Logic Knowledge Base Refinement Using Unlabeled or Limited Labeled Data

CHAN, Ki Cecia

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Systems Engineering and Engineering Management

The Chinese University of Hong Kong
August 2010

Thesis/Assessment Committee

Professor Yu Xu, Jeffery (Chair)
Professor Lam Wai (Thesis Supervisor)
Professor Lam Kai Pui (Committee Member)

Abstract

In many text mining applications, knowledge bases incorporating expert knowledge are beneficial for intelligent decision making. Refining an existing knowledge base from a source domain to a different target domain solving the same task would greatly reduce the effort required for preparing labeled training data in constructing a new knowledge base. We investigate a new framework of refining a kind of logic knowledge base known as Markov Logic Networks (MLN). One characteristic of this adaptation problem is that since the data distributions of the two domains are different, there should be different tailor-made MLNs for each domain. On the other hand, the two knowledge bases should share certain amount of similarities due to the same goal. We investigate the refinement in two situations, namely, using unlabeled target domain data, and using limited amount of labeled target domain data.

When there is no manual label given for the target domain data, we refine an existing MLN via two components. The first component is the logic formula weight adaptation that jointly maximizes the likelihood of the observations of the target domain unlabeled data and considers the differences between the two domains. Two approaches are designed to capture the differences between the two domains. One approach is to analyze the distribution divergence between the two domains and the other approach is to incorporate a penalized degree of difference. The second component is logic formula refinement where logic formulae specific to the target domain are discovered to further capture the characteristics of the target domain.

When manual annotation of a limited amount of target domain data is possible, we exploit how to actively select the data for annotation and develop two active learning approaches. The first approach is a pool-based active learning approach taking into account of the differences between the source and the target domains. A theoretical analysis on the sampling bound of the approach is conducted to demonstrate that informative data can be actively selected. The second approach is an error-driven approach that is designed to provide estimated labels for the target domain and hence the quality of the logic formulae captured for the target domain can be improved. An error analysis on the cluster-based active learning approach is presented. We have conducted extensive experiments on two different text mining tasks, namely, pronoun resolution and segmentation of citation records, showing consistent improvements in both situations of using unlabeled target domain data, and with a limited amount of labeled target domain data.

摘要

在眾多的文本挖掘的應用，將專家知識納入知識庫有利於智能決策。優化來源域(source domain)的知識庫到目標域(target domain)解決相同的任務，將大大減少所需的準備工作。我們研究了一個新的框架以優化一種被稱爲馬爾可夫邏輯網絡(Markov Logic Network (MLN))的邏輯知識庫。這問題的一個特點是由於數據分佈在兩個域的不同，而應該有不同的並度身訂做的邏輯網絡。另一方面，這兩個知識庫應該有某度上的相似之處。我們調查在兩種情況下，分別爲使用沒有標籤的目標域數據，或利有有限數量的已標籤的目標域數據，以作優化。

當目標域沒有已標籤的數據時，我們通過兩個組成部分完善已有的邏輯網絡。第一部分是邏輯公式的重量優化，我們同時考量目標域的觀測 數據的最大化及這兩域之間的分別。一種方法是分析兩個域之間數據分佈的不同，而另一種方法則是加入處罰。第二部分是邏輯公式的優化，此部分的目的爲發掘目標域的邏輯公式，以進一步捕捉目標域的特徵。

當標籤有限數量的目標域數據是可能時，我們設計出兩個主動學習的方法來用積極選擇要標籤的數據；第一種方法是基於集合的主動學習法， 它主要考來源和目標的區域的不同，而我們提出理論分析以立證明我們的方法能積極選擇有用數據作爲取樣；第二種方法是錯誤驅動法，旨在爲目標堿提供估計標籤， 而使目標區域的邏輯公式的質量得到改善。 我們在兩個不同的文本挖掘的任務即代名詞指代問題和引用記錄切分問題，進行了的實驗都顯示出持續的改善。

Acknowledgments

First, I would like to express my most sincere gratitude to my supervisor and mentor, Dr. Wai Lam. His patience and advice are the most valuable supports to help me get through all the challenges that I faced in study and in research. I am deeply in debt to Dr. Lam, as he has devoted so much time and efforts in giving me precious advices and teaching me. Without him, I would never have been able to accomplish this research and this thesis.

I wish to express my warmest gratitude to all the professors, staffs and my fellow schoolmates from the Department of Systems Engineering and Engineering Management for their help and support. My special thanks go to Dr. Gatien Wong who has helped me so much.

Moreover, I would like to thank my dearest friends, Christine, Yanny, Kisha, Mr. and Mrs. Leung for encouraging me when I am frustrated, and cheering me up when I am under great pressure. They are always there for me. I am the luckiest person in this world to have so much help and support from the ones I love and care. I am really grateful to all of them.

Above all, I would like to give my special thanks to my parents for their love and support. To me, they are the most understanding parents in the world. They are always there by my side, loving me and giving me the strength to come through all the good times and bad times in my life. They have also given me the courage to make my own plans and pursue my own goals.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In many information systems, different information processing components are required for building intelligent applications. Knowledge bases are particularly useful in aiding decision making as expert knowledge can be flexibly captured and utilized. However, we often encounter situations where we already have an existing knowledge base from a source domain and we wish to apply it to solve the same task in a target domain which is different from the source domain. This situation is particularly common to many text mining problems. Extensive analysis and efforts are carried out to build the knowledge base for solving the problem in a particular domain. To date, we are facing an overwhelming stream of information in this fast-changing world. New information appears raising the need for a new knowledge base to tackle the new and unseen domains. For text mining, domains may refer to text documents from different information sources, different topical categories, or different registers in linguistics. For example, one may have labeled documents from the Wall Street Journal where tokens in sentences are annotated with their corresponding part-of-speech information, but the actual goal is to develop a knowledge base for performing the task of part-of-speech tagging for biomedical texts. The text documents from the Wall Street Journal and the biomedical texts are referred to as the source and the target domains respectively. Due to the difference in the distributions between the two domains, the existing knowledge base for the source domain may not be adequate for the target domain. Typically, direct application of the source knowledge base to the target domain would result in large degradation in performance. Even when the text mining tasks of the source and target domains remain the same, there exists considerable differences in them, such as feature space, hindering effective direct application of the source model to the target domain. In some situations, the distribution of data even in the same domain may change over time. The learned model may no longer be adequate for the updated data.

One solution is to acquire expert knowledge for the target domain to manually refine the knowledge base. Alternatively, another solution is to collect sufficient amount of labeled data via manual annotations in the target domain so that the knowledge base can be automatically discovered. Labeled data refers to pieces of information containing the answers or labels provided by experts to certain data in the domain. A model can be constructed using these labeled data for solving the task related to the domain. Such model can then be used to aid the prediction of the answers to data given some observations. But additional expert knowledge is expensive to acquire and manual annotations for sufficient data in the target domain may be costly or even infeasible. Hence, a useful approach would be refining the existing available source domain knowledge base to the target domain using unlabeled target domain data or a very small amount of labeled target domain data.

We investigate the refinement of a kind of logic knowledge base known as Markov Logic Networks (MLN) [50]. A standard MLN is a combination of probabilistic and first-order logic graphical models. It consists of a first-order knowledge base which is a set of first-order logic formulae describing the logic relations of the elements in the task and a set of weights, in which a weight is associated with each formula. The representation of first-order logic enables flexible model construction capturing knowledge such as relations among entities. It has the ability to incorporate a wide range of domain knowledge. Many statistical relational learning tasks, such as collective classification, link prediction, and object identification, can be concisely represented in MLN. An MLN serves as a promising tool for information extraction and decision making. Similar to other statistical approaches, the accommodation of an existing MLN model to another domain would ease the burden of manual annotation.

This motivates our investigation of performing refinement for an MLN. Suppose we need to solve a particular task, such as part-of-speech tagging, typically an existing source domain MLN suitable for problem solving of the task in the source domain is available. Now we wish to refine it so that it is suitable for the target domain. However, manual annotation of sufficient labeled data of the target domain is not available for learning a new model specifically for the target domain. Our aim is to reduce the amount of labeled data required while improving the performance of the refined model. Hence, we investigate how to effectively and automatically refine an existing MLN, constructed for the source domain to a target domain under two situations, namely, the situation of using unlabeled target domain data and the situation of using a limited amount of labeled target domain data. In the first situation, we assume manual annotation of the target domain data is infeasible, only unlabeled data from the target domain is given. The second situation is that limited resources for manual annotation is available such that a very limited amount of labeled target domain data can be acquired through expert annotation.

1.1 Our Framework

We first investigate the situation of using unlabeled target domain data. When labeled data from the target domain is not available, we refine the existing model in two components. The first component exploit the unlabeled data by jointly maximizing the likelihood of the observations of the target domain and considering the difference between the source and the target domains. The rationale is that although the source and target domain datasets may have different data distributions, they also share certain similarities since they solve the same task. Hence, it can be observed that the distribution of the target domain will not deviate too far from the the source domain. We develop two approaches to capturing the differences between the domains. In the first approach, it minimizes the distribution divergence between the two domains. For the second approach, we incorporate a penalized degree of difference, such that the influence of differences between the two domains to predictions are considered. The second component for exploiting the unlabeled data is to discover new logic formulae for the target domain using dependency information on the unlabeled target domain data only. In the absence of labeled data in the target domain, it is quite difficult to directly discover target domain specific logic formulae. The rationale of our approach is that although the new logic formulae for the target domain are not applicable in the source domain, they possess some similar behaviors with certain source domain logic formulae. We extract potential logic formulae from the unlabeled data of the target domain and analyze their correlations with the source domain model. Then we construct the new logic formulae for the target domain.

Next, we investigate the situation of using a limited amount of labeled target domain data. When a small amount of manual annotations can be obtained from the target domain, we exploit such limited resources at the largest advantage. We develop active learning methods by actively asking the expert to provide labels (or answers) of a very small amount of automatically selected data. Two methods of incorporating active learning into knowledge base refinement for the target domain are proposed, namely, pool-based and error-driven methods. In the pool-based method, we investigate how to actively select target domain data such that the labels obtained can aid the process of model refinement. Our rationale is that logic formulae specific to the target domain represent different characteristics of the source domain and the target domain. The impact of the new logic formulae can serve as important clues for refining the predictions and hence for better estimation of the target domain's distribution. The selection of data is achieved by analyzing the influence of potential logic formulae in the target domain data. The actively labeled data is then utilized to filter potential relations and for adapting the model to the target domain.

In the error-driven method, since the size of labeled data is also an important factor for learning, limited labeled target domain data still inhabits the actual analysis of the target domain. Hence, this method estimates complete labels for the target domain data by annotating a small number of actively selected target domain data. The estimated labels are then used for evaluating the error of the potential logic formulae. By identifying groups of data with similar characteristics, we can capture potential relations from the target domain. Such clustering of data can be obtained with the help of actively selected data. Potential logic formulae in the unlabeled target domain data are discovered using the estimated labels.

1.2 Our Contribution

We investigate techniques for employing first-order knowledge base, specifically Markov Logic Networks (MLN), to solve text mining problems. As informative relational information can be obtained from textual documents [5], text mining problems can be flexibly modeled in MLN [9]. We develop a relational knowledge base for solving pronoun resolution [9, 6]

We develop a new framework for MLN knowledge base refinement from a source domain to a target domain in two situations. For the situation where only unlabeled data can be obtained for the target domain, we propose two different MLN refinement approaches, namely, distribution divergence approach [10] and penalty-based approach [12], by exploiting the differences between the source and the target domains to tailor the model to the target domain data. The source domain MLN is refined to conform to the characteristics of the target domain achieving improved performance on the target domain data. We also propose methods [11, 7] to discovering logic formulae for the target domain using unlabeled target domain data. Logic formulae not found in the source domain knowledge base but beneficial to the target domain are discovered to capture the target domain specific logic formulae.

For the second situation where only limited labeled data can be annotated, we investigate the refinement process by actively selecting data for labeling to tackle the tasks of logic formula discovery and knowledge base refinement [8]. Two methods, namely, pool-based method and error-driven method, are designed to analyze how the limited amount of labeled data can be exploited for the knowledge base refinement. The pool-based method is developed for integrating the selection of data and logic formulae to improve the performance for the target domain. In the error-driven method, we demonstrate that informative relations can be selected by examining the predicted labels obtained from actively clustering of data and examples.

We also conduct a theoretical analysis on the assignment error bound.

It shows that the pool-based approach can generate useful data for evaluating the changes of the target domain knowledge base and improving the performance of the target domain knowledge base over the source domain knowledge base. For the cluster-based active learning approach, an analysis on the error of the estimated labels is presented.

We have implemented our framework and conducted experiments on two text mining applications, namely, pronoun resolution and segmentation of citations. Pronoun resolution is to determine a pronoun's antecedent among a set of noun phrases appeared in text documents. Segmentation of citations is to identify the fields of a technical paper citation for extracting the bibliographical records. Our experimental results on the two applications demonstrate that our proposed framework is able to improve the performance on the target domain in both the situation of having only the unlabeled target domain data or having only a very limited amount of labeled target domain data. The knowledge base is refined so that new formulae useful to the target domain can be automatically discovered and weights for the formulae are learned.

1.3 Thesis Organization

The organization of this thesis will be as follows:

Chapter 1: Introduction. This chapter provides an introduction to the work.

Chapter 2: Literature Reviews. This chapter describes some previous works regarding the area of domain adaptation, MLN transfer learning and Active Learning.

Chapter 3: Background. In this chapter, we provide details on Markov Logic Networks and a problem definition of knowledge base refinement.

Chapter 4: Using MLN to Tackle Text Mining. This chapter investigates the construction of an MLN for pronoun resolution, a text mining task.

Chapter 5: Knowledge Base Refinement with Unlabeled Target Domain Data. In this chapter, we present our approaches of knowledge base refinement using unlabeled target domain data. Experimental results are presented.

Chapter 6: Knowledge Base Refinement Using Limited Labeled Target Domain Data. We describe our approaches of incorporating active learning with knowledge base refinement and the experiments carried for the approaches.

Chapter 7: Conclusions. This chapter presents the conclusions and discusses some future directions.

