semantic processing of natural-language text, wherein the semantic role labels of the arguments associated with the predicates in a sentence are predicted. Recently, SRL has become increasingly popular as natural-language processing technology advances. The purpose of SRL is to find “who does what to whom, when, and where” in natural-language text by recognizing the semantic roles of the arguments of the predicates.

As a result of performing SRL on a given sentence and its predicate, each word in the sentence is assigned a semantic role label. By combining the labels for the words, the output of SRL can be viewed as a sequence of semantic role labels. The sequence is generated for each predicate. For example, as in Fig. 1, the semantic role A0 represents the “agent” of “wants” and the semantic role A1 denotes the thing “being wanted.” The information produced as a result of an SRL task is valuable for IE and other natural-language understanding tasks such as question answering [1] and online advertising services [2].

In previous research, most works on SRL focused on

Manuscript received July 1, 2013; revised Oct. 30, 2013; accepted Nov. 5, 2013.

This work was supported by the IT R&D program of MSIP/KEIT (10044577, Development of Knowledge Evolutionary WiseQA Platform Technology for Human Knowledge Augmented Services).

Soojong Lim (phone: +82 42 860 1297, isj@etri.re.kr), Pum-Mo Ryu (pmryu@etri.re.kr), Hyunki Kim (hkk@etri.re.kr), and Sang Kyu Park (parksk@etri.re.kr) are with the SW-Content Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Changki Lee (leeck@kangwon.ac.kr) is with the Department of Computer Science, Kangwon National University, Chuncheon, Rep. of Korea.

Dongyul Ra (corresponding author, dyra2246@gmail.com) is with the Division of Computer &Telecommunication Engineering, Yonsei University, Wonju, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.14.0113.0645

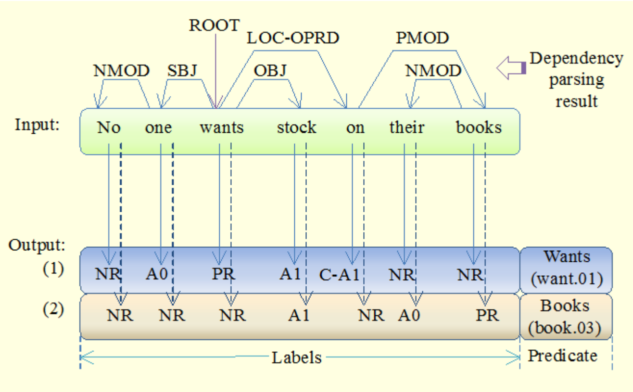


Fig. 1. Example of an SRL system's input and output.

documents from the newswire domain. While SRL systems perform well on sentences from the domain of the training data used to develop the system (source domain), such systems show a sharp performance drop when they are tested on domains other than the source domain—namely, target domains [3]–[6]. For example, all systems of CoNLL-2005 shared task [3] on SRL show a performance degradation of almost 10% or more when tested on a target domain. Although in recent years, there have been a number of efforts to apply existing SRL systems to various domains other than the source domain, development of state-of-the-art SRL systems for target domains is inhibited by a lack of large training data that comes annotated with semantic role labels. Constructing training data for a new domain is time consuming and expensive.

The task of domain adaptation is to adapt an SRL system—based upon training data from a source domain—to another target domain without experiencing significant performance drop. The domain adaptation problem is important in natural-language understanding, because there exists sufficient annotated data only for a few domains and it is very expensive to construct annotated data for new domains. With improved domain-adaptation techniques, high-performance systems can be built for a new domain for which only a small amount of annotated data is available.

In this paper, we introduce a domain-adaptation technique for developing a multi-domain SRL system. In building our system, SRL is carried out based on a structural learning model, actually a structural SVM, because it has been shown that the model is instrumental in building an SRL system with state-of-the-art performance [7]. Out of several domain-adaptation methodologies, we choose an approach originally introduced in [8]. This approach was referred to as the "*prior* model" in [9]. Based upon these two major strategies for system design, we devise a training procedure for the structural SVM in charge of SRL for target-domain texts so that the procedure can facilitate domain adaptation.

We demonstrate that our domain-adaptation technique can be applied to adapt an SRL system developed for the newswire domain (where a large annotated corpus is available) to several other target domains (for which only a small amount of annotated data is available). In this way, we can leverage existing annotated data in the newswire domain (source domain) and significantly reduce the cost of developing SRL systems for various target domains. We choose the domains "general fiction" and "biomedical" as target domains in English. In addition, we also select a "legislation" domain in German as an additional target domain. The main contributions of this paper are as follows:

- For the first time, we show that exploiting a structural learning model for an SRL domain-adaptation task can enable one to build a multi-domain SRL system that is state-of-the-art.
- We discover that combining a *prior* model approach with a structural learning model leads to an effective domain-adaptation technique for SRL.
- We demonstrate experimentally that our method is effective in domain adaption even though usage (sense) of a predicate in a target domain is different from that in a source domain.
- Ours is the first work that provides a comparative evaluation of three recently proposed domain-adaptation frameworks for the task of SRL using three target domains.

The experiments on three different target domains reveal that our proposed method outperforms other domain-adaptation strategies in developing a multi-domain SRL system.

The organization of this paper is as follows. Related research is discussed in section II. Section III explains structural learning for SRL. Section IV describes our domain-adaptation method. Experimental results are given in section V. Section VI concludes the paper.

II. Related Research

Over the years, many domain-adaptation frameworks have been proposed. Some of them focused on how to use a small amount of labeled data from a target domain in conjunction with a large amount of labeled data from a source domain [8]–[12]. Other works on domain adaption (DA) focused on adapting their models from the perspective of learning, based on the labeled data sets of the source and target domains [13], [14].