Chapter 2

Literature Review

2.1 Domain Adaptation

One related area is domain adaptation which has been applied for other machine learning methods. For statistical learning, Daumé III [21, 20] has proposed two models of domain adaptation on analyzing general domain and specific domain distributions and applied them on a series of sequence labeling tasks, such as named entity recognition, shallow parsing, and part-of-speech tagging. Many of the investigations focused on the fully supervised scenario with both labeled data from the source and the target domains [15, 26]. More recently, domain adaptation models using unlabeled data from the target domain have been investigated. Some works have attempted to learn a new representation for bridging the source and the target domain. Blitzer et al. [3] proposed a method called structural correspondence learning (SCL) to learn a common feature representation for feature-based discriminative model by selecting some domain independent pivot features. Raina et al. [49] learned the sparse basis from unlabeled data which does not necessarily come from the same domain as the labeled data. Other works try to evaluate the difference in distributions between two domains by a non-parametric distance estimate. Pan et al. [44] applied the Kernel Maximum Mean Discrepancy to learn the embedded space where the distance between distributions of the source and the target domain is minimized. Guo et al. [27] developed a model using latent semantic association to overcome the distribution gap between domains. Another research direction for domain adaptation is instance weight assignment. Jiang and Zhai [30] proposed an instance weighting technique for domain adaptation in NLP. Recently, Zhong et al. [60] utilized the Kernel Discriminative Analysis (KDA) to seek a common feature space which makes the marginal distributions from two domains close and then re-selects and re-weighs source domain examples to remove the bias of the mapping. However, most of these works assume that the conditional distribution of the label values given a data instance is unchanged between the source and target domains.

2.2 Transfer Learning in Markov Logic Networks

Traditional MLN structure learning methods aim at constructing logic formulae of MLN with labeled training data. For example, Kok and Domingos [32] first introduced a probabilistic method for learning MLN structure which outperforms previous inductive logic programming (ILP) methods. Mihalkova and Mooney [40] have proposed a bottom-up approach to addressing the problem of local maxima in previous top-down structure learning approaches. More recently, Kok and Domingos [33] presented an approach which directly utilizes the data for constructing candidate clauses by considering the relational database as a hypergraph.

Although the above works of structure learning can refine an existing MLN structure, considerable amount of labeled data on the target domain has to be provided. In contrast, our proposed framework adapts an existing MLN model and revises it for the target domain. However, little investigations have been carried on such domain adaptation in MLN. Instead, there is a related area known as transfer learning in MLN. Transfer learning in MLN focuses on mapping a knowledge base learned from one task to a different task, where the predicates and variables are different. For example, TAMAR, proposed by Mihalkova et al. [40], treats the transfer process as two subtasks. It first identifies the best mapping for the formula and then it revises the formula to improve its fit to the target domain data. Another model, called DTM proposed by Davis and Domingos [22], performs deep transfer based on a form of second-order Makov Logic. Given a set of first-order formulae in the source domain, they lift them into second-order logic by replacing the predicate names with variables and are then instantiated by the predicate names in the target domain. Mikhalkova and Mooney [41] have proposed another model, called SR2LR, which introduces the single-entity-centered setting to transfer a source domain MLN to a target domain. One common characteristic for the above models is that their problem setting is to adapt a source domain MLN to the target domain solving different tasks. The sets of predicates defined for the source domain and the target domain have to be different. As a result, these models are not effective when applying to our problem setting of solving the same task but different data distributions in both domains. They find effective mapping between the set of predicates from the source domain to the different set of predicates from the target domain. They focus on identifying the best formula for the target domain among different candidate formulae generated from the mappings of predicates. Though our problem setting is for solving the same task in both domains, we not only refine the formulae for the target domain but also dicover new formulae for the target domain whereas they do not discover new formulae for the target domain.

2.3 Active Learning

For many machine learning problems, labeled data are costly to prepare. Active learning provides a way to reduce the quantity of labeled training examples in achieving better performance. In active learning, queries, which are unlabeled examples, are selected automatically to acquire an expert's knowledge in labeling it. Active learning has been extensively studied in many machine learning problems and in different settings such as synthesizing queries [31], selective sampling [17], or pool-based active learning [37]. In query synthesis, queries can be generated by the learner to request for labeling, while in selective sampling, queries are processed one at a time to decide if it is processed for labeling. Pool-based active learning is motivated by the situation that in many real-world learning problems, such as text classification [57, 28] and processing [13], image classification and retrieval [56, 59], large collections of unlabeled documents are usually available. Hence, it is one of the most commonly employed active learning setting for different learning problems. One issue to be addressed in pool-based active learning is how to select queries from a pool of unlabeled data for users or experts to label and to reduce the number of labeled queries required for learning.

Different strategies for evaluating and ranking the unlabeled examples for querying are investigated. The simple and the most commonly employed one is uncertainty sampling. Lewis and Gale [37] presented an uncertainty sampling approach which selects examples with unclear class membership in an iterative manner. Entropy [52] and confidence measures are also employed in uncertainty sampling for different models such as conditional random fields [35] and support vector machines [18]. Query-by-committee(QBC) algorithms [53, 39] are alternatives to uncertainly sampling. It maintains a committee of models representing competing hypotheses. Queries are selected based on the level of disagreement between the committee members. Roy and Macallum [51] presented a querying strategy using estimated error reduction. Examples are selected according to the reduced error rate using naive Bayes models. Beygelzimer [2] recently proposed an importance weighted active learning framework based on general loss function to correct sampling bias.

Moreover, research works on combining active learning with other related research areas have been carried out. For example, some research works have exploited the use of clustering with active learning [43]. Others investigated the use of active learning in feature level [47].

Though active learning has been studied in a range of learning problems and applications, little attention has been paid to combine active learning with domain adaptation. Chan and Ng [13] proposed an approach to using active learning with domain adaptation for word sense disambiguation task. They employed a pool-based active learning approach of selecting examples for labeling and for learning naive bayes model to perform word sense disambiguation. Shi et al. [54] have also presented an approach of combining transfer learning with active learning. They designed an ensemble of classifiers learned with labeled source domain data and target domain data separately. However, different from our approach, their approach requires an initial pool of labeled target domain data for the construction of the target domain classifier. Only with a learned source domain classifier and a learned target domain classifier, the approach performs active learning to acquire labels for queries. More recently, Rai et al. [48] proposed an active learning model in a selective sampling setting to learn the best possible domain separator hypothesis to rule out certain examples for labeling. Our approach differs with the approach in that our approach utilizes active learning for not only acquiring informative examples for constructing labeled target domain data but also selecting informative relations for the target domain.

Chapter 3

Background and Problem Definition

3.1 Markov Logic Networks

Many text mining and natural language processing applications involve relational information. Representing such information using logic is considered as an attractive model. It enables effective incorporation of complex knowledge. Recently, research on combining first-order logic representation with probabilistic models has shown that superior results over pure probabilistic models are obtained. A successful model known as Markov Logic Networks (MLN) [50], proposed by Richardson and Domingos, has been applied to different text mining tasks [45]. MLN is a combination of probability and

Table 3.1: An example of a logic formula and its weight in MLN

Formula	Weight
number(m, plural) ∧ number(p, plural) ∧ isAntecedent(m,p)	1.39

first-order logic graphical models. It aims at representing the knowledge in first-order logic together with a probabilistic model for handling uncertainty. The representation of first-order logic enables flexible model construction involving relations between entities. It is composed of a knowledge base containing a set of first-order formulae and each formula is associated with a weight. Considering a text mining problem, namely pronoun resolution, whose objective is to identify the antecedents of pronouns in a text document, we can design some first-order predicates, for examples, $number(p, n)$, and $isAntecendent(m, p)$. The predicate $number(p, n)$ indicates whether the pronoun p is of number type n, where $n \in \{singula, plural, unknown\}$. The predicate $isAntecendent(m, p)$ indicates whether the mention m is an antecedent of the pronoun p. We can construct a logic formula using the predicates. Table 3.1 depicts an example of a logic formula in MLN.

Given an MLN and a set of constants, a ground Markov network can be obtained by applying the formulae to the set of constants, i.e., grounding of formulae. An example of constants for the pronoun p is "them" and

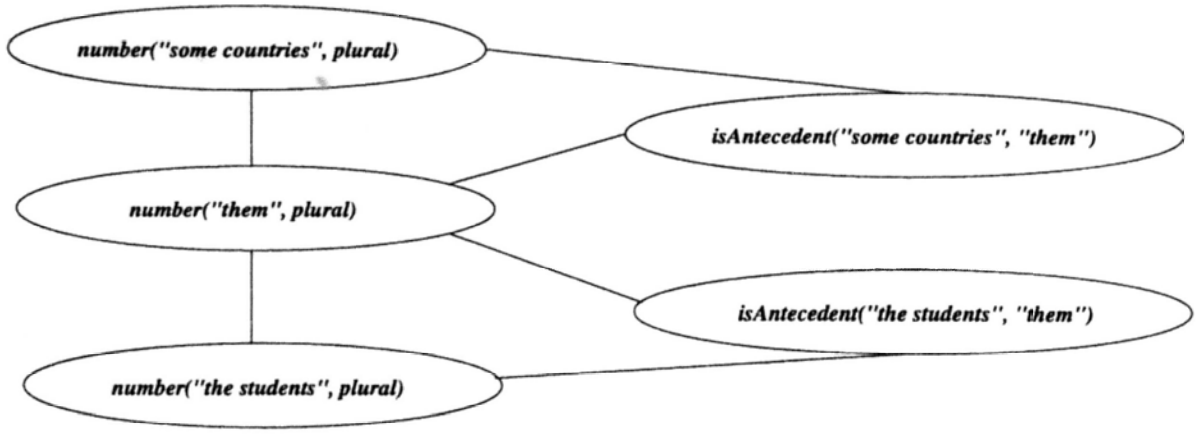
number("some countries", plural)
isAntecedent("some countries", "them")
number("them", plural)
isAntecedent("the students", "them")
number("the students", plural)

Figure 3.1: An example of a ground Markov network.

an example for the mention m is "the students". By applying the constants to the predicates of the formula, groundings of candidates, such as $number(\text{"}them\text{"}, plural)$, $isAntecedent(\text{"}thestudents\text{"}, \text{"}them\text{"})$, can be obtained. Of the ground Markov network constructed, a node corresponds to a grounding of the predicates specified in the formulae and there is a truth value associated with each node. Two nodes are connected by an edge if their corresponding ground predicates appear together in the same formula. An example of a ground Markov network is illustrated in Figure 3.1.

The probability distribution of a ground Markov network, X, over a possible world, x, can be expressed as follows:

$$P(X = x) = \frac{1}{Z}exp(\sum_{i}^{F} w_i n_i(x)) = \frac{1}{Z}\prod \phi_i(x_i)^{n_i(x)} \tag{3.1}$$

where $P(X)$ refers to the probability distribution over all possible worlds x, the assignment of truth values; F represents the number of formulae in the

MLN; w_i refers to the weight for the i-th formula; $n_i(x)$ refers to the number of true groundings of a formula in the possible world x; x_i is the truth value of the atoms, the groundings of the predicates, appeared in the formula, and $\phi_i(x_i) = e^{w_i}$. Z refers to the normalizing factor which is the sum of the probabilities over all possible worlds. As a result, given the truth values for some nodes in the ground network, one can infer the truth values for other nodes. MLN weights w_i can be learned by maximizing the likelihood of the relational database.

Richardson and Domingos [50] suggested to use Gibbs sampling to perform inference over the ground Markov network. First, it samples one ground predicate X_l given its Markov blanket $MB(X_l)$. The Markov blanet of a ground predicate is the set of ground predicates that appear in some grounding of a formula with it. Then, given its Markov blanket, it calculates the probability that X_l takes on a particular truth value x_l using the formula:

$$P(X_l = x_l|MB(X_l)) = \frac{exp(\sum_{i=1}^{F} w_i n_i(x))}{exp(\sum_{i=1}^{F} w_i n_i(x_{[X_l=0]})) + exp(\sum_{i=1}^{F} w_i n_i(x_{[X_l=1]}))} \tag{3.2}$$

where $n_i(x)$ refers to the number of true groundings of the i-th formula in x. $n_i(x_{[X_l=0]})$ refers to the number of true groundings of the i-th formula when we force $X_l = 0$. $n_i(x_{[X_l=1]})$ refers to the number of true groundings of the

i-th formula when we force $X_l = 1$.

3.2 Problem Definition

3.2.1 Knowledge Base Construction and Learning

For solving a typical text mining problem, we can design two sets of first-order predicates, namely, evidential predicates denoted by E and query predicates denoted by Y. Evidential predicates refer to the predicates whose truth values can be determined from the observations, while query predicates refer to the predicates whose truth values are not known from the observations. For instance, in pronoun resolution, $number(p, n)$ is one of the evidential predicates while $isAntecendent(m, p)$ is a query predicate. Next we can construct a set of logic formulae F based on expert knowledge using the predicates to capture the relations between the evidential predicates and the query predicates for a domain D. With the set of formulae F, specially designed for the domain D, standard MLN learning aims at automatically learning the weight W for each of the logic formulae. Standard MLN weight learning is performed when a set of training examples L in the domain D, such that the truth values of the groundings for query predicates are known, is given. For example, we may have a set of documents in which the antecedents of the pronouns are manually annotated by human experts. MLN learning aims at automatically learning the weight W_s for each of the logic formulae in *MLN*. To achieve this, we used a more efficient alternative to Equation 3.1, that is to optimize the pseudo-likelihood of the training data x as follows [50]:

$$P_W(X = x) = \prod_{l=1}^{n} P_W(X_l = x_l | MB_X(X_l)) \tag{3.3}$$

where X_l refers to the l-th grounded predicate; x_l refers to the state(truth values) of X_l in the training data; $MB_X(X_l)$ refers to the state of the Markov blanket of X_l; and n is the total number of grounded predicates in the training data; the subscript W denotes that the probability is computed using the weight W. Since the objective function is convex, the optimal weights W^* can be obtained efficiently using the limited-memory BFGS algorithm [38].

As a result, an MLN, denoted by *MLN*, is constructed. The learned MLN can then be applied to the operational (testing) documents in the same domain D, that is to infer the truth values of the groundings of the query predicates given the truth values of the groundings of the evidential predicates.

3.2.2 Knowledge Base Refinement

One major limitation of existing MLN learning is that the learned MLN_s for the source domain D_s cannot be effectively applied to a target domain

D_t, from another information source, solving the same task. Following the example of pronoun resolution, the suitability of the existing source domain formulae and the bags of words may be different for the two domains. New formulae describing the new words from the target domain have to be discovered and weights for the formulae have to be modified. The refined MLN will be more suitable for the target domain. In principle, a new set of training examples with manually annotated labels are required to learn the new target domain MLN, denoted as MLN_t, for D_t.

The problem setting investigated is described as follows: Suppose we need to solve a particular task, typically an existing source domain MLN suitable for problem solving in the source domain is available. Now we wish to refine it so that it is suitable for the target domain. However, manual annotation of sufficient labeled data of the target domain is not available for learning a new model specifically for the target domain. Given an existing source domain MLN, MLN_s, for a source domain D_s, we aim at learning an MLN, denoted as MLN_t , tailored to the target domain D_t by modifying MLN_s. The refined MLN_t can differ from MLN_s in the weights (i.e., W_s and W_t) as well as the formulae (i.e., F_s and F_t). Our aim is to obtain an MLN_t that achieves better performance in D_t than direct application of MLN_s to D_t.

During the refinement, two situations are handled:

1. **Using unlabeled target domain data**: In this situation, we are given only the unlabeled data U_t from the target domain. Under this situation, we are given the source domain MLN, MLN_s and extensive amount of unlabeled target domain data U_t.

2. **Using a limited amount of labeled target domain data**: In this situation, a very limited amount of labeled target domain data L_t can be acquired from the unlabeled target domain data U_t. The limited amount of target domain data L_t is selected automatically and the truth values (annotations) of the query predicates to the data are acquired from experts. This limited amount of labeled target domain data L_t and the remaining unlabeled target domain data U_t' are used to refine the source domain MLN for the target domain.

Let X_l be the l-th ground predicate whose truth value x_l is known and Y_l be the l-th ground query predicate whose truth value y_l is unknown. X_l can be an evidential predicate or a query predicate. Labeled data, L, is a dataset consisting of ground predicates X_l such that all the truth values, x_l, for X_l are known. Unlabeled data, U, is a dataset consisting of ground predicates X_i and Y_j, where the truth values y_i of Y_j are unknown. Note that unlabeled target domain data, U_t', also refers to the dataset whose query predicates are not selected for annotations.

Our goal is to develop a framework to effectively and automatically adapting an existing relational logic model, specifically, an MLN, constructed for the source domain to a target domain. This problem can be divided into two sub-problems, namely formula weight adaptation and logic formula refinement. The formula weight adaptation sub-problem is to learning the weights W_t of the logic formulae for the target domain. With the insufficient amount of labeled data, the learning of weights for the target domain using standard weight learning methods become infeasible. Hence, weights have to be adapted for the target domain. The logic formula refinement sub-problem is to automatically discover logic formulae F_t specific to the target domain. Since logic formulae in the source domain model are designed specifically for the source domain, they may not be adequate to the target domain. The existing logic formulae F_s may fail to capture the characteristics of the target domain.

Chapter 4

Using MLN to Tackle Text Mining

Logic knowledge base is suitable for tackling some text mining problems. Informative relational information can be obtained from textual documents [5] and hence many text mining problems can be solved using MLN. One way to construct a knowledge base for MLN is to acquire expert knowledge for building the logic formulae. For example, MLN has been investigated for solving text segmentation problem [55], where expert knowledge is employed to develop logic formulae for resolving the segmentation problem as determining the field of a token in text. We investigate using MLN for the task of pronoun resolution [6, 9]. To our knowledge, our work is the first to use MLN for resolving pronoun resolution.

Pronoun resolution is different from coreference resolution on proper nouns where surface features, such as string comparison, are not as significant. Despite the fact that pronouns are lack of rich semantic information, they are crucial in maintaining the coherence of knowledge representation in text. We investigate how to effectively characterize the pronoun coreference resolution process through conducting inference upon a variety of conditions. The influence of different types of constraints are also investigated. With MLN, expert knowledge, such as, linguistic features or constraints as heuristic rules can be incorporated into pronoun resolution, with the benefits of handling uncertainties.

4.1 Problem Description

Pronoun resolution is to identify the antecedents of the pronouns in text. The task of pronoun resolution can be regarded as determining a pronoun's antecedent among a set of noun phrases, to which are referred as candidate mentions. In our MLN model, all pairs of pronoun and candidate mention within a document will be jointly resolved for their antecedents. This differs with other pronoun resolution systems, which only considered pairs of pronoun and noun phrase independently [23].

Moreover, our goal of this task is to handle all kinds of personal pronouns and noun phrases in a single resolution model. An antecedent of pronoun can be any noun phrases, such as person, thing, and even temporal expressions, while a pronoun can be referential or non-referential. Different from many existing works on this task, we handle both referential and non-referential pronouns in a single model.

Personal pronouns include subjective (e.g. "I", "she"), objective (e.g. "me", "her"), possessive (e.g. "mine", "my", "hers") and reflexive pronouns (e.g. "myself", "herself"). A pronoun may be referential or non-referential. A referential pronoun means that it refers to another noun phrase in text, where non-referential pronouns do not refer to any specific noun phrases and hence they do not have an antecedent. The pleonastic "it" is an example of non-referential pronouns.

4.2 Model Design

From the linguistic point of view, the distribution and location of different mentions within texts are governed by certain restrictions. In other words, through identifying whether mentions satisfy the constraints or not, the referential linkage can be deduced. Knowledge base can be constructed with these constraints and hence corresponds to a logic network for reasoning. Hence, pronoun resolution can be well described in first-order logic. Also, the use of Markov logic network can support the handling of uncertainties in pronoun resolution.

To construct the logic formulae describing the constraints of pronoun resolution, we design different types of predicates to capture information regarding the pronouns and the candidate mentions. First, we design a first-order predicate $isAntecendent(m_i, p_j)$ as the query predicate which represents that the noun phrase m_i is an antecedent of the pronoun p_j. If a pronoun is non-referential, it is treated as having a null antecedent, which is represented as $isAntecedent(null, p_j)$.

Though referential and non-referential pronouns have different characteristics, they are closely dependent. This dependency can be modeled using the following formula:

$$isAntecedent(m_i, p_j) \wedge \neg isNull(m_i) \Rightarrow \neg isAntecedent(null, p_j) \qquad (4.1)$$

As mentioned, pronoun resolution can be described as the inference result of certain constraints, hence we can use the following general formulation to construct the logic formulae.

$$\text{constraints on } m_i \;\&\; p_j \Rightarrow m_i \text{ is antecedent of } p_j \qquad (4.2)$$

Our model aims at establishing the relations between pronouns and their corresponding antecedents. Constraints are constructed using predicates for capturing the information about pronouns and candidate mentions, hence we refer them as evidential predicates. These evidential predicates describe the information for each pronoun, each candidate mention, and also information between pairs of pronoun and candidate mention. As a result, we design different types of predicates to capture different information regarding the pronouns and the mentions.

- Lexical Predicates: This type of predicates describes the string comparison information.
  - *same_str*(p_i, m_j) - whether the pronoun, p_i, and the candidate antecedent, m_j, are the same.
- Positional Predicates: Positional information provides a proximity distance value between the pronoun p_i and its candidate mention m_j. It is believed that the closer the candidate mention is to the pronoun, the more likely it is the antecedent of the pronoun.
  - *sent_sent*(p_i, m_j) - whether the pronoun and the candidate mention are within the same sentence in the text.
  - *prev_sent*(p_i, m_j) - whether the candidate mention is in the previous sentence relative to the pronoun.
  - *prev_two_sent*(p_i, m_j) - whether the candidate mention is located

in two sentences ahead of the pronoun.

- $within(p_i, m_j)$ - whether the pronoun is located within the candidate mention.

- $after(p_i, m_j)$ - whether the pronoun's position located behind that of candidate mention.

○ Semantic Predicates: For a mention to be the antecedent of a pronoun, they have to agree semantically in gender, types and number information. The following predicates are designed to capture the semantic information regarding the pronoun and the mention.

- $gender(m_i, t)$, $gender(p_i, t)$ - indicates whether m_i or p_i is of gender t, where $t \in$ "female", "male", "neutral" or "unknown".

- $number(m_i, t)$, $number(p_i, t)$ - indicates whether m_i or p_i is of number type n, where $n \in$ "singular", "plural", "unknown".

- $person(p_i, t)$ - indicates whether the pronoun, p_i, is a "first", "second" or "third" person pronoun.

- $type(m_i, t)$ - indicates whether the candidate mention, m_i, is a proper noun of type, "Person", "Organization" or "Location".

- $article\_type(m_i, a_j)$ - whether the candidate mention, m_j, has an article type, a, indicating it to be either an quantified noun phrase, indefinite or definite noun phrase.

- Grammatical Predicates: The behavior and the relations between the antecedents of pronouns are highly affected by different types of pronouns. For example, reflexive pronouns and possessive pronouns are less likely to be the antecedents of other pronoun and more likely to be non-referential pronouns. Hence, we design the following predicates to capture the types of pronoun:

  - $pronoun(m_i)$ - whether candidate mention, m_i, is a pronoun
  - $reflexive(m_i)$, $reflexive(p_i)$ - whether the candidate mention, m_i, or the pronoun, p_i, is a reflexive pronoun
  - $possessive(m_i)$, $possessive(p_i)$ - whether the candidate mention, m_i, or the pronoun, p_i, is a possessive pronoun

- Contextual Predicates: Since the pronoun itself provides little information, contextual information regarding the pronoun can provide more clues to resolve a pronoun with its antecedent. Contextual information describing the surrounding information of the pronoun and the mention are captured by the following predicates.

  - $has\_term(m_i, w_k)$, $has\_term(p_i, w_k)$ - indicates that the candidate mention or the pronoun contains the term, w_k.

- $left\_pos\_tag(m_i, t_k)$, $left\_pos\_tag(p_i, t_k)$ - the POS tag, t_k, of the word on the left of the candidate mention or the pronoun.

- $right\_pos\_tag(m_i, t_k)$, $right\_pos\_tag(p_i, t_k)$ - the POS tag, t_k of the word on the right of the candidate mention or the pronoun.

- $around\_pos\_tag(m_i, t_k)$, $around\_pos\_tag(p_i, t_k)$ - the POS tag, t_k of the words around the candidate mention or the pronoun.

- *embeded*(m_i) - indicates whether the candidate mention is embedded in another noun phrase.

- $nearest(p_i, m_j)$ - indicates whether the candidate mention is the nearest noun phrase toward the pronoun that agrees in number, gender and person type.

With the above predicates, constraints can be captured. First-order logic formula can be constructed in form of Equation 4.2 as exemplified as:

$$has\_term(p_j, w_k) \wedge type(m_i, t_l) \Rightarrow isAntecedent(m_i, p_j) \tag{4.3}$$

This formula captures the preference of a pronoun toward a specific type of proper nouns. Another example is:

$$same\_sent(p_j, m_i) \wedge same\_str(p_j, m_i) \Rightarrow isAntecedent(m_i, p_j) \tag{4.4}$$

It is obvious that a referential pronoun has to agree in gender with its antecedent. For number agreement, similar rules are designed:

$$gender(p_j, g_k) \wedge gender(m_i, g_k) \Rightarrow isAntecedent(m_i, p_j) \tag{4.5}$$

$$isAntecedent(m_i, p_j) \wedge gender(p_j, g_k) \Rightarrow gender(m_i, g_k) \tag{4.6}$$

Moreover, the probabilities of a certain type of pronoun to be non-referential are captured by the following formulae for each w_k:

$$has\_term(p_j, w_k) \Rightarrow isAntecedent(null, p_j) \tag{4.7}$$

$$left\_pos\_tag(p_j, t_k) \Rightarrow isAntecedent(null, p_j) \tag{4.8}$$

$$right\_pos\_tag(p_j, t_k) \Rightarrow isAntecedent(null, p_j) \tag{4.9}$$

$$around\_pos\_tag(p_j, t_k) \Rightarrow isAntecedent(null, p_j) \tag{4.10}$$

Similar rules are also constructed for capturing the probability of POS tags.

In resolving pronoun resolution, salience value on the candidate mentions are usually evaluated by the distance between the candidate mention and the pronoun. A negated prior weight is assigned to each pair of pronoun and the candidate mention, to favour closer candidate mentions.

We have conducted experiments demonstrating the use of our pronoun resolution knowledge base in knowledge base refinement approaches. The results are presented in Chapter 5.3.1, Chapter 6.1.2 and Chapter 6.3.3.

Chapter 5

Knowledge Base Refinement with Unlabeled Target Domain Data

In this chapter, we look into the situation of using unlabeled target domain data of our problem settings. One common requirement in different learning tasks for text mining is the preparation of labeled data. Manual annotation for text data becomes extremely time consuming due to the high complexity and large amount of textual information. In contrast, unlabeled data are more readily available. Hence, we propose a framework of knowledge base refinement for Markov Logic Networks (MLN) under a situation where an

MLN has already been learned from the labeled data in the source domain and only unlabeled data from the target domain is used.

The overview of our framework is depicted in Figure 5.1. As mentioned in Chapter 3.2, the knowledge base refinement of MLN consists of two components, namely, formula weight adaptation and logic formula refinement. First, the formula weights of the source domain MLN, MLN_s, is revised by the formula weight adaptation component. Second, by analyzing the adapted weights W_t, we refine the set of source domain formulae F_s. Finally, the refined set of formulae F_t together with updated weights W_t' is proposed with formula weight adaptation to obtain the final target domain MLN, MLN_t.

In the following sections, we describe the propose approaches for tackling each of the two components.