Daumé and Marcu [15] categorized and evaluated many of these DA approaches, which include the following: source-only (SRC-only) baseline method, whereby the target-domain data is ignored and training is done using only the source-domain data. In target-only (TGT-only) baseline method, training is done on only the target-domain data and source-domain data is never used. The source-and-target

http://dx.doi.org/10.4218/etrij.14.0113.0645

method uses the combined data from both domains for training. In the PRED baseline method, a SRC-only model is first built based on the source-domain data and then run on the target-domain data. The output from the SRC-only model is added as additional features to the features from the target-domain data. Finally, the system is built by using the increased feature data for training. In the linearly-interpolation (LIN-INT) baseline method, the SRC-only and TGT-only models are independently run, and their outputs are linearly interpolated to come up with the final output.

In addition to the above domain-adaptation methods, Daumé [10] introduced a feature augmentation (FA) method in which the feature space is augmented to achieve domain adaptation. The idea, proposed in [8], is to utilize the source-domain data to obtain a Gaussian distribution for parameters of maximum entropy models, which is then used as a *prior* in estimating the model parameters during adaptation using target-domain data. The final target model being trained "prefers" the *prior* weights unless the target data forces the model to take different weights. Lee and Jang [9] used the basic idea of Chelba and Acero to obtain the target structural SVM influenced by the SVM constructed for the source domain. Lee and Jang referred to this approach as the *prior* model. Note that *prior* is not used (in their case of adapting SVMs) in the statistical sense as in prior probabilities.

A structural SVM was found to be suitable for SRL [7]. In this paper, we adopt the *prior*-model approach to facilitate domain adaptation in developing a structural SVM–based SRL system. Our work is different from that of Chelba and Acero [8] in that we used the *prior*-model approach for a structural learning model of SVM; whereas they used it for a maximum-entropy Markov model. Our work is similar to Lee and Jang [9] in that both works use the *prior*-model approach for adapting structural SVMs. However, Lee and Jang have tried to apply the idea of the *prior* model in adapting the 1-slack cutting plane algorithm of 1-slack structural SVM [16]. In contrast, we use the *prior*-model approach in adapting the stochastic gradient descent (SGD)-based structural SVM for SRL.

III. Structural Learning Model for SRL

To present our domain-adaptation method for SRL, it is necessary to describe the basic model used to perform SRL in our system, especially from the point of view of machine learning. In our SRL model we adopt a structural SVM, which was developed to build an SRL system and found to be effective for performing SRL [7]. In this section, we provide explanation for theoretical aspects of the structural SVM described in that work for completeness and readability of this paper.

1. The Pegasos Framework for Building an SVM

To build a machine-learning model for binary classification, it can be assumed that we are given training data $S=\{\mathbf{x}_i, y_i\}_{i=1}^{m}$, where $\mathbf{x}_i$ is a feature vector and y_i is an output label taking either +1 or –1. A classical SVM for binary classification is a machine-learning model to solve the following constrained optimization problem [17]:

$$\min_{\mathbf{w},\xi,b}\left(\frac{1}{2}\|\mathbf{w}\|^2+\sum_i \xi_i\right), \tag{1}$$

under the constraints that $y_i(\mathbf{w}^T\mathbf{x}_i - b) \geq 1-\xi_i$ for all i, $1 \leq i \leq m$. The slack variable ξ_i for each i, $1 \leq i \leq m$, is introduced to implement an idea of soft margin. If there exists no hyperplane that can split all "yes" and "no" examples, a hyperplane will be chosen that splits the examples as cleanly as possible while allowing some misclassified examples.

The Pegasos framework is a methodology for developing a learning model for binary classification given training data like S above. However, unlike classical SVMs, it makes use of SGD schemes [18]. These schemes aim at fast computation for optimization problems. The Pegasos algorithm showed a competitive performance among the SGD methods.

The Pegasos algorithm takes the approach of finding a parameter, vector **w**, that minimizes the following unconstrained objective function:

$$f(\mathbf{w};A_t)=\frac{\lambda}{2}\|\mathbf{w}\|^2+\frac{1}{k}\sum_{(\mathbf{x}_i,\mathbf{y}_i)\in A_t} l(\mathbf{w};(\mathbf{x}_i,y_i)), \tag{2}$$

where the loss function $l(\mathbf{w};(\mathbf{x}_i,y_i))=\max\{0,1-y_i\mathbf{w}^T\mathbf{x}_i\}$. The parameter λ is for regularization. The subset A_t of S is prepared by selecting its members randomly from S. Its cardinality $|A_t|$ is denoted by k. If the subgradient of the approximate objective is taken, it is

$$\nabla f(\mathbf{w};A_t)=\lambda\,\mathbf{w}-\frac{1}{k}\sum_{(\mathbf{x}_i,y_i)\in A_t^+} y_i\mathbf{x}_i\,. \tag{3}$$

In (3), $A_t^+=\{(\mathbf{x},\ y)\in A_t : y\mathbf{w}^T\mathbf{x}<1\}$. Following the principle of gradient update, **w** is set to a new value $\hat{\mathbf{w}}=\mathbf{w}-c\nabla f(\mathbf{w})$. The variable c represents a preset learning rate.

2. Structural SVM

Because our SRL component needs to carry out sequence labeling to find the semantic role labels of the words in a sentence, a model for binary classification is not enough. We need a model with a structural output such as a label sequence.

Tsochantaridis and others [19] introduced structural SVMs that can produce structural outputs such as trees or sequences. In this structural learning problem, the output label y_i in training examples of binary classification should be switched to $\mathbf{y}_i$, taking a structural value such as a label sequence. In their framework, a discriminant function F: $X \times Y \rightarrow \mathbb{R}$ is exploited, where X is the input space, Y the output space, and $\mathbb{R}$ the set of all real numbers. The discrimmant function F is formed as the inner product of the vectors $\mathbf{w}$ and $\Psi(\mathbf{x}, \mathbf{y})$ as follows:

$$F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^T \Psi(\mathbf{x}, \mathbf{y}), \tag{4}$$

where $\mathbf{w}$ is a parameter vector and $\Psi(\mathbf{x}, \mathbf{y})$ is a feature vector that represents the input/output pair ($\mathbf{x}$, $\mathbf{y}$).