5.1 Formula Weight Adaptation

We propose two approaches, namely distribution divergence approach and penalty-based approach, to tackle the formula weight adaptation problem. Recall that given the labeled training examples L_s in domain D_s, the ordinary MLN learning is to find a set of weight W_s of MLN_s, such that the objective function, the pseudo-likelihood of the groundings of all the evidential and query predicates in the training example as shown in Equation 3.3

is maximized. This is equivalent to maximizing the pseudo-log-likelihood function as follows:

$$\Gamma(W_s, L_s) = \sum_{l=1}^{n} \log P_{W_s}(X_l = x_l | MB(X_l)) \tag{5.1}$$

Since this function is convex, its gradient can be expressed as follows:

$$\frac{\partial}{\partial w_i}\Gamma(W_s, L_s)$$
$$= \sum_{l=1}^{n} [n_i(x) - P_{W_s}(X_l = 0|MB(X_l))n_i(x_{[X_l=0]}) - P_{W_s}(X_l = 1|MB(X_l))n_i(x_{[X_l=1]})] \tag{5.2}$$

where n is the number of ground predicates in the MLN, and $n_i(x)$ is the number of true groundings for the i-th formula in MLN_s considering the training examples L_s; $MB(X_l)$ refers to the turth values of the Markov blanket of X_l where the Markov blanet of the ground predicate X_l is the set of ground predicates that appear in some grounding of a formula with it; $n_i(x_{[X_l=h]})$ is the number of true groundings for the i-th formula when forcing $X_l = h$. Since the truth value for each grounding of the query predicates is known in L_s, we can employ efficient algorithm like limited-memory BFGS algorithm to solve the problem. However, target domain D_t, the truth value of the ground query predicate is unknown since only unlabeled data is available. In other words, the objective of MLN learning in the target domain D_t is to find a set of weights, namely, W_t, which is different from W_s in principle, such that the learned MLN, denoted as MLN_t can accurately predict the truth value of the ground query predicates in D_t. As a result, MLN learning in the target domain becomes nontrivial and Equation 5.1 is not adequate. The ordinary MLN learning cannot be applied to learn MLN_t, the MLN tailored to D_t

The main idea of our proposed approaches is that although the source and the target domain datasets may have different data distributions, they share certain similarities since they solve the same task. Hence, it can be observed that the distribution of the target domain may not deviate far from the the source domain. Our proposed approaches jointly seek to maximize the likelihood of the target domain and analyze the difference between the MLNs of the source and the target domains. The two approaches differ in their perspective of how to evaluate the differences of the source and the target domains. The distribution divergence approach directly evaluates the difference of the distributions of the source and the target domains, while the penalty-base approach considers the difference of the source and the target domains from the level of the truth values of the ground predicates in the network.

5.1.1 Distribution Divergence Approach

The distribution divergence approach is designed based on two rationales. The first rationale is that since we do not have training examples in the target domain D_t, instead of maximizing the pseudo-log-likelihood of the groundings of both evidential and query predicates, we consider the groundings of the evidential predicates, and the expected truth value of the groundings of the evidential predicates. Essentially, we consider the following pseudo-log-likelihood function in the target domain:

$$\Gamma'(W_t, U_t) =$$
$$\sum_{l=1}^{n} \sum_{Y' \in Y(X_l)} \log \sum_{y'=0,1} \{P_{W_t}(X_l = x_l | MB(X_l), Y' = y')P(Y' = y' | MB(Y'))\} \tag{5.3}$$

where n is the total number of grounded atoms in the unlabeled data U_t in D_t; $Y(X_l)$ is the set of grounded atoms for the query predicates that are contained in $MB(X_l)$. The second rationale of our framework is that since the source and target domains should share certain amount of similarities, MLN_s and MLN_t are likely to be similar. However, according to the gradient expressed in Equation 5.2, W_s and W_t are different as long as the distribution of the number of true groundings of any formula in the MLNs is different. As a result, we consider the maximization of Equation 5.3, at the same time, we aim at minimizing the difference between the distributions $P_{W_s}(Y =$

$y|MB(Y))$ and $P_{W_t}(Y = y|MB(Y))$, where Y is the grounded atoms for the query predicates. To achieve this, we employ Kullback-Leibler (KL) divergence to measure the distance between the distributions. Consequently, we define the objective function of our MLN adaptation framework as follows:

$$\begin{aligned}\Gamma''(W_s, W_t, U_t) &= \sum_{l=1}^{n} \log P_{W_s}(X_l^s = x_l^s | MB(X_l^s)) \qquad (5.4)\\ &+ \sum_{l=1}^{n} \sum_{Y' \in Y(X_l^t)} \log \sum_{y'=0,1} \{P_{W_t}(X_l^t = x_l^t | MB(X_l^t), Y' = y')P(Y' = y'|MB(Y'))\}\\ &- KL(P_{W_s}(Y|MB(Y))||P_{W_t}(Y|MB(Y))\end{aligned}$$

where the superscripts denote the domain from which the data comes from; $KL(P||Q)$ is the KL divergence between the probability distributions of P and Q. Consider the term

$$\sum_{l=1}^{n} \sum_{Y' \in Y(X_l^t)} \log \sum_{y'=0,1} \{P_{W_t}(X_l^t = x_l^t | MB(X_l^t), Y' = y')P(Y' = y'|MB(Y'))\} \qquad (5.5)$$

in Equation 5.4. If the truth value of Y' is known in the unlabeled data U in D_t, we can then set $P(Y' = h|MB(Y'))$ to 0 or 1, where $h = 0, 1$ accordingly. This results in the original pseudo-log-likelihood function of MLN learning. However since these values are unknown in the target domain, we derive the following expected pseudo-log-likelihood function:

$$\sum_{l=1}^{n} \sum_{Y' \in Y(X_l^t)} \sum_{y'=0,1} \log \{ P_{W_t}(X_l^t = x_l^t | MB(X_l^t), Y' = y') \qquad (5.6)$$

$$P(Y' = y' | MB(Y')) \}$$

According to Jensen's inequality and the concave property of the logarithm function, Equation 5.5 is bounded below by Equation 5.6. Hence, we can maximize the following revised objective function:

$$\begin{aligned} \Gamma'''(W_s, W_t, U_t) &= \sum_{l=1}^{n} \log P_{W_s}(X_l^s = x_l^s | MB(X_l^s)) \qquad (5.7) \\ &+ \sum_{l=1}^{n} \sum_{Y' \in Y(X_l^t)} \sum_{y'=0,1} \log \{ P_{W_t}(X_l^t = x_l^t | MB(X_l^t), Y' = y') \\ &P(Y' = y' | MB(Y')) \} - KL(P_{W_s}(Y|MB(Y)) || P_{W_t}(Y|MB(Y)) \end{aligned}$$

As $\Gamma'' \geq \Gamma'''$, we can approximate the maximum of Γ'' by maximizing Γ'''. We can find the gradient of Equation 5.7 and apply the limited-memory BFGS algorithm to find the optimal set of W_s and W_t. However, Equation 5.7 is no longer convex and the optimization may lead to local maximum. We develop our learning algorithm as depicted in Figure 5.2 to address the local optimal problem. Our algorithm first learns an MLN, namely, MLN_s using the training examples L_s in the source domain D_s. Since we have labeled training examples in D_s, we can learn the weight W_s which is optimal to D_s using standard MLN learning algorithm. After that, we aim at learning the MLN, namely, MLN_t for the target domain D_t by making use of the learned

MLN_s and the unlabeled data U_t in D_t. Referring to Equation 5.7, we can fix W_s as the weight obtained in MLN_s and obtain the following expression involving W_t by expanding the KL divergence:

$$\begin{aligned} F(W_t) &= \sum_{l=1}^{n} \sum_{Y' \in Y(X_l^t)} \sum_{y'=0,1} \log P_{W_t}(X_l^t = x_l^t | MB(X_l^t), Y' = y') \\ &\qquad\qquad P(Y' = y' | MB(Y')) \qquad (5.8) \\ &+ \sum_{Y' \in NE} \sum_{y'=0,1} P_{W_s}(Y' = y' | MB(Y')) \log P_{W_t}(Y' = y' | MB(Y')) \end{aligned}$$

where NE refers to set of all query predicates.

The gradient of $F(W_t)$ with respect to W_t can be derived as follows:

$$\begin{aligned}
&\tfrac{\partial}{\partial w_i} F(W_t) = \\
&\sum_{l=1}^{n} \sum_{Y' \in Y(X_l)} \sum_{y'=0,1} \{P_{W_t}(Y' = y'|MB(Y'))[f_i(\cdot, Y' = y') \\
&-P_{W_t}(X_l = 0|MB(X_l), Y' = y') f_i(X_l = 0, Y' = y') \\
&-P_{W_t}(X_l = 1|MB(X_l), Y' = y') f_i(X_l = 1, Y' = y')] \\
&+\log P_{W_t}(X_l = x_l|MB(X_l), Y' = y')\{ \\
&P(Y' = y'|MB(Y'))[f_i(\cdot, Y' = y') \\
&-P_{W_t}(X_l = 0|MB(X_l), Y' = y') f_i(X_l = 0, Y' = y') \\
&-P_{W_t}(X_l = 1|MB(X_l), Y' = y') f_i(X_l = 1, Y' = y')]\}\} \\
&+\sum_{l=1}^{n} \sum_{Y' \in Y(X_l)} \sum_{y'=0,1} \{P_{W_t}(Y' = y'|MB(Y'))[f_i(\cdot, Y' = y') \\
&-P_{W_t}(X_l = 0|MB(X_l), Y' = y') f_i(X_l = 0, Y' = y') \\
&-P_{W_t}(X_l = 1|MB(X_l), Y' = y') f_i(X_l = 1, Y' = y')] \\
&+\sum_{Y' \in NE} \sum_{y'=0,1} \{P_{W_s}(Y' = y'|MB(Y'))[f_i(\cdot, Y' = y') \\
&-P_{W_t}(Y' = 0|MB(Y')) f_i(\cdot, Y' = y') \\
&-P_{W_t}(Y' = 1|MB(Y')) f_i(\cdot, Y' = y')\}
\end{aligned} \tag{5.9}$$

where $f_i(\cdot, Y' = y')$ refers to the truth value of the i-th formula when Y' is constrained to be equal to y'; $f_i(X_l = h, Y' = y')$ refers to the truth value of the i-th formula when X_l and Y' are constrained to be equal to h and y' respectively. We can apply the limited-memory BFGS algorithm to optimize Equation 5.8.

There are two advantages for our algorithm compared with optimizing Equation 5.7 directly. The first advantage is that we start learning MLN_t based on the MLN_s, which is optimal to D_s. Although the objective function is still not convex, we can guarantee that the learned MLN_t using our framework achieves better performance compared with the direct application of MLN_s in D_t. The second advantage is that our method can effectively reduce the training time, Since MLN_s and MLN_t are learned separately using the data from their own domains, each training involves less amount of data.

```
# Our Adaptation Framework
INPUT: $MLN_s = \langle F_s, W_s \rangle$: An MLN for source domain $D_s$;
    $U_t$: A set of unlabeled data in target domain $D_t$
OUTPUT: $MLN_t$: An MLN for $D_t$
ALGORITHM:
1: $W_t \leftarrow$ Perform weight adaptation on $Fs, W_s$
2: $F_t, W_t' \leftarrow$ Perform Formula Refinement on $F_s, W_t$
3: $W_t'' \leftarrow$ Perform weight adaptation on $F_t, W_t'$
4: $MLN_t = \langle F_t, W_t'' \rangle$
```

Figure 5.1: An Outline of Our Refinement Framework in Using Unlabeled Target Domain Data.

```
# The Distribution Divergence Approach
INPUT: $L_s$: A set of training examples in source domain $D_s$
        $U_t$: A set of unlabeled data in target domain $D_t$
OUTPUT:$MLN_t$: An MLN for $D_t$
1 Apply ordinary MLN learning to train $MLN_s$ from $L_s$
    (This is conducted by invoking limited-memory
    BFGS algorithm to optimize Equation 5.3
    with the gradient depicted in Equation 5.2)
2 Create $MLN_t$
3 Initialize $W_t$ by setting $W_t = W_s$
4 Learn $MLN_t$ by making use of $U_t$ in $D_t$
    (This is conducted by invoking limited-memory
    BFGS algorithm to optimize Equation 5.8
    with the gradient depicted in Equation 5.9
```

Figure 5.2: An Outline of the Distribution Divergence Approach.

5.1.2 Penalty-based Approach

Recall that given a set of labeled training data L_s in the source domain D_s where the ground evidential predicates and the truth value of the ground query predicates are known. Let MLN_s be the MLN for D_s with the set of weights W_s. The pseudo-log-likelihood function of the training examples can be expressed as follows:

$$\Gamma(W_s, L_s) = \sum_{l=1}^{n} P_{W_s}(X_l = X_l | MB(X_l)) \tag{5.10}$$

where X_l and x_l refer to the l-th atom of any query predicate and the truth value of the l-th ground query predicate respectively. $MB(X_l)$ refers to the state of the X_l's Markov blanket. In the target domain D_t, the truth values of the ground query predicates in the unlabeled data U_t are unknown. As a result, MLN learning in the target domain becomes nontrivial and Equation 5.10 is not adequate.

Our penalty-based adaptation approach is designed based on two objectives. Similar to the distribution divergence approach, the first objective is that we aim at learning the set W_t such that MLN_t should be tailored to U_t. On the other hand, we observe that the source domain D_s and the target domain D_t should share certain similarity. Therefore, our second objective is to ensure that MLN_t will not deviate too far away from MLN_s. In light of this, we aim at maximizing the following objective function in our MLN adaptation approach:

$$\begin{aligned}
&\Gamma^*(U_t, W_s, W_t) \\
&= \sum_{l=1}^{n} \sum_{Y' \in Y(X_l)} \log \sum_{y'=0,1} \{P_{W_t}(X_l = x_l | MB(X_l), Y' = y') P(Y' = y' | MB(Y'))\} \\
&\quad -\delta Q(U_t, W_s, W_t)
\end{aligned} \tag{5.11}$$

where $Y(X_l)$ is the set of ground query predicates that are contained in $MB(X_l)$. $Q(U_t, W_s, W_t)$ is a penalty function with respect to U_t, W_s, and W_t, and δ is the penalty parameter.

Recall that our first objective is to find a set of W_t such that MLN_t is tailored to U_t. As the truth values of the ground query predicates are unknown, this is insufficient for learning MLN_t using the pseudo-log-likelihood expressed in Equation 5.10. Instead, for each ground query predicate Y_l in U_t, we aim at maximizing the likelihood of the ground evidential predicates X_l connected to Y_l. The first term of Equation 5.11 refers to the expected pseudo-log-likelihood function on the ground evidential predicates in U_t, with respect to the $P_{W_t}(Y_l = y' | MB(Y_l))$. Our second objective is to prevent MLN_t from deviating too far away from MLN_s. To achieve this, we introduce a penalty function $Q(U_t, W_s, W_t)$ that is defined as follows:

$$Q(U_t, W_s, W_t) = \sum_{l=1}^{N} \chi(y_l|_{W_s}, y_l|_{W_t}) \tag{5.12}$$

N is the number of ground query predicates whose truth values are unknown.