For a given input $\mathbf{x}$, F is used to generate a prediction (output) by choosing $\hat{\mathbf{y}}$ as an output in such a way that F is maximum at $(\mathbf{x}, \hat{\mathbf{y}})$ among all possible $\mathbf{y}$.

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y} \in \boldsymbol{Y}} F(\mathbf{x}, \mathbf{y}; \mathbf{w}). \tag{5}$$

The problem of learning the structural SVM is to find a parameter vector $\mathbf{w}$ that is optimal according to the given training data S={($\mathbf{x}_i$, $\mathbf{y}_i$): i = 1, 2,…, m}. Following the margin-rescaling paradigm, the structural SVM model [18] is formulated as a constrained optimization problem as follows:

$$\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i, \quad \text{s.t. } \forall i \in [1, m], \ \xi_i \geq 0 \ \text{ and}$$

$$\forall i \in [1, m], \forall \mathbf{y} \in \boldsymbol{Y} \setminus \mathbf{y}_i : \mathbf{w}^T \delta\Psi_i(\mathbf{x}_i, \mathbf{y}) \geq L(\mathbf{y}_i, \mathbf{y}) - \xi_i. \tag{6}$$

In (6), it is defined that $\delta\Psi_i(\mathbf{x}_i, \mathbf{y}_i) = \Psi(\mathbf{x}_i, \mathbf{y}_i) - \Psi(\mathbf{x}_i, \mathbf{y})$. Hamming loss function L($\mathbf{y}_i$, $\mathbf{y}$) is the count of the element positions of the input vectors at which the corresponding elements of $\mathbf{y}$ and $\mathbf{y}_i$ are not the same. The symbol C indicates the regularization constant. Removing an element from a set is what is meant by the symbol "\" in $\boldsymbol{Y} \setminus \mathbf{y}_i$.

3. Structural SVM for SRL

In a similar way that the Pegasos framework was used to build an efficient learning model of (2) for binary classification problems originally given as (1), the Pegasos algorithm was applied to the structural learning problem of (6) to obtain an efficient structural learning model, which is actually a structural SVM for SRL. In this subsection, we provide a brief description on how this was done in [7].

In a core component of an SRL system, the input consists of both a sentence and a predicate, and the output is a label sequence. Therefore, training data D for an SRL can be represented as follows:

$$D = \{(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}) : i = 1, 2, \ldots, m$$

$$\text{and } \ j = 1, \ldots, m_i \ \text{ for each } \}i. \tag{7}$$

Note that a predicate $\mathbf{pr}$ needs to be added as input to the discriminant and feature functions F and Ψ as in $F(\mathbf{x}, \mathbf{pr}, \mathbf{y}; \mathbf{w})$ and $\Psi(\mathbf{x}, \mathbf{pr}, \mathbf{y})$.

Following the Pegasos framework, the unconstrained objective function for the structural SVM can be chosen as follows:

$$f(\mathbf{w}; A_t) = \frac{\lambda}{2}\|\mathbf{w}\|^2 + \frac{1}{k}\sum_{(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}) \in A_t} l(\mathbf{w}; (\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij})). \tag{8}$$

The model needs to find an optimal vector $\mathbf{w}$ that minimizes f without any constraints. We choose A_t, of size k, randomly from D. The loss function l is defined to be $l(\mathbf{w}; (\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}))$=$\max\{0, \max\{L(\mathbf{y}_{ij}, \mathbf{y}) - \mathbf{w}^T \delta\Psi_{ij}(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y})\}\}$, where $\delta\Psi_{ij}(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}) = \Psi(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}) - \Psi(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y})$. As explained previously, $L(\mathbf{y}_{ij}, \mathbf{y})$ is the Hamming loss function. If we take the subgradient of $f(\mathbf{w}; A_t)$, we obtain

$$\nabla f(\mathbf{w}; A_t) = \lambda\mathbf{w} - \frac{1}{|A_t|}\sum_{(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}) \in A_t^+} \delta\Psi_{ij}(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}^*), \tag{9}$$

where $\mathbf{y}_{ij}^* = \arg\max\{L(\mathbf{y}_{ij}, \mathbf{y}) - \mathbf{w}^T \delta\Psi_{ij}(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y})\}$ and

$A_t^+ = \{(\mathbf{x}, \mathbf{pr}, \mathbf{y}) \in A_t : l(\mathbf{w}; (\mathbf{x}, \mathbf{pr}, \mathbf{y})) > 0\}$.

Let $\mathbf{w}_t$ be the parameter vector at any point during training. Then, the updated parameter $\mathbf{w}_{t+1}$ is obtained by setting it to $\mathbf{w}_t - \eta_t \cdot \nabla f(\mathbf{w}_t; A_t)$, where $\eta_t = 1/(\lambda t)$ is the learning rate. Using (9), we obtain

$$\mathbf{w}_{t+1} = (1 - \eta_t\lambda)\mathbf{w}_t + \frac{\eta_t}{k}\sum_{(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}) \in A_t^+} \delta\Psi_{ij}(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}^*). \tag{10}$$

IV. Developing a Multi-domain SRL System

In this section, we explain how a multi-domain SRL system can be constructed based on the structural SVM introduced in the previous section. In particular, we describe how the structural SVM for the source domain is adapted to accommodate the *prior* model to be effective for domain adaptation.

1. Basic SRL System for Source Domain

In developing a multi-domain SRL system, we first build an SRL component for the source domain, which is our basis subsystem. The structural SVM for SRL explained in section III, subsection 3, is used to build our basic source-domain SRL component. For training the structural SVM, we follow the method that was introduced in [7].