$y_l|_{W_s}$ and $y_l|_{W_t}$ are the predicted truth values for the ground query predicate Y_l in U_t using W_s and W_t respectively, and $\chi(x, y)$ is an indicator function which is equal to 1 if $x = y$ and 0 otherwise. It is obvious that $Q(U_t, W_s, W_t)$ increases as the number of disagreements for predicting the truth value of the ground predicates using MLN_s and MLN_t increases. As a result, by adjusting the penalty parameter δ, we can reduce the disagreement on prediction using MLN_s and MLN_t, and hence prevent MLN_t from deviating too far away from MLN_s in learning.

By assuming a certain degree of difference between the source and the target domain datasets, effects of the difference are considered. The weights of the formulae are learned with the implied effect of the difference towards the query predicates.

5.2 Logic Formula Refinement

A reason for poor performance for direct application of the source domain MLN_s to the target domain is that the set of logic formulae F_s in MLN_s may not be adequate for the target domain as it is originally designed for the source domain. Our approach aims at performing logic formula refinement which adapts the formulae of the MLN. The overview of our adaptation framework is depicted in Figure 5.3. Our algorithm first learns a set of formulae F_s for the source domain MLN_s using the source domain labeled data L_s as shown in Step 1. Since we have training examples in D_s, we can learn the formulae F_s using the standard structure learning from the source domain. In fact, an alternative way to obtain F_s is to directly construct the formulae manually from expert knowledge if available. After that, we aim at applying the logic formula adaptation on the set of source domain formulae F_s to obtain the set of target domain formulae F_t as shown in Step 3. Finally, MLN_t is created from the formulae F_t and the weights W_t. The logic formula adaptation algorithm in Step 3 of our framework is an important component. It aims at capturing the differences in the relations between the source domain and the target domain and adapt the relations from the source domain to the target domain. The logic formulae for the source domain may not be able to completely describe the relations of the

```
# The Logic Formula Refinement Framework
INPUT:
    $L_s$: A set of training examples in source domain $D_s$
    $U_t$: A set of unlabeled data in target domain $D_t$
OUTPUT:
    $MLN_t$: An MLN for $D_t$
ALGORITHM:
1: Apply ordinary MLN formula learning to discover
    the set of formulae $F_s$ from $L_s$
2: Apply ordinary MLN weight learning to train the
    weight $W_s$ for the set of formulae $F_t$ from $L_s$
3: Perform logic formula refinement algorithm to train
    the set of formulae $F_t$ from $F_s$, $L_s$, and $U_t$
        3.1:    $F_c \leftarrow$ Core formula identification algorithm
        3.2:    $P_f \leftarrow$ Candidate pattern identification algorithm
        3.3:    $(F_a, W_a) \leftarrow$ New formula construction algorithm
        3.4:    $F_t \leftarrow F_s \cup F_a$; $W_t \leftarrow W_t \cup W_a$
4: Create $MLN_t$ with $F_t$ and $W_t$
```

Figure 5.3: An Outline of the Logic Formula Refinement.

target domain. For example, for the pronoun resolution task, the bags of words are different for the two domains, new formulae describing the unseen words from the target domain have to be discovered. The set of source domain logic formulae F_s has to be modified to include the new relations of the target domain. We develop the logic formula refinement algorithm to discover these new relations of the target domain using the unlabeled data U_t in the target domain. The challenge of discovering the new relations lies in that with only unlabeled data of the target domain where the truth values of the query predicates are unknown, it is not easy to establish the relations between the evidential predicates and the query predicates. Hence, we propose a method utilizing not only the unlabeled data U_t, but also the source domain labeled data L_s and formulae F_s. We focus on analyzing the relations describing the implication of the query predicates from the evidential predicates. Specifically, the pattern of evidential predicates, which leads to the conclusion of query predicates, is expressed in the form of $(E_1 \wedge E_2 \cdots \wedge E_n) \Rightarrow N$ where $E_i \in E$ (evidential predicates) and $N \in Y$ (query predicates).

Our rationale is that the relations in the target domain share some similar behaviors with certain source domain relations. Some formulae of F_s should be equally applicable for both the source and the target domains.

As shown in Step 3.1 in Figure 5.3, we develop an algorithm to identify these core formulae F_c. Then, we try to discover the new relations by considering some frequent patterns that are specific to the target domain. These potential candidate patterns P_f for constructing new formulae of the target domain are identified by our proposed algorithm as given in Step 3.2. Finally, the new formulae F_a capturing the new relations are constructed using our proposed algorithm as shown in Step 3.3. We analyze the correlations of the candidate patterns P_f with the core formulae F_c and establish the relations for the target domain. In the final Step 3.4, the new formulae F_a together with the source domain formulae F_s form the set of refined formulae for the target domain F_t.

5.2.1 Core Formula Identification Algorithm

The identification of core formulae aims to discover which formulae are of great significance to both domains. It is the step 3.1 for our logic formulae adaptation as shown in Figure 5.3. We develop an approach as depicted in Figure 5.4 to identify the core formulae using the unlabeled data of the target domain. However, without the truth values of the query predicates in the unlabeled target domain data, it is challenging to capture the significance of the formulae towards the target domain. First, for each source domain for-

```
# Identifying core formulae $F_c$ for
        both the source and the target domain
INPUT:
    $L_s$: A set of training examples in source domain $D_s$
    $U_t$: A set of unlabeled data in target domain $D_t$
    $F_s$: A set of formulae $F_s$ for the source domain
    $\theta$: A threshold for selecting the core formulae
    $Y$: Query predicates
OUTPUT:
    $F_c$: A set of core formulae
ALGORITHM:
$P \leftarrow \emptyset$; $F_c \leftarrow \emptyset$
for each $f_i \in F_s$
    Create $p_i$ by removing the query predicates $Y$ from $f_i$
    Create $p'_i$ by extracting the query predicates $Y$ from $f_i$
    $l \leftarrow$ Find the length of pattern $p_i$
    Create a set of patterns $P_i^c$ where $\forall p_j \in P_i^c$ has length $\leq l$
```

```
7   for each $p_j \in P_i^c$
8       Create the formula $f'_j$ from $p_j$ and $p'_i$
9       Calculate $S(f'_j, L, U)$
10      if $(S(f'_j, L, U) \geq \theta)$ & $(p_j \ni P_c)$
11          Add $f'_j$ to $F_c$
12      end if
13  end for
```

Figure 5.4: Core Formula Identification Algorithm

mula f_i, instead of the whole formula, we remove the query predicates from the formulae to form a set of patterns P_i^c as shown in Line 3 in Figure 5.4. A pattern $p_j \in P_i^c$ can be any subset of evidential predicates from the formula f_i. For example, suppose a formula $A \wedge B \Rightarrow C$, where A and B are the evidential predicates and C is the query predicate. After removing the query predicate C, three patterns, namely, $A \wedge B$, A, and B are obtained. Then, in Line 4, we create a pattern p'_i by retaining only query predicates in f_i. Returning to the above example, the query predicate pattern p'_i is C. As in Line 8, a pattern p_j together with the query predicate pattern p'_i of f_i forms a candidate formula f'_j. Following the same example, the three candidate formulae are constructed, namely, $A \wedge B \Rightarrow C$, $A \Rightarrow C$, and $B \Rightarrow C$.

Definition 1 A pattern $p_i : (g_1 \wedge g_2 \cdots \wedge g_n)$ **is satisfied by** I *if the grounding of the pattern* p_i *is true given an interpretation, i.e. a set of constants,* $I \in D$, $\models_I p_i$ *where* $g_1, g_2, \ldots, g_n$ *represent the predicates and* D *represents the dataset.*

Definition 2 A formula $f_j : p_j \Rightarrow p'_j$ **is satisfied given the patterns** p_j **and** p'_j *in dataset* D *if*
$\forall I \in D, \models_I (p_j \Rightarrow p'_j)$ *if* $\models_I p_j$ *and* $\models_I p'_j$

The next task is to develop a method for selecting core formulae among the candidate formulae. First, in most of the time, f_j should be satisfied given the corresponding evidential pattern p_j and the query pattern p'_j in the source domain labeled data L_s. Definition 2 provides a definition for the conditions for a formula is satisfied given the patterns. If for all groundings of the pattern p_j by L_s that is true and that the same grounding of the pattern p'_j is true, then the corresponding grounding of the formula f_j by L_s is also true. Under such condition, we regard f_j as being satisfied given the patterns p_j and p'_j. Let $N_p(x, D)$ be the number of true groundings for formula or pattern x given the dataset D, and $N_t(x, D)$ be the number of grounding for formula or pattern x given the dataset D. This criterion can be computed by $N_p(f'_j, L)/N_p(p_j, L)$, the ratio of whether the formula is satisfied given the patterns p_j and p'_j in the source domain labeled data L_s. Second, the corresponding pattern p_j is relatively frequent in the target domain unlabeled data U_t. This criterion is evaluated with $N_p(p_j, U)/N_t(p_j, U)$, the ratio of the pattern p_j is true in the target domain unlabeled data U_t. Hence, we design a significance score, $S(f'_j, L, U)$, based on the above criteria for each formula f'_j. In Line 9, we calculate the significance score for each candidate formula f'_j by Equation 5.13.

$$S(f'_j, L, U) = \frac{N_p(f'_j, L)}{N_p(p_j, L)} * \frac{N_p(p_j, U)}{N_t(p_j, U)} \tag{5.13}$$

Finally, in Lines 10 to 11, we select the formulae f'_j where $S(f'_j, L, U) \geq \theta$ to form the set of core formulae F_c. This ensures that a core formula's corresponding pattern p_j appears both in the source and the target domain.

5.2.2 Candidate Pattern Identification Algorithm

Frequent patterns of predicates in the target domain are more likely to have greater influence in the inference of truth values for the groundings of the

```
# Identifying candidate patterns $P_f$
INPUT:
    $L_s$: A set of training examples in source domain $D_s$
    $U_t$: A set of unlabeled data in target domain $D_t$
    $max\_len$: the maximum length of the candidate patterns
    $min\_freq$: the minimum frequency for a candidate pattern to be selected
    $V$: the types of constants to be variablized
OUTPUT: $P_f$: A set of candidate patterns
FUNCTION: $n(p)$: the frequency for a pattern $p$
ALGORITHM:
    Compare $L_s$ with $U_t$ to obtain the distinctive
    ground evidential predicates $E_t$
    $P \leftarrow \emptyset$; $P_f \leftarrow \emptyset$
    for $l \leq max\_len$
        Create a set of paths $P'$ by generating all the paths
        of connected ground predicates with length $l$ from $U_t$
        for each $p' \in P'$
            if $P'$ contains any $e_j \in E_t$
                Add $p'$ to $P$
            end if
        end for
```

```
for each $p \in P$
    $p'' \leftarrow$ variablize $p$ by replacing the constants of type $v \in V$
    if $p'' \in P_f$
        $n(p'') \leftarrow n(p'') + 1$
    else
        $P_f \leftarrow P_f \cup p;\ n(p'') \leftarrow 1$
    end if
end for
for each $p \in P_f$
    if $n(p'') \leq min\_freq$
        Remove $p''$ from $P_f$
    end if
end for
```

Figure 5.5: Candidate Pattern Identification Algorithm.

query predicates. These frequent patterns can be candidate patterns for constructing the new formulae for the target domain. Since one of our goals is to discover new relations in the target domain, we aim at finding patterns involving ground evidential predicates that are specific to the target domain.

Definition 3 *A ground predicate $g_i(a_1, a_2, ..., a_i, ..., a_m)$ is* **connected** *to another ground predicate $g_j(b_1, b_2, ..., b_j, ..., b_n)$ if $\exists a_i \; \exists b_j \; : \; (a_i = b_j)$ where g_i, g_j are m-ary and n-ary predicate respectively, and $a_1, a2, ..., a_m, b_1, b_2, ..., b_n$ are the constants for the arguments respectively.*

Definition 4 *A* **path** *p of* **length** *l is a series of l distinct ground predicates $p = (g_1, g_2, ..., g_k, ..., g_l), \forall g_i \in D$ where D represents the database, such that for $1 < k \leq l$:*

1. *the kth ground predicate g_k is connected to the $(k-1)$th ground predicate g_{k-1}, and*
2. *for $1 < i \leq k-1$, $g_k \neq g_i$.*

Figure 5.5 depicts the algorithm of identifying the candidate patterns. We first identify the set of ground evidential predicates E_t which only appear in the target domain unlabeled data U_t as shown in Line 1. Candidate patterns are then constructed from paths of connected ground evidential predicates in U_t. A path is a series of connected ground predicates as given in Definitions 3 and 4. For example, the path, $E(a,b) \wedge F(a,c)$, where the two ground predicates, $E(a,b)$ and $F(a,c)$, are connected with the constant, a, is of length 2. In Line 4, paths whose length are shorter or equal to the maximum length value specified are added to the set of candidate paths P'. Then in Lines 5 to 9, candidate paths in P' which contain at least one ground predicate from E_t are selected. The candidate paths are variablized where some constants are replaced with variables to form candidate patterns p'' in Line 11. Using the above example, in the logic expression $E(a,b) \wedge F(a,c)$, since the constants b and c are not the focus of interest, they are replaced by variables ν_1 and ν_2 to form the pattern $E(a,\nu_1) \wedge F(a,\nu_2)$. Finally, in Lines 18 to 20, only patterns p'' with a frequency $n(p'')$ of value greater than min_freq are selected to form a set of candidate patterns P_f.

5.2.3 New Formula Construction Algorithm

Given the candidate patterns P_f, it is not sufficient for constructing formulae from them as we do not know how the query predicates are associated with the patterns with only unlabeled data U_t from the target domain data D_t. However, if a set of formulae is known to agree with the target domain D_t,

```
# Constructing new formulae $F_a$
INPUT:
    $U_t$: A set of unlabeled data in target domain $D_t$
    $Y$: Query predicates
    $F_c$: the core formulae
    $P_f$: the candidate patterns
    $W^s$: the set of weights for $F_s$
OUTPUT:
    $F_a$: A set of adapted formulae for the target domain $D_t$
FUNCTION:
    $n(p)$: the frequency for a pattern $p$
ALGORITHM:
for each $p_m \in P_f$
    $A_m \leftarrow \emptyset$
    for each $f_n \in F_c$
        Create $p_n$ by removing the query predicates $Y$ from $f_n$
        Create $p'_n$ by extracting the query predicates $Y$ from $f_n$
        Add $p'_n$ to $P$
        if $\rho(p_m, p_n) \geq 0$
            $n(p'_n) \leftarrow n(p'_n) + 1$
            Add $f_n$ to $A_m$
        end if
    end for
    $p'' \leftarrow$ select $p'_n \in P$ with maximum $n(p'_n)$
    Create the formula $f'_i$ from $p_m$ and $p''$
    Add $f'_i$ to $F_a$
    $w_i \leftarrow \sum_{f_i \in A_m} {w^s}_i/|A_m|$ where ${w^s}_i \in W_s$
end for
```

Figure 5.6: New Formula Construction Algorithm.

it have some closely related patterns from P_f providing us hints on how those patterns are associated with the query predicates. This is performed by analyzing the correlations between the candidate patterns P_f and the core formulae F_c.

Figure 5.6 depicts the algorithm for constructing the new formulae F_a. For each target domain candidate pattern p_m, we calculate its correlations with each of the core formulae in F_c. This is done by first obtaining the evidential predicate pattern p_n and the query predicate pattern p'_n of each core formula f_c in Lines 4 and 5. Then, in Line 7, for each pair of $\langle p_m, p_n \rangle$, we calculate the correlation coefficient $\rho(p_m, p_n)$ as given in Equation 5.14. The positively correlated core patterns p_n are identified for the candidate pattern p_m. A positive correlation coefficient $\rho(p_m, p_n)$ indicates that the candidate pattern p_m is closely related to core formula f_n. Since each correlated important pattern p_n is associated with a query predicate pattern p'_n, a query predicate p'_n will be selected by majority voting on the set of query predicate patterns associated. In Lines 11 to 13, the most frequent query predicate pattern among the correlated important patterns are selected and combined with the target domain candidate pattern p_m to form a new formula. Finally, in Line 15, the weight of the new formula f'_i is set as the normalized sum of the weights of the correlated core formulae.

The correlation coefficient for two patterns p_m and p_n is defined as:

$$\rho(p_m, p_n) = \frac{Cov(p_m, p_n)}{\sqrt{Var(p_m) * Var(p_n)}} \tag{5.14}$$

where $Cov(p_m, p_n)$ denotes the covariance between patterns p_m and p_n as defined in Equation 5.17, and $Var(p_m)$ and $Var(p_n)$ represent the variances of patterns p_m and p_n as defined in Equations 5.15 and 5.16 respectively,

$$Var(p_m) = E[(p_m - \mu)^2] = E[(p_m)^2] - (E[p_m])^2 \tag{5.15}$$

$$Var(p_n) = E[(p_n - \mu)^2] \tag{5.16}$$

$$Cov(p_m, p_n) = E[(p_m - \mu)(p_n - \mu)] = E[p_m, p_n] - E[p_m]E[p_n] \tag{5.17}$$

Being variables for calculating the covariance and variance, the patterns p_m and p_n are considered as either satisfied or not, i.e., true(1) or false(0). As a result, the expected values of the variables p_m and p_n can be computed as follows:

$$E[p_m] = (1)R(p_m) + (0)(1 - R(p_m)) = R(p_m) \tag{5.18}$$

$$E[(p_m)^2] = 1^2 R(p_m) + 0^2(1 - R(p_m)) = R(p_m) \tag{5.19}$$

$$E[p_m, p_n] = (1)R(p_{mn}) + (0)(1 - R(p_{mn})) = R(p_{mn}) \tag{5.20}$$

where $R(p_{mn})$, $R(p_m)$, and $R(p_n)$ denote the ratio of the number of true groundings over the number of groundings for the patterns p_{mn}, p_m, and p_n

in the target domain unlabeled data U_t respectively. p_{mn} represents the joint pattern formed by the conjunction of the patterns p_m and p_n. Let $N_p(x, D)$ and $N_t(x, D)$ denote the number of true groundings for formula or pattern x and the number of grounding for formula or pattern x given the dataset D respectively.

$$R(p_m) = \frac{N_p(p_m, U)}{N_t(p_m, U)} \tag{5.21}$$

$$R(p_n) = \frac{N_p(p_n, U)}{N_t(p_n, U)} \tag{5.22}$$

$$R(p_{mn}) = \frac{N_p(p_{mn}, U)}{N_t(p_{mn}, U)} \tag{5.23}$$

Hence, by the substitution of Equations 5.18, 5.19, and 5.20 in Equations 5.15, 5.16, and 5.17 respectively, $Var(p_m)$, $Var(p_m)$, and $Cov(p_m, p_n)$ can be expressed as:

$$Var(p_m) = R(p_m) - (R(p_m))^2 \tag{5.24}$$

$$Var(p_n) = R(p_n) - (R(p_n))^2 \tag{5.25}$$

$$Cov(p_m, p_n) = R(p_{mn}) - (R(p_m) * R(p_n)) \tag{5.26}$$

5.3 Experiments

We have conducted experiments on two different tasks, namely, pronoun resolution and segmentation of citations. For each of the task, we prepared a source domain training dataset from which the source domain MLN is obtained. For the target domain, an unlabeled training dataset is prepared for adapting the source domain MLN to the target domain. Finally, a separate target domain testing dataset is used for evaluating the performance. Our framework is implemented based on the Alchemy system [34], which provides algorithms in statistical relational learning on the Markov Logic Network.

Since little related works have been developed for performing domain adaptation on MLN solving the same task, as a baseline for comparative evaluation, we employ a state-of-the-art transfer learning system of MLN, called SR2LR, which can be applied to conduct domain adaptation on MLN solving the same task [41]. SR2LR transfers a source domain MLN model to a target domain MLN model using single-entity-centered examples with respect to an entity in the target domain. In a single-entity-centered example, the truth values of all the facts, i.e., ground predicates, involving the same entity, i.e., the central entity, are given and the truth values of all other facts not involving the central entity are not necessarily provided. It is applicable to revising an existing source domain MLN model to a target domain solving the same task by assuming there is only one available predicate mapping between the source and the target domain, SR2LR algorithm hence filters out formulae which are not informative regarding the target domain. The weight of the target domain formula is assigned the same weight as that in the source domain MLN under such single-predicate mapping setting.

5.3.1 Pronoun Resolution

Task Description

We used two text document corpora, namely, the ACE 2004 [24] corpus and the OntoNotes [29, 46] project for conducting the pronoun resolution experiment. The objective of this application is to determine a pronoun's antecedent among a set of noun phrases, to which are referred as candidate mentions. As described in Chapter 4, in our MLN model, all pairs of pronoun and candidate mention within a document will be jointly resolved for their antecedents. Moreover, our goal of this task is to handle all kinds of personal pronouns and noun phrases in a single resolution model.

In the ACE corpus, only noun phrases with ACE named entity types (i.e. Person, Organization, Location, Facility, Weapon, Vehicle and Geo-Political Entity) are annotated. Pronouns referring to a non-ACE named entity are unannotated. Under such definition of annotation which is not complete, we can only assume those unannotated pronouns as non-referential pronouns. This assumption is commonly made in all previous works that use the ACE corpus for pronoun resolution in their experiments. The OntoNotes corpus has annotations for all coreference relations of all kinds of entities. Therefore, unlike the ACE corpus, the OntoNotes corpus does not restrict the annotation of coreference relations over ACE types of entities. It also addresses the general anaphoric coreference relations. Coreference noun phrases are annotated over all possible noun phrases, even including temporal expressions. Hence, all referential pronouns are annotated with their corresponding antecedents where all other unannotated pronouns are non-referential pronouns.

Experimental Setup

In our experiments, 96 text documents from the newswire segment of the ACE corpus were randomly extracted and used as the source domain dataset. 99 unlabeled text documents were randomly extracted from the OntoNotes corpus for the target domain dataset. The dataset of target domain unlabeled documents is randomly split into two subsets: 50% for the unlabeled target domain training dataset and 50% for the testing dataset. As a result, there are 1,842 pronouns in the source domain dataset, and 931 and 703 pronouns in the target domains unlabeled training dataset and testing dataset respectively. There are 8,918 candidate mentions in the source domain dataset, and

7,125 and 5,569 candidate mentions in the target domains unlabeled training dataset and testing dataset respectively. Queries are selected automatically for labeling from the target domain unlabeled training dataset.

As with common experimental setup in previous works, we focus on the evaluation of the resolution ability, we use the true mentions as candidate mentions. In the ACE corpus, true ACE mention boundaries are annotated. However, for the OntoNotes corpus, we follow the annotation guidelines and identify the true boundaries of noun phrases as candidate mentions. In both corpora, all personal pronouns are identified by a predefined list of pronouns. Positive training instances are created by pairing the pronoun with its closest antecedent. Negative instances are created for all pairs of pronouns and preceding noun phrases within a context window. We adopt a window size of 3 for our experiments, meaning that noun phrases appeared within the sentence containing the pronoun and the preceding 3 sentences are considered.

All documents were preprocessed with the Stanford Named Entity Recognizer [1][25]. Sentences were preprocessed with the POS tagger [2][58]. We employed the noun gender data developed by Bergsma [3][1]. With this corpus, we obtained the gender and number information for a mention.

The source domain MLNs is constructed on the preprocessed labeled data

[1] http://www-nlp.stanford.edu/software/CRF-NER.shtml

[2] http://www-nlp.stanford.edu/software/tagger.shtml

[3] http://www.cs.ualberta.ca/ bergsma/Gender/

based on the knowledge base design described above and in Chapter 4. For the experiment on SR2LR, we randomly selected a pronoun p_i as the central entity, and created the single-entity example. For example, the truth value of the ground query $isAntecendent(m_j, p_i)$, where m_j is a candidate mention in the same document, is given in the example. The number of ground query predicates to be labeled in the single-entity-centered example is 729. In the experiment for our model, the penalty parameter δ was set to a value of 5 in the penalty-based formula weight refinement. The maximum length of candidate patterns was set to 3 and the minimum frequency for a candidate pattern to be selected was 10.

We also investigate the performance of our two proposed formula weight refinement approaches, namely, distribution divergence and penalty-based. In our complete model, we perform knowledge base refinement using our framework as described in Figure 5.1 using both the penalty-based formula weight refinement approach (Chapter 5.1.2) and logic formula refinement (Chapter 5.2).

Evaluation Metric

As we also handle non-referential (non-anaphoric) pronouns, we adopt the modified accuracy metric, known as resolution etiquette [42], which is also used in Cherry and Bergsma [16], and Charnial and Elsner [14]. It is defined as the proportion of pronouns correctly resolved, either to a candidate mention or to the non-referential category.

Experimental Results

Table 5.1 depicts the performance of our approaches comparing to the SR2LR model. The significant increase in the performance of all our approaches and our complete model demonstrates that they are able to refine the existing source domain model for the target domain. The improvement of our complete model over the penalty-based approach also demonstrates that the new logic formulae discovered is useful for capturing the difference in the underlying relations between the source and the target domains. Precisely, 127 new formulae were discovered which represent the new relations for the target domain. An example of a new formula is:

$$next\_sent(x, y) \wedge has\_term(y, \text{``}schemer\text{''}) \Rightarrow isAntecedent(y, x)$$

In the above formulae, the term "schemer" is a word specifically appeared in the target domain only. Through our refinement model, we are able to establish relations between the evidential predicates containing the term with the query $isAntecedent(y, x)$. All the formulae are also refined to fit the data distribution of the target domain. For example, the following formula

System	Accuracy
SR2LR	16.1%
distribution divergence	22.1%
penalty-based	31.3%
our complete model	34.7%

Table 5.1: Experimental Results of Using Unlabeled Target Domain Data for Pronoun Resolution

originally has a high weight of 5.03 in the source domain MLNs.

$$has\_term(x, \text{“}it\text{”}) \Rightarrow isAntecedent(null, x)$$

After weight refinement, it has a much lower weight of 1.22 in the target domain MLN, MLN_t. This captures one characteristic that referential pronouns have different definitions between the source ACE corpus and the target OntoNotes corpus.

Since most of the groundings of the formulae in the existing source pronoun resolution MLN have ground predicates not involving the central entity, those formulae will be directly selected as the formulae for the target domain pronoun resolution MLN by the SR2LR model. The final target domain MLN is hence roughly the same as the source domain MLN with the same set of weights. Our proposed model has the advantage of discovering new formulae and refining the corresponding weights of the formulae to the data distribution of the target domain.

5.3.2 Segmentation of Citation Records

Task Description

The goal of segmentation of citation records is to extract bibliographical records of technical paper citations and identify the candidate fields, namely, title, author, and venue, from the citation strings. We employed the segmentation MLN model developed by Singla and Domingos [55] for our experiments. The corresponding query is $InField(i, f, c)$. $InField(i, f, c)$ is true if and only if the i-th position of the c-th citation is part of the field f, where $f \in \{title, author, venue\}$. The main evidential predicate is $HasToken(t, d, c)$ which represents the citation strings. $HasToken(t, d, c)$ is true if and only if the c-th citation has a token t in the d-th position where t is a the token in the source domain labeled data.

Moreover, there are predicates describing a string, the information on punctuations, and positional information regarding the citations. With the predicates, four main categories of relations, namely, mutual exclusive, word and field regression, signature words and position rules, are formulated in the segmentation model. For example, the following formula, describing whether a word t_j is part of the field f_n, is created for every word in the source domain data.

$$HasToken(t_j, i, c) \Rightarrow InField(i, f_n, c) \tag{5.27}$$

Experimental Setup

We conducted experiments on CiteSeer, one of the standard datasets used for information extraction on citations. The CiteSeer dataset was first created by Lawrence et al. [36]. The CiteSeer dataset has approximately 1,500 citations to 900 papers, and it contains four different topic sections, namely, constraint satisfaction, face recognition, automated reasoning, and reinforcement learning. Each of the four sections are used as the source domain data separately. For target domain data, we combine two of the sections. Then the target domain data is randomly split into two subsets: 50% for the target domain unlabeled training dataset and 50% for the testing dataset. For example, when the section, face recognition, is used as the source domain, we combine two of the remaining sections, constraint satisfaction and reinforcement learning, and divide them randomly to form the target domain unlabeled training dataset and the testing dataset. We perform experiments on all the combinations of source domain and target domain assignments. A total of 12 sets of datasets are obtained.

We perform experiments on the 12 datasets using the SR2LR model, our two formula weight adaptation approaches, namely, distribution divergence and penalty-based, and our complete model. In our complete model, we perform knowledge base refinement using our framework as described in Figure 5.1 using the penalty-based formula weight refinement approach (Chapter 5.1.2) and logic formula refinement (Chapter 5.2).