It is assumed that the training data D_{src} for the source domain is available. Specifically we used the procedure shown in Algorithm 1 for building the source-domain SRL component. This training procedure is based on the weight-parameter update scheme given in (10).

2. Domain Adaptation with Structural SVM

In the scenario of domain adaptation, the model built on the source-domain data undergoes a domain-adaptation process by utilizing the target-domain data. The performance of the multi-domain SRL system will decrease dramatically if the model trained on the source domain is applied directly to the target domain without domain adaptation.

As our domain-adaptation scheme, we take the approach referred to as the *prior* model by Lee and Jang [9] and originally proposed by Chelba and Acero [8]. The basic intuition behind the training process following the *prior* model is that it keeps the target-domain model as close to the source-domain model as possible; unless there is strong evidence in the target-domain data to move the newly trained model away from the source-domain model.

```
Algorithm 1. A training method for source domain
Inputs:  $D_{src}$, $\lambda$, $T$, $k$
1: $\mathbf{w}_1 = 0$ // Initialization.
2: For $t = 1, 2, \ldots, T$ do
3:    Select $A_t \subseteq D_{src}$,  where $|A_t| = k$
4:    $A_t^+ = \{(\mathbf{x}, \mathbf{pr}, \mathbf{y}) \in A_t : l(\mathbf{w}_t; (\mathbf{x}, \mathbf{pr}, \mathbf{y})) > 0\}$.
5:    $\forall (\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}) \in A_t^+$:
          $\mathbf{y}_{ij}^* = \text{argmax}\{L(\mathbf{y}_{ij}, \mathbf{y}) - \mathbf{w}_t^T \Psi(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y})\}$
6:    $\eta_t = 1/\lambda t$
7:    $\mathbf{w}_{t+1/2} = (1 - \eta_t \lambda)\mathbf{w}_t$
              $+ \frac{\eta_t}{k} \sum_{(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}) \in A_t^+} \delta \Psi_{ij}(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}^*)$
8:    $\mathbf{w}_{t+1} = \min\left\{1, \frac{1/\sqrt{\lambda}}{\|\mathbf{w}_{t+1/2}\|}\right\} \mathbf{w}_{t+1/2}$
9: Return  $\mathbf{w}_{T+1}$ as output
```

Following the *prior* model, our multi-domain SRL system is constructed by utilizing a domain-adaptation method that consists of two training stages, as depicted in Fig. 2. The source-domain model constructed with a structural SVM is built by performing training with source-domain training data D_{src}. The basic SRL system introduced in the previous subsection is the system constituting this stage. As a result, weight vector $\mathbf{w}_{src}$ is obtained, which represents the source (domain) model.

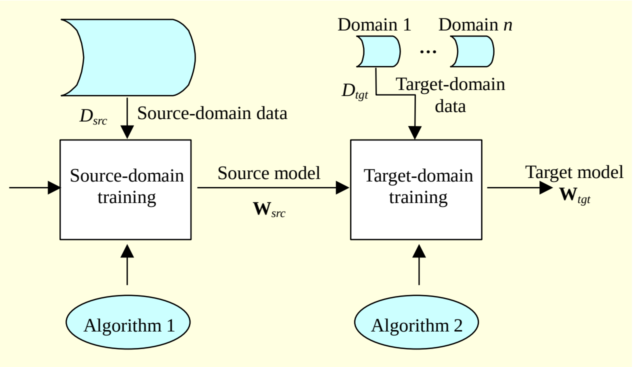


Fig. 2. Domain adaptation with two-stage training.

In the second stage, the target model is acquired by carrying out the training of the structural SVM using the target-domain training data D_{tgt}. Our multi-domain SRL system can be easily ported from one target domain to another by choosing a target domain and feeding its data as D_{tgt} in this second stage. Another input to stage two is the source-model weight vector $\mathbf{w}_{src}$ resulting from the first stage, which is to realize the idea of the *prior* model. The training procedure for stage two needs to be developed so that a domain-adaptation effect can be achieved by the resulting target model.

To accommodate the *prior* model, we use an objective function, which is obtained by modifying (8) as follows:

$$f(\mathbf{w}; A_t) = \frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}_{src}\|^2 + \frac{1}{k} \sum_{(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}) \in A_t} l(\mathbf{w}; (\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij})). \quad (11)$$

This formula is based upon the idea that the closer the new model $\mathbf{w}$ is to the source model $\mathbf{w}_{src}$, the better it is; while it also manages to minimize the second term of (11) in parallel, which corresponds to the attempt of satisfying the constraints of the original optimization problem. The subgradient of $f(\mathbf{w}; A_t)$ is

$$\nabla f(\mathbf{w}; A_t) = \lambda(\mathbf{w} - \mathbf{w}_{src}) - \frac{1}{|A_t|} \sum_{(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}) \in A_t^+} \delta \Psi_{ij}(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}^*). \quad (12)$$

As explained before, $\mathbf{w}_t$ is updated to $\mathbf{w}_{t+1}$ by $\mathbf{w}_t - \eta_t \cdot \nabla f(\mathbf{w}_t; A_t)$, where the learning rate η_t is 1/(λt). By using (12), the update formula becomes

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \lambda (\mathbf{w}_t - \mathbf{w}_{src}) + \frac{\eta_t}{k} \sum_{(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}) \in A_t^+} \delta \Psi_{ij}(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}^*). \quad (13)$$

The training procedure using the target-domain data based upon (13)—which reflects our domain-adaptation strategy—is given in Algorithm 2.