For the experiment on the SR2LR model, we randomly selected a citation c_i as the central entity for each of the target domains, and created the single-entity example. For example, the truth value of the ground query $InField(c_i, author, p_j)$ is true and is given in the example. The number of ground query predicates required for labeling in the single-entity-centered examples is 18.

In the experiment for our approaches, the penalty parameter δ was set to a value of 1 in the weight refinement. The maximum length of candidate patterns was set to 3 and the minimum frequency for a candidate pattern to be selected was 10.

Evaluation Metric

We measure the performance of the query predicate $InField(i, c, p)$. The F-measure metric is used for evaluating the citation segmentation performance. Specifically, the F_1 measure is calculated by the weighted harmonic mean of Precision(P) and Recall(R).

$$P = \frac{tp}{tp + fp} \tag{5.28}$$

$$R = \frac{tp}{tp + fn} \tag{5.29}$$

where tp, true positive, is the number of query predicts correctly labeled as true; tn, true negative, is the number of query predicts correctly labeled as false; fp, false positive, is the number of query predicts incorrectly labeled as true; and fn, false negative, is the number of query predicts incorrectly labeled as false.

$$F_1 = 2\frac{P \cdot R}{P + R} \tag{5.30}$$

Experimental Results

Table 5.2 shows our average refinement performance of the 12 datasets for the segmentation task. The performance of our model demonstrates consistent improvement obtained by our model over the SR2LR model. Both the distribution divergence approach and the penalty-based approach outperform the baseline approach. The improvement of our complete model over using formula weight refinement approach alone shows that our logic formula refinement approach is able to construct useful new formulae for the target domains. The detailed results of each of the 12 datasets are shown in

System	Average F-measure	p-value(compared with SR2LR)
SR2LR	75.0%	-
distribution divergence	76.2%	0.008
penalty-based	76.5%	0.028
our complete model	76.8%	0.012

Table 5.2: Experimental Results of Using Unlabeled Target Domain Data for Segmentation of Citation Records

Table 5.3.2. and performance for each field, namely, author, title and venue, are shown in the Appendix A.1.

Some examples of new formulae are:

$$HasToken(\text{``}satisfaction\text{''}, i, c) \Rightarrow InField(i, \text{``}title\text{''}, c)$$

$$HasComma(c, i) \wedge FollowBy(c, i, COMMA) \wedge HasToken(\text{``}co\text{''}, i, c) \Rightarrow InField(c, \text{``}venue\text{''}$$

where $HasComma(c, i)$ and $FollowBy(c, i, t)$ describes if a citation c has or is followed by a certain punctuation at the i-th position. The first formula depicts that the word "satisfaction" is part of the title, while the second formula depicts that if the word "co" contains a comma, and is also followed by a comma, then it is part of the venue.

Moreover, we conducted statistical significance test using McNemar's paired t-tests over the 12 datasets and all our approaches were found to be statistically better that SR2LR (p-value < 0.05 with a 95% confidence interval). When comparing our complete model with the penalty-based formulae refinement approach, the p-value is 0.0014. Hence, the improvement of our complete model is statistically significant.

For the SR2LR model, since most of the formulae in the source domain model consist of predicates with constants, and facts about those constants. For example, the truth value of the ground evidential predicate $IsDate(t_j)$, which describes if the string t_j is a date, is unknown as it does not involve the centered entity c_i. Those formulae are directly regarded as target domain formulae without any modification on the weights by SR2LR, whereas our model discovers new formulae and refines all formulae of the source MLN with regard to the data distribution of the target domain.

Source Domain	Target Domain	SR2LR	distribution divergence	penalty based	our complete model
reasoning	constraint+face	77.4%	78.0%	78.2%	78.4%
reinforcement	constraint+face	73.2%	74.1%	73.4%	74.1%
face	constraint+reasoning	73.8%	76.5%	78.5%	78.9%
reinforcement	constraint+reasoning	75.3%	76.0%	75.7%	76.3%
face	constraint+reinforcement	74.3%	76.6%	79.1%	79.2%
reasoning	constraint+reinforcement	80.9%	81.2%	81.4%	81.4%
constraint	face+reasoning	72.0%	73.4%	72.5%	74.7%
reinforcement	face+reasoning	74.4%	74.9%	74.3%	74.7%
constraint	face+reinforcement	72.2%	73.2%	72.5%	72.8%
reasoning	face+reinforcement	79.8%	80.4%	80.3%	80.5%
constraint	reasoning+reinforcement	73.5%	74.3%	73.8%	73.9%
face	reasoning+reinforcement	72.8%	75.9%	78.2%	78.3%

Table 5.3: Experimental Results of Using Unlabeled Target Domain Data for All Datasets of Segmentation of Citation Matching

Chapter 6

Knowledge Base Refinement Using Limited Labeled Target Domain Data

In this chapter, we investigate the situation of using a limited amount of labeled target domain data for knowledge base refinement. When a small amount of manual annotations can be obtained, we investigate on how to utilize such limited resources at the largest advantage. We develop two approaches of active learning for knowledge base refinement for the target domain, namely, pool-based and error-driven approaches.

The source domain and the target domain share some similarities and yet they are different. To refine the source domain knowledge for the target domain, we try to capture the differences among the two domains. In Chapter 5.2, we discover the differences using unlabeled data. Potential logic formulae for the target domain are discovered and a different target domain MLN, MLN_t, can be constructed. But if we are given a small amount of labeled data, we could further improve the performance of MLN_t by evaluating these potential logic formulae using labeled data. To achieve this objective, we develop two approaches, namely, pool-based approach and error-driven approach.

6.1 Pool-based Approach

In the pool-based approach, we investigate how to actively select query predicates for manual labeling such that the labeled obtained can aid the process of model refinement. Our rationale is that logic formulae specific to the target domain represent different characteristics of the source domain and the target domain. The impact of the new logic formulae can serve as important clues for refining the predictions and hence for better estimation of the target domain's distribution. The logic formulae captured for the target domain MLN, MLN_t, which are not included in the source domain MLN, MLN_s, cause the ground query predicates in the target domain to have different inference results. When no labeled data for the target domain is given, the change in inference results cannot be verified. There is a risk that the logic formulae may deduce an incorrect conclusion. If a limited amount of labeled data can be obtained, then we can use the resources to review the results and conduct refinement to the target domain MLN, MLN_t.

Therefore, we design an approach where unlabeled query predicates are actively selected by analyzing the difference between the inference results using the source domain MLN, MLN_s and the target domain MLN, MLN_t. The actively labeled query predicates are then utilized to filter potential formulae and to adjust the weight learning of the formulae for the target domain MLN, MLN_t.

6.1.1 The Proposed Algorithm

The outline of our proposed approach is depicted in Figure 6.1. The main idea of the proposed approach is to actively select ground query predicates for manual annotation by comparing the inference results of the target domain MLN, MLN_t, and the source domainMLN, MLN_s. At iteration j, given a source domain MLN, MLN_s^j, where the weights are learned using labeled data from the source domain in Step 4. We perform inference on the target domain unlabeled data to obtain the predicted results, that is the truth

# Pool-based Approach

INPUT:

MLN_s: An MLN for the source domain D_s

U_t: A set of unlabeled query predicates in target domain D_t

J: the number of iterations

F: a set of candidate formulae

N: the maximum number of query predicates selected for manual annotation in each iteration

OUTPUT: MLN_t: An MLN for D_t

Notations: $MLN(x)$: the prediction of truth values of the ground query predicate, x.

ALGORITHM:

```
$L_t \leftarrow \emptyset$; $L'_t \leftarrow \{MLN_s(y_i)\}$
$MLN_s^0 \leftarrow MLN_s$
for $j = 1$ to $J$
    Train $MLN_s^j$ with $L_s$
    Perform inference on $U_t$ using $MLN_s^j$
    for each formula $f_i \in F$
        compute $\delta(f_i)$
    end for
    Select a set of $N$ formulae, $F_j = \{f_{i1}, ..., f_{iN}\} \in F$ with highest $\delta(f_i)$
    $MLN_t^j \leftarrow MLN_s^i \cup F_j$
    Train $MLN_t^j$ with $L_t'$
    Perform inference on $U_t$ using $MLN_t^j$
    for all ground unlabeled query predicates $y_i | y_i \in U_t$
        compute $D(y_i)$
    end for
    Sample from $D$, a subset $N$ of $M$ query predicates from $U_t$.
    Label the query predicates in $N$
    $U_t \leftarrow U_t \setminus N$
    $L_t \leftarrow N$
    for each unlabeled query predicate $y_i$, $y_i' = MLN_t^j(y_i)$
        $L_t' \leftarrow L_t \cup y_i'$
    end for
    Train $MLN_t^j$ with $L_t'$
    $MLN_t^i \leftarrow$ filter the relatively less important formula from $F_j$
    $MLN_s^i \leftarrow MLN_t^j$
end for
Train $MLN_t^j$ with $L_t'$
$MLN_t \leftarrow MLN_t^J$
```

Figure 6.1: The Pool-based Approach.

values of the query predicates y_i in the target domain D_t, represented by $MLN_s(y_i)$ in Step 5. In the next step, we select some potential formulae f_i by evaluating the impact score $\delta(f_i)$ to the target domain dataset.

The impact score of a formula is defined as:

$$\delta(fi) = \sum_k |n_i(y[y_k = 0]) - n_i(y[y_k = 1])| \tag{6.1}$$

where $n_i(y[y_i = 1])$ refers to the number of true groundings of the i-th formula when we force the truth value of grounding predicate y_i to be 1, and for $n_i(y[y_i = 0])$, the truth value of y_i is forced to be 0.

Recall that inference with MLN can be done by calculating the probability that a predicate takes on a particular truth value using Equation 3.2. Suppose we add a formula to the MLN, the probability of a ground query predicate becomes:

$$P(X_l = x_l | MB(X_l)) \tag{6.2}$$

$$= \frac{exp(\sum_{i=1}^{F+1} w_i n_i(x))}{exp(\sum_{i=1}^{F+1} w_i n_i(x_{[X_l=0]})) + exp(\sum_{i=1}^{F+1} w_i n_i(x_{[X_l=1]}))}$$

$$= \frac{exp(\sum_{i=1}^{F} w_i n_i(x) + w_{F+1} n_{F+1}(x))}{exp(\sum_{i=1}^{F} w_i n_i(x_{[X_l=0]}) + w_{F+1} n_{F+1}(x_{[X_l=0]})) + exp(\sum_{i=1}^{F} w_i n_i(x_{[X_l=1]}) + w_{F+1} n_{F+1}(x_{[X_l=1]})}$$

As a result, the probabilities of the truth value of the ground predicate to be

1 or 0 are:

$$P(X_l = 1|MB(X_l)) \tag{6.3}$$

$$= \frac{1}{exp(\sum_{i=1}^{F} w_i n_i(x_{[X_l=0]}) - \sum_{i=1}^{F} w_i n_i(x_{[X_l=1]}))(e^{w_{F+1}n_{F+1}(x_{[X_l=0]})-w_{F+1}n_{F+1}(x_{[X_l=1]})}) + 1}$$

$$= \frac{1}{exp(\sum_{i=1}^{F} w_i[n_i(x_{[X_l=0]}) - n_i(x_{[X_l=1]})])(e^{w_{F+1}[n_{F+1}(x_{[X_l=0]})-n_{F+1}(x_{[X_l=1]})}]) + 1}$$

$$P(X_l = 0|MB(X_l)) \tag{6.4}$$

$$= \frac{1}{exp(\sum_{i=1}^{F} w_i[n_i(x_{[X_l=1]}) - n_i(x_{[X_l=0]})])(e^{w_{F+1}[n_{F+1}(x_{[X_l=1]})-n_{F+1}(x_{[X_l=0]})}]) + 1}$$

One of the factor affecting the probability of a certain truth for the predicate is the value of $e^{w_{F+1}[n_{F+1}(x_{[X_l=0]})-n_{F+1}(x_{[X_l=1]})]}$ or $e^{w_{F+1}[n_{F+1}(x_{[X_l=1]})-n_{F+1}(x_{[X_l=0]})]}$. As a result, we can assess the relative influence of a formula to the inference by evaluating the accumulated difference of the number of truth groundings of the $(F+1)$-th formula and forcing the truth values of the predicates to be 0 or 1, i.e. computing $\delta(f_{F+1})$. A formula f_i with larger $\delta(f_i)$ is more important to the MLN and is selected. These selected formulae are added to the target domain MLN, MLN_t^j. Weights are learned with the predicted and manual labeled truth values of the query predicates in the target domain. Inference on the target domain unlabeled data is performed to obtain the predicted results represented by $MLN_s(y_i)$ in Step 12. We then compare the inference results $MLN_s(y_i)$ and $MLN_t(y_i)$ and compute the score of selection for labeling for each unlabeled query predicate in Step 13.

The selection score for labeling a query predicates is defined as:

$$\tau(y_i) = \frac{1}{\theta(y_i) + 1} \tag{6.5}$$

$$\begin{aligned} \theta(y_i) &= \beta[\xi(MLN_s(y_i) = MLN_t(y_i))]\upsilon(MLN_t(y_i)) + \\ &\quad (1-\beta)[\xi(MLN_s(y_i) \neq MLN_t(y_i))]\upsilon(MLN_t(y_i)) \end{aligned} \tag{6.6}$$

where L_t is the set of labeled ground query predicates and U_t is the set of unlabeled query predicates. y_i is the i-th ground query predicate in the target domain data U_t. $MLN_t(y_i)$ and $MLN_s(y_i)$ are the predicted truth values of ground query predicate y_i using the target domain MLN, MLN_t, and the source domain MLN, MLN_s, respectively. $\xi(X) = 1$ if X is true. $\upsilon(MLN_t(y_i))$ represents the confidence of the prediction using the target domain MLN, MLN_t. One way to measure the confidence of the prediction by an MLN is to consider the conditional probability of the predication.

$$\upsilon(MLN_t(y_i)) = \max_{l=\{0,1\}}(P(MLN_t(y_i) = l|X_t)) \tag{6.7}$$

where X_t is the set of evidential predicates with known truth values in the target domain data. The higher the selection score of a query predicate, the higher the probability it is selected for manual annotation. Hence, the unlabeled ground query predicates are sampled with probability proportional to $\tau(y_i)$. The probability distribution of a query predicate to be selected for labeling is:

$$D(y_i) = \frac{\tau(y_i)}{\sum_{k=1}^{|U_t|} \tau(y_k)} \tag{6.8}$$

As a result, M query predicates are selected and the labels are requested and annotated in Step 17. For the remaining unlabeled query predicates, they are automatically labeled by the predicted truth values obtained using MLN_t^0 in Steps 20 to 22. The labeled query predicates in conjunction with the predicated labels of the unlabeled query predicates form the updated labeled data L_t'. Weights in MLN_t^0 are learned using the new set of labeled query predicates in L_t'. The logic formulae with exceptionally low weights are removed from MLN_t^0 as shown in Step 25. Finally, the refined MLN_t is obtained by learning with the predicated and manual labeled truth values of the query predicates in the target domain.