Algorithm 2 receives five inputs: D_{tgt} (training data for target domain), λ (regularization constant), T (the preset number of iterations), k (the number of examples for calculating the subgradients), and $\mathbf{w}_{src}$ (the weight vector trained on the source-domain data). On each iteration t, the algorithm randomly chooses the set A_t, of cardinality k, from the training data D_{tgt} (line 3) and determines A_t^+ consisting of training examples with positive loss (line 4). Then it computes the label sequence $\mathbf{y}_{ij}^*$ with "largest violation" for every $(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij})$ in A_t^+ (line 5). Updating $\mathbf{w}_t$ to $\mathbf{w}_{t+1}$ according to (13) is done at line 7.

Algorithm 2. A S-SVM.Prior algorithm for SRL

```
Inputs: $D_{tgt}, \lambda, T, k, \mathbf{w}_{src}$
1: $\mathbf{w}_1 = 0$ // Initialization.
2: For $t = 1, 2, \ldots, T$ do
3:     Choose $A_t \subseteq D_{tgt}$, where $|A_t| = k$
4:     Set $A_t^+ = \{(\mathbf{x},\mathbf{pr},\mathbf{y}) \in A_t : l(\mathbf{w}_t;(\mathbf{x},\mathbf{pr},\mathbf{y})) > 0\}$
5:     $\forall(\mathbf{x}_i,\mathbf{pr}_{ij},\mathbf{y}_{ij}) \in A_t^+$:
           $\mathbf{y}_{ij}^* = \mathrm{argmax}\{L(\mathbf{y}_{ij},\mathbf{y}) - \mathbf{w}_t^T \Psi(\mathbf{x}_i,\mathbf{pr}_{ij},\mathbf{y})\}$
6:     $\eta_t = 1/\lambda t$
7:     $\mathbf{w}_{t+1/2} = \mathbf{w}_t - \eta_t \lambda(\mathbf{w}_t - \mathbf{w}_{src}) + \frac{\eta_t}{k} \sum_{(\mathbf{x}_i,\mathbf{pr}_{ij},\mathbf{y}_{ij}) \in A_t^+} \delta\Psi_{ij}(\mathbf{x}_i, \mathbf{pr}_{ij}, \mathbf{y}_{ij}^*)$
8:     $\mathbf{w}_{t+1} = \min\left\{1, \frac{1/\sqrt{\lambda}}{\|\mathbf{w}_{t+1/2}\|}\right\} \mathbf{w}_{t+1/2}$
9: Return $\mathbf{w}_{T+1}$ as output
```

V. Experiments on Performance

1. Data Sets

For SRL experiments, we choose the newswire domain (the *Wall Street Journal* corpus from CoNLL-2008 Shared Task) as the source domain. The first target domain in our experiment is the biomedical domain (BioProp). In addition, we also choose the general fiction domain (Brown corpus from CoNLL-2008 Shared Task) as another target domain.

In SRL data, predicates are given for each sentence, and the system has to predict semantic roles for each predicate. In the training data of the *Wall Street Journal* (WSJ) and Brown corpora, semantic role annotation is available for all verbs and nouns. In the case of BioProp, the creators of annotated BioProp concentrated on 30 important or frequent verbs from the biomedical domain.

BioProp was created from 500 MEDLINE article abstracts. The articles were selected based on the keywords: human, blood cells, and transcription factor. To our knowledge, BioProp is the only resource for biomedical SRL that uses full syntactic parse trees. The dependency parse trees are available from the GENIA Treebank [20] using constituent-to-dependency conversion [5].

Table 1. Datasets of source and target domains in English.

	Source data	Target data	
	Newswire	General fiction	Biomedical
Sentences	36,090	404	1,635
Unique predicates	8,408	702	30
Training examples	182,303	1,280	1,982
Overlapped predicates with source predicates	-	617	26

The statistics of the data sets are given in Table 1. It is obvious that Brown corpus and BioProp are much smaller than WSJ corpus, not only in terms of the number of sentences, but also in the number of predicate-argument structure and verbs that are covered.

In addition to English, we use the newswire domain (TIGER newspaper corpus) as the source domain and use the legislation domain as the target domain (sampled from the EUROPARL corpus) in German. These corpora are a part of CoNLL-2009 Shared Task.

2. Experimental Setup

We have implemented our SRL system for domain adaptation with structural learning for experimentation. The performance of our basic SRL system (the source model tested on the source-domain test data) on CoNLL-2008 data (WSJ corpus) is measured to be 83.21% in F-score.

We have carried out two different experimentations. The goal of the first experimentation is to compare the various domain-adaptation methods, including ours, based on their performances on both of our target domains. For this purpose, we have constructed three SRL modules corresponding to FA, PRED, and our proposed method. The aim of the second experimentation is to have a more sophisticated evaluation. In particular, we want to see how our proposed method performs in cases where SRL becomes more difficult. For example, it is when the usage (meaning) of a predicate in the target domain is different from that in the source domain. For this experimentation, the biomedical domain has been chosen as the target domain.

All experiments use five-fold cross validation on the target-domain data set. The training examples in the target-domain data set are divided into five partitions of equal size. Partitioning is done randomly to guard against any selection bias. Our system is built using four of the partitions plus the whole source-domain data for training and is tested on the

http://dx.doi.org/10.4218/etrij.14.0113.0645

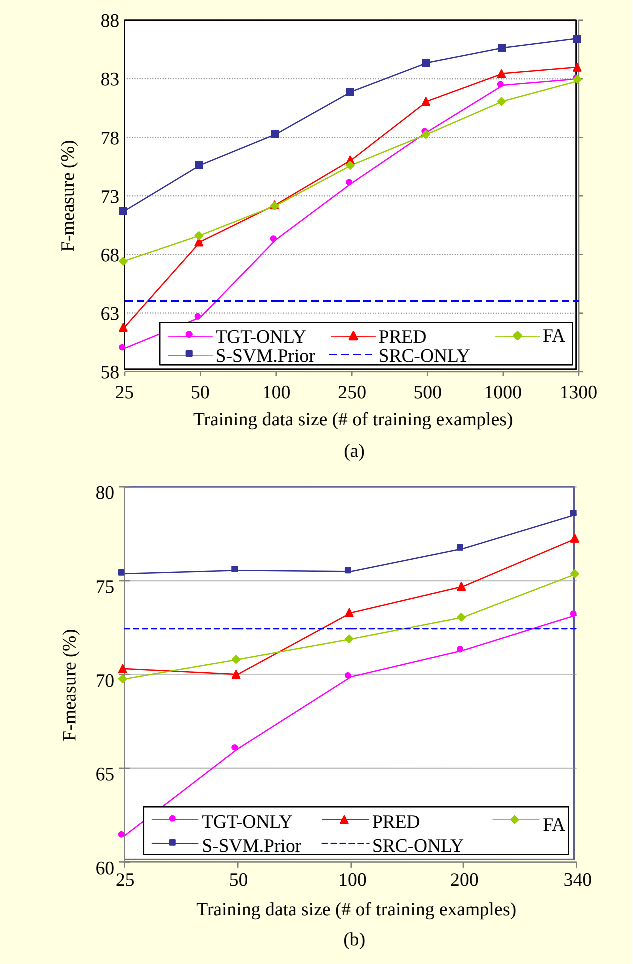
remaining partition of the target-domain data. All experiments have been carried out under the environment of Intel Core i5 CPU of 3.40 GHz with 32 GB RAM and Linux with 64-bit OS.

3. Evaluation Metrics

To measure performance of our system, we have used the evaluation tool distributed for CoNLL-2008 with no change. Our system is evaluated in terms of precision (*p*), recall (*r*), and F-measure. Precision measures how accurate the predictions are. It is calculated as the number of correct predictions divided by the total number of predictions. Recall is measured as the number of correct predictions divided by the actual number of relevant instances in the test set. F-measure combines precision and recall into a single metric by computing the harmonic mean of the two.

4. Experimental Results

Our proposed domain-adaptation method is called S-SVM.Prior in the experiments. The purpose of the first experimentation is to see how effective our proposed method is for domain adaptation in general. The experimental result on the biomedical domain as target is shown in Table 2 and Fig. 3(a).



(a)

(b)

Fig. 3. Performance results for various DA methods: (a) biomedical domain and (b) general-fiction domain.

Table 2. F-measures of compared methods for biomedical domain.

	0	25	50	100	250	500	1,000	1,300
SRC-only	64.09	-	-	-	-	-	-	-
TGT-only	-	59.99	62.59	69.31	74.08	78.46	82.42	82.99
PRED	-	61.80	69.08	72.25	76.09	81.06	83.44	84.02
FA	-	67.41	69.66	72.22	75.65	78.24	81.07	82.79
S-SVM.Prior	-	71.70	75.68	78.31	81.90	84.39	85.66	86.41

Table 3. F-measures of compared methods for general-fiction domain.

	0	25	50	100	200	340
SRC-only	72.46	-	-	-	-	-
TGT-only	-	61.42	66.02	69.88	71.27	73.15
PRED	-	70.30	70.03	73.30	74.68	77.24
FA	-	69.76	70.79	71.88	73.06	75.38
S-SVM.Prior	-	75.39	75.55	75.50	76.69	78.53

The result shows how two baselines (SRC-only, TGT-only), two domain-adaptation algorithms (FA, PRED), and S-SVM.Prior perform in the SRL task as the training data size varies. The SRC-only baseline achieves 64.09%, which corresponds to a performance drop of near 20% from the WSJ source domain results. The TGT-only baseline performs poorly in the beginning but improves quickly as the number of target-domain training data (on the first row of the table) increases. Our proposed method, S-SVM.Prior, performs better than other domain-adaptation algorithms and the two baselines.

The result from the experiment using the general-fiction domain as target is shown in Table 3 and Fig. 3(b). The result of the general-fiction domain is similar to that of the biomedical domain. The SRC-only baseline achieves 72.46%, which is 10.75% lower than the source domain performance. The TGT-only baseline does not reach comparable results with domain-adaptation algorithms despite all training data being added. Our proposed method shows best performance in this experimentation, too.

The result of the first experimentation shows that our proposed algorithm for domain adaptation is the best on both target domains. Our method also achieves best performance for every training-data size.

Table 4. Verb classification with usage.

	Yes	No
Is the usage different in the source and target domains?	Activate, bind, encode, express, interact, modulate, mutate, phosphorylate, promote, transactivate	Affect, alter, associate, block, decrease, differentiate, enhance, increase, induce, inhibit, mediate, prevent, reduce, regulate, repress, signal, stimulate, suppress, transform, trigger

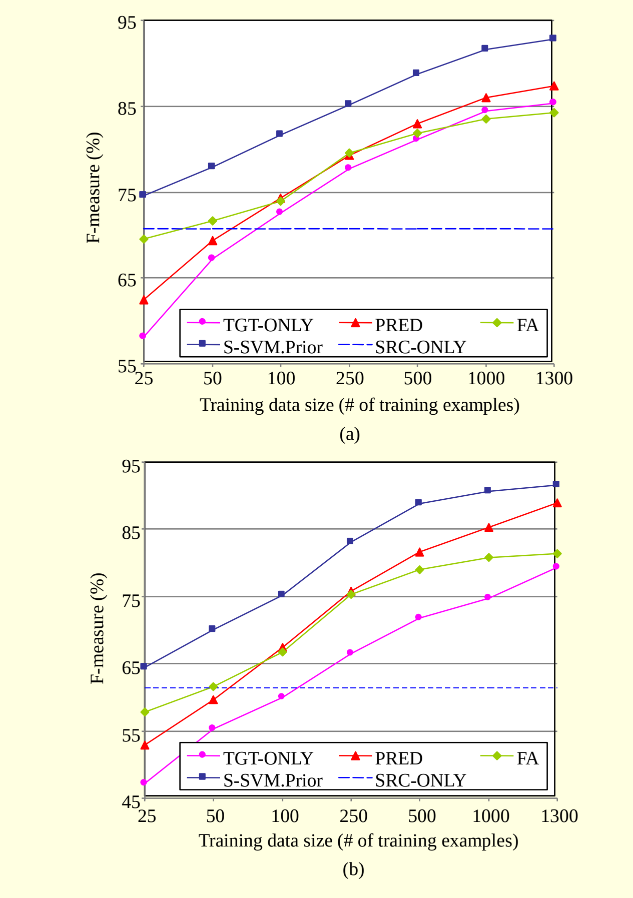


Fig. 4. Performance data according to usage: (a) same usage and (b) different usage.

The second experimentation examines how our proposed domain-adaptation method performs when there are a lot of variations in difficulty in the SRL task. Difficulty in SRL increases when usages (meanings) of predicates are different between the source and target domains. For example, if there is a change of usage of a predicate when the domain is switched from the source to the target, it is hard for the systems to achieve correct SRL for the predicate.

Consider the following two examples for the predicate *increase* [21]:

Source domain: [Sales]$_{A1}$ *increased* a more modest [4.8%]$_{A2}$ in the [South]$_{AM\text{-}LOC}$

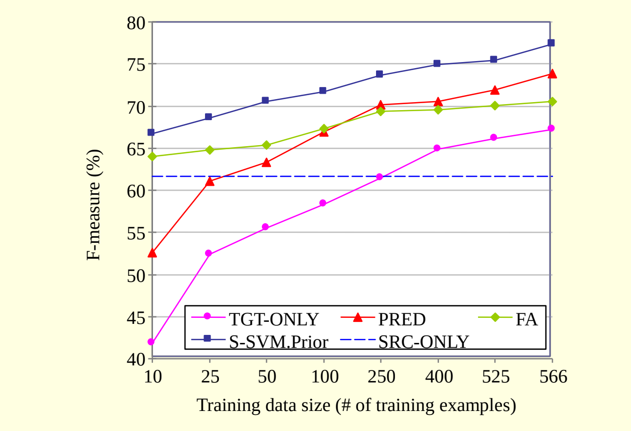


Fig. 5. Performance measured for a target domain in German.

Target domain: [LTB4]$_{A0}$ *increased* the expression of the c-fos [gene]$_{A1}$ in a time- and concentration-dependent [manner]$_{AM\text{-}MNR}$

In the example, "*increased*" in the source domain has an intransitive usage, and "Sales" is A1 (thing increasing). This usage can typically be found in the source domain. In contrast, "*increased*" in the target domain is a transitive verb, and "LTB4" is A0. Predicates with different usage in the source and target domains can cause difficulty for domain adaptation.

To quantify the difficulty caused by usage difference, we split the test data of the target biomedical domain into two sets: a set (labeled "same usage") containing the predicates whose usage is the same in the source and target domains, and the other set (labeled "different usage") with the predicates whose usage in the source domain is different from that in the target domain. To have this categorization of predicates, we refer to the data provided in [22]; the result of which is given in Table 4.

We have tested our proposed DA method using the data sets resulting from splitting. The results are shown in Fig. 4. Performance results for "SRC-only" indicate that SRL for "different usage" is more difficult than that for "same usage." In the case of "same usage," methods other than our own show similar performance; while TGT-only is far worse than others in "different usage." However, what is most notable is that our method is superior to all others regardless of usage and data size. When the training data size gets large, our method's performance reaches almost the same high value in both usage cases. This observation suggests our DA method is effective even in difficult SRL cases. The third experimentation examines the performance of our proposed method against another language (German) in another target domain (legislation). The third experimentation has been carried out on German using the same experimental setup as before. The result of the third experimentation is shown in Fig. 5. As it can be seen in the result, our proposed algorithm also gives the best performance.

All three experimentations explained so far, indicate that our domain-adaptation technique for SRL proposed in this paper is effective compared to the previous other methods.

VI. Conclusion

In this paper, we propose a new domain-adaptation technique for semantic role labeling systems that is based on structural SVMs to perform SRL and exploits a *prior* model to achieve domain adaptation. We show, by several experimentations, that a state-of-the-art multi-domain SRL system can be developed by utilizing our proposed method. In particular, we introduce a training procedure for a structural SVM that adapts the source-domain SVM to a new target domain. It is demonstrated in experimentations that our proposed domain-adaptation method is superior to other methods for the three different target domains used. Furthermore, our proposed domain-adaptation method shows high performance on various splits of target-domain data by usage difference of predicates between the source and target domains.

References

[1] M. Surdeanu et al., “Using Predicate-Argument Structures for Information Extraction,” *Proc. ACL*, vol. 1, July 2003, pp. 8–15.

[2] H-J. Oh, C.K. Lee, and C-H. Lee, “Analysis of the Empirical Effects of Contextual Matching Advertising for Online News,” *ETRI J.*, vol. 34, no. 2, Apr. 2012, pp. 292–295.

[3] X. Carreras and L. Marquez, “Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling,” *Proc. CoNLL*, Ann Arbor, Michigan, USA, June 30, 2005, pp. 152–154.

[4] S. Pradhan, W. Ward, and J. Martin, “Towards Robust Semantic Role Labeling,” *Computational Linguistics*, vol. 34, no. 2, June 2008, pp. 289–310.

[5] M. Surdeanu et al., “The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies,” *Proc. CoNLL*, Manchester, UK, Aug. 2008, pp. 159–177.

[6] J. Hajic et al., “The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages,” *Proc. CoNLL*, Boulder, CO, USA, June 2009, pp. 1–18.

[7] S. Lim, C. Lee, and D. Ra, “Dependency-Based Semantic Role Labeling Using Sequence Labeling with a Structural SVM,” *Pattern Recogn. Lett.*, vol. 34, no. 6, Apr. 2013, pp. 696–702.

[8] C. Chelba and A. Acero, “Adaptation of Maximum Entropy Capitalizer: Little Data Can Help a Lot,” *Comput. Speech Language*, vol. 20, no. 4, Oct. 2006, pp. 382–399.

[9] C. Lee and M. Jang, “A Prior Model of Structural SVMs for Domain Adaptation,” *ETRI J.*, vol. 33, no. 5, Oct. 2011, pp. 712–719.

[10] H. Daumé, “Frustratingly Easy Domain Adaptation,” *Proc. ACL*, Prague, Czech, June 2007, pp. 256–263.

[11] J. Jiang and C. Zhai, “Instance Weighting for Domain Adaptation in NLP,” *Proc. ACL*, Prague, Czech, June 2007, pp. 264–271.

[12] J.R. Finkel and C.D. Manning, “Hierarchical Bayesian Domain Adaptation,” *Proc. NAACL*, Boulder, CO, USA, May 2009, pp. 602–610.

[13] J. Blitzer, R. McDonald, and F. Pereira, “Domain Adaptation with Structural Correspondence Learning,” *Proc. EMNLP*, Sydney, Australia, July 22–23, 2006, pp. 120–128.

[14] F. Huang and A. Yates, “Distributional Representations for Handling Sparsity in Supervised Sequence-Labeling,” *Proc. IJCNLP AFNLP*, Singapore, vol. 1, Aug. 2–7, 2009, pp. 495–503.

[15] H. Daumé and D. Marcu, “Domain Adaptation for Statistical Classifiers,” *J. Artif. Intell. Res.*, vol. 26, no. 1, May 2006, pp. 101–126.

[16] C. Lee and M. Jang, “A Modified Fixed-Threshold SMO for 1-Slack Structural SVMs,” *ETRI J.*, vol. 32, no. 1, Feb. 2010, pp. 120–128.

[17] C. Cortes and V. Vapnik, “Support-Vector Networks,” *Mach. Learning*, vol. 20, no. 3, Sept. 1995, pp. 273–297.

[18] S. Shalev-Shwartz et al., “Pegasos: Primal Estimated Sub-Gradient Solver for SVM,” *Proc. ICML*, Corvallis, Oregon, USA, June 20–24, 2007, pp. 807–814.

[19] I. Tsochantaridis et al., “Support Vector Machine Learning for Interdependent and Structured Output Space,” *Proc. ICML*, July 2004.

[20] J. Kim et al., “GENIA Corpus-a Semantically Annotated Corpus for Bio-textmining,” *Bioinformat.*, vol. 19, no. 1, July 2003, pp. i180–i182.

[21] D. Dahlmeier and H.T. Ng, “Domain Adaptation for Semantic Role Labeling in the Biomedical Domain,” *Bioinformat.*, vol. 26, no. 8, Apr. 2010, pp. 1098–1104.

[22] R. Tsai et al., “BIOSMILE: A Semantic Role Labeling System for Biomedical Verbs Using a Maximum-Entropy Model with Automatically Generated Template Features,” *BMC Bioinformat.*, vol. 8, no. 325, Sept. 2007.

Soojong Lim received the BS in mathematics from Yonsei University, Seoul, Rep. of Korea, in 1997. He received his MS in computer science from Yonsei University, Seoul, Rep. of Korea in 1998. He received the PhD degree in computer science from Yonsei University, Seoul, Rep. of Korea, in 2014. Currently he is a principal researcher in Electronics and Telecommunications Research Institute (ETRI), Daejeon, Rep. of Korea. His research interests include natural language processing, machine learning and question answering.

Changki Lee received his BS in computer science from KAIST, Daejeon, Rep. of Korea, in 1999. He received his MS and PhD in computer engineering from POSTECH, Pohang, Rep. of Korea, in 2001 and 2004, respectively. From 2004 to 2012, he was a researcher at the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Rep. of Korea. Since 2012, he has been with Kangwon National University, Rep. of Korea, as an assistant professor. He has served as a reviewer for international journals such as Information System, Information Processing & Management, and ETRI. His research interests are natural-language processing, information retrieval, data mining, and machine learning.

Pum-Mo Ryu received his BS degree in computer engineering from Kyungpook National University, Daegu, Rep. of Korea, in 1995, and his MS degree in computer engineering from POSTECH, Pohang, Rep. of Korea, in 1997. He received his PhD degree in computer science from KAIST, Daejeon, Rep. of Korea, in 2009. Currently he is a senior researcher at the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Rep. of Korea. His research interests include natural-language processing, text mining, knowledge engineering, and question answering.

Hyunki Kim is a director of the Knowledge Mining Research Section at the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Rep. of Korea. He received his BS and MS degrees in computer science from Chunbuk National University, Jeonju, Rep. of Korea, in 1994 and 1996, respectively. He received his PhD in computer engineering from the University of Florida, Gainesville, USA, in 2005. His research interests include natural-language processing, machine learning, question answering, and social big-data analytics.

Sang Kyu Park received his BS degree in computer engineering from Seoul National University, Seoul, Rep. of Korea, in 1982. He received his MS and PhD degrees in computer science from Korea Advanced Institute of Science and Technology, Daejeon, Rep. of Korea, in 1984 and 1998, respectively. Currently he is in charge of the Automatic Speech Translation and Artificial Intelligence Research Center, ETRI, Daejeon, Rep. of Korea. His research interests include automatic speech translation, natural-language processing, speech recognition, knowledge mining, and question answering.

Dongyul Ra received his BS in electronics engineering from Seoul National University, Seoul, Rep. of Korea, in 1978. He received his MS and PhD in computer science from KAIST, Daejeon, Rep. of Korea, in 1980, and Michigan State University, USA, in 1989, respectively. He has been a faculty member of Yonsei University since 1991. His research interests include natural-language processing, artificial intelligence, and information retrieval.