6.1.2 Experiments

We have conducted experiments on two different tasks, namely, pronoun resolution and segmentation of citations. We used the same datasets described in Chapter 5.3 for experiments. For each task, we prepared a source domain training dataset from which the source domain MLN is obtained. For the target domain, an unlabeled training dataset is prepared for adapting the source domain MLN to the target domain. Finally, a separate target domain testing dataset is used for evaluating the performance. Our approaches are also implemented based on the Alchemy system [34]. For pronoun resolution, documents are preprocessed and source domain MLN is constructed as described in Chapter 5.3.1. For segmentation of citations, documents are preprocessed and source domain MLNs used are described in Chapter 5.3.2. Similar to experiments in Chapter 5.3, resolution etiquette is used as the evaluation metric for the task of pronoun resolution, and F_1 measure is used for the task of segmentation of citations.

We compare the pool-based approach with a random sampling approach. In the random sampling approach, query predicates from the unlabeled target domain data is selected randomly for annotation. The labeled target domain data and the predicted labels of the target domain data using the source domain MLN, MLN_s, are used to learn the weights of the target domain MLN, MLN_t.

Both the pool-based approach and the random sampling approach use the same amount of actively selected labeled target domain data. For the tasks of pronoun resolution and segmentation of citation records, the amounts of selected query predicates are 6% and 10% of the total number of query predicates in the target domain, respectively. In the experiments for the pool-based approach, β is set to the value of 0.9.

Pronoun Resolution

Table 6.1 depicts the experimental results. The improvement of the pool-based approach over the random sampling approach demonstrates that our approach is able to select more informative query predicates and hence assist the capturing of differences between the source and the target domain models.

System	Accuracy
Random Sampling	18.5%
Pool-based Approach	25.2%

Table 6.1: Experimental Results of Pool-based Approach for Pronoun Resolution Using Limited Amount of Labeled Data

Segmentation of Citations

Table 6.2 shows the average refinement performance of the 12 datasets for the segmentation task. The pool-based approach shows consistent improvements over random sampling. The detailed results of each of the 12 datasets are shown in Table 6.3.

System	Average F-measure	p-value
Random Sampling	75.3%	–
Pool-based Approach	76.1%	0.0003

Table 6.2: Experimental Results of Pool-based Approach for Segmentation of Citation Records Using Limited Amount of Labeled Data

6.2 Theoretical Analysis of the Assignment Error for Pool-based Approach

As a query predicate is sampled according to the probability distribution in Equation 6.8, a query predicate y_i has a probability of $D(y_i)$ to be selected for manual annotation. Therefore, the probability of the query predicate not selected for manual annotation is $1 - D(y_i)$. When a query predicate is not selected for manual annotation, we entrust it with the predicted truth values deduced from the target domain MLN, MLN_t^j. Since the MLN_t^j is trained upon the set of truth values for the unlabeled query predicates U_t, it is subjected to the influence of the accuracy of the those assigned truth values.

Hence, to demonstrate the fitness of the refined MLN_t to the target domain, we analyze the influence of the truth values by deriving the error of the assigned truth values of the query predicates during the active learning

Source Domain	Target Domain	Random Sampling	Pool-based Approach
reasoning	constraint+face	77.0%	77.7%
reinforcement	constraint+face	72.7%	73.8%
face	constraint+reasoning	77.1%	78.3%
reinforcement	constraint+reasoning	75.2%	75.3%
face	constraint+reinforcement	77.2%	78.4%
reasoning	constraint+reinforcement	80.4%	80.6%
constraint	face+reasoning	71.2%	72.1%
reinforcement	face+reasoning	74.3%	74.7%
constraint	face+reinforcement	71.0%	71.7%
reasoning	face+reinforcement	78.8%	79.3%
constraint	reasoning+reinforcement	71.9%	73.0%
face	reasoning+reinforcement	76.6%	77.7%

Table 6.3: Experimental Results of Pool-based Approach for All Datasets of Segmentation of Citations Using Limited Amount of Labeled Data

process. Such assignment error has been investigated in some active learning approaches [4, 54] to demonstrate the effectiveness of the samples. With similar motivations, we analyze the error bound of the sample selection process.

Theorem 6.2.1 *Let ϵ_t and ϵ_s denote the expected error of the target domain MLN, MLN_t and the source domain MLN, MLN_s respectively, and N denote the number of ground query predicates in the target domain. The assignment error ϵ of the pool-based approach satisfies*

$$\epsilon \leq \begin{cases} \epsilon_s \epsilon_t \frac{\beta}{\beta(1-\epsilon_t)+1} & \text{if } MLN_t(y_i) = MLN_s(y_i), \\ (1-\epsilon_s)\epsilon_t \frac{(1-\beta)}{(1-\beta)(1-\epsilon_t)+1} & \text{if } MLN_t(y_i) \neq MLN_s(y_i). \end{cases} \tag{6.9}$$

Proof A query predicate will be assigned an incorrect truth value when it is not selected for sampling and based on the predicted truth values of the two MLNs, MLN_t and MLN_s. $v(MLN_t(y_i))$ represents the confidence of the predicted truth value $MLN_t(y_i)$. This probability depends on the accuracy of MLN_t.

$$v(MLN_t(y_i)) = 1 - \epsilon_t \tag{6.10}$$

In Equation 6.6, the sampling score is affected by whether the two predicted truth values, $MLN_t(y_i)$ and $MLN_s(y_i)$ obtained using MLN_t and MLN_s respectively, are the same or not. Hence, we deduce the assignment error in two cases:

1. $MLN_t(y_i) = MLN_s(y_i)$

2. $MLN_t(y_i) \neq MLN_s(y_i)$

Case 1: when $MLN_t(y_i) = MLN_s(y_i)$,

$\xi(MLN_t(y_i) = MLN_s(y_i)) = 1$ and $\xi(MLN_t(y_i) \neq MLN_s(y_i)) = 0$

$$\theta(y_i) = \beta \upsilon(MLN_t(y_i)) \tag{6.11}$$

$$\begin{aligned}
\epsilon &\leq \epsilon_s \epsilon_t (1 - D(y_i)) \\
&= \epsilon_s \epsilon_t (1 - \frac{\tau(y_i)}{\sum_{k=1}^{|U_t|} \tau(y_k)}) \\
&\leq \epsilon_s \epsilon_t (1 - \frac{\tau(y_i)}{|U_t|}) \\
&\leq \epsilon_s \epsilon_t (1 - \frac{\tau(y_i)}{N}) \\
&= \epsilon_s \epsilon_t \frac{1}{N} [N - \frac{1}{\beta(\upsilon(MLN_t(y_i))) + 1}] \\
&= \epsilon_s \epsilon_t \frac{1}{N} [N - \frac{1}{\beta(1 - \epsilon_t) + 1}] \\
&= \epsilon_s \epsilon_t [\frac{\beta(1 - \epsilon_t) + 1 - 1/N}{\beta(1 - \epsilon_t) + 1}] \\
&\leq \epsilon_s \epsilon_t [\frac{\beta}{\beta(1 - \epsilon_t) + 1}]
\end{aligned} \tag{6.12}$$

Case 2: when $MLN_t(y_i) \neq MLN_s(y_i)$,

$\xi(MLN_t(y_i) = MLN_s(y_i)) = 0$ and $\xi(MLN_t(y_i) \neq MLN_s(y_i)) = 1$

$$\theta(y_i) = (1 - \beta) \upsilon(MLN_t(y_i)) \tag{6.13}$$

$$
\begin{aligned}
\epsilon &\leq (1-\epsilon_s)\epsilon_t(1-D(y_i)) \\
&= (1-\epsilon_s)\epsilon_t(1-\frac{\tau(y_i)}{\sum_{k=1}^{|U_t|}\tau(y_k)}) \\
&\leq (1-\epsilon_s)\epsilon_t(1-\frac{\tau(y_i)}{|U_t|}) \\
&\leq (1-\epsilon_s)\epsilon_t(1-\frac{\tau(y_i)}{N}) \\
&= (1-\epsilon_s)\epsilon_t\frac{1}{N}[N-\frac{1}{(1-\beta)(\upsilon(MLN_t(y_i)))+1}] \\
&= (1-\epsilon_s)\epsilon_t\frac{1}{N}[N-\frac{1}{(1-\beta)(1-\epsilon_t)+1}] \\
&= (1-\epsilon_s)\epsilon_t[\frac{(1-\beta)(1-\epsilon_t)+1-1/N}{(1-\beta)(1-\epsilon_t)+1}] \\
&\leq (1-\epsilon_s)\epsilon_t[\frac{(1-\beta)}{(1-\beta)(1-\epsilon_t)+1}] \qquad (6.14)
\end{aligned}
$$

∎

6.3 Error-driven Approach

In the error-driven method, since the size of labeled data is also an important factor for learning, the limited amount of labeled target domain data still hinders the analysis of the target domain. Hence, obtaining an estimated labeled dataset by querying a small number of actively selected query predicates is an interesting issue towards knowledge base refinement. We investigate the use of estimated labeled dataset for the discovery of logic formulae when the manual annotation of a very small amount of data in the target domain is possible. The idea of our approach is that certain query predicates have similar behavior. By identifying groups of query predicates with similar characteristics, we can capture potential logic formulae from the target domain. Such clustering of query predicates can be obtained with the help of actively selected query predicates. Our method analyzes the unlabeled target domain data and actively asks the expert to provide labels (or answers) of a very small amount of automatically selected query predicates. With the actively selected query predicates, a suitable clustering of query predicates is obtained and estimate the labels for the entire data. Potential logic formulae in the unlabeled target domain data are discovered using the estimated labels.

Our error-driven approach consists of 2 components:

1. Clustered-based active learning

   An estimated set of labels is obtained by performing active learning with a hierarchical clustering of unlabeled query predicates.

2. Pattern discovery and filtering

   Potential patterns are discovered for the target domain from the unlabeled target domain data, particularly, the evidential predicates. We then filter out low confidence patterns using the estimated set of labels of query predicates.

6.3.1 Cluster-based Active Learning

Cluster structure has been shown to be beneficial to incorporate with active learning to improve performance [43, 19]. We develop a cluster-based active learning component inspired by the model in Dasgupta and Hsu [19]. The labels of the entire target domain unlabeled data are estimated by actively selecting query predicates for annotation. The outline of the cluster-based active learning is depicted in Figure 6.2. The rationale of this approach is that when a cluster of query predicates has a relatively high ratio of labeled query predicates with truth values equal to a particular label, it is assigned with that majority label and we move on to investigate on labeling query predicates from other clusters. A clustering with relatively pure cluster labels and distribution not deviate too far from the source domain MLN, MLN_s^*, can be obtained.

First, hierarchical clustering H of the target domain unlabeled query predicates is performed and a hierarchical tree (root) of clusters is obtained. Starting from a pruning $C = \{\nu_i\}$, which is a disjoint set of clusters in the hierarchical clustering H, a node ν_i from C is selected based on the probability distribution $D^*(\nu_i)$. The probability distribution is defined as Equation 6.22.

$$D^*(\nu_i) = \frac{w_{\nu_i}(1 - P^{UB}_{\nu_i,L^*(\nu_i)})(1 - R_{\nu_i,L^*(\nu_i)})}{\sum_{\nu_k \in C} w_{\nu_k}(1 - P^{UB}_{\nu_k,L^*(\nu_k)})(1 - R_{\nu_k,L^*(\nu_k)})} \tag{6.15}$$

where the probability of a node to be selected depends on three factors, the ratio of the size of the node denoted by $w_{\nu_i} = |\nu_i| / \sum_{\nu_k \in C} |\nu_k|$, the upper bound of how pure the node ν_i is in its labels denoted by $P^{UB}_{\nu_i,L^*(\nu_i)}$, as well as the agreement of the majority label between the current estimated label and the results of inference using the source domain MLN, MLN_s, denoted by $R_{\nu_i,L^*(\nu_i)}$.

$$P^{UB}_{\nu_i,L^*(\nu_i)} = \frac{N^{L^*(\nu_i)}_{\nu_i}}{N_{\nu_i}} + [\frac{1}{|\nu_i|} + \sqrt{\frac{(N^{L^*(\nu_i)}_{\nu_i}/N_{\nu_i})(1 - (N^{L^*(\nu_i)}_{\nu_i}/N_{\nu_i})}{(\nu_i)}}] \tag{6.16}$$

where $N^l_{\nu_i}$ is number of labeled query predicates in ν_i having the truth value equal to l, N_{ν_i} is the number of labeled query predicates in ν_i, and $L^*(\nu_i) = \underset{l}{\operatorname{argmax}} \; N^l_{\nu_i}$.

$$R_{\nu_i,L^*(\nu_i)} = \frac{\sum_k \xi(y'_k = L^*(\nu_i))}{|\nu_i|} \tag{6.17}$$

```
# Cluster-based Active Learning

INPUT:
    $U_t$: A set of unlabeled query predicates in target domain $D_t$
    $J$: the number of iterations
    $N$: the maximum number of query predicates selected for manual annotation

OUTPUT: $L_t^*$: Estimated labels for $U_t$

Notations:
    $L(C_i)$: the label of the class $C_i$ of query predicates
    $\nu_i$: a node in the hierarchical clustering, $H$
    $P^{UB}(x)$: the upper bound of probability $P(x)$
    $P_{\nu_i,l}(j)$: the probability of labeled predicates in node $\nu_i$ with label $l$
    $\mu(\nu, l)$: the number of query predicates in $\nu$ having label l

ALGORITHM:
    $H = \text{root}, \nu_i, ... \leftarrow$ perform hierarchical clustering on the set of query predicates $U_t$
    $C \leftarrow \text{root}$; $L(root) \leftarrow 1$
    for $j = 1$ to $J$
        for $i = 1$ to $N$
            $\nu_i \leftarrow$ select a node $\nu_i \in C$ by the probability distribution $D^*(\nu_i)$
            $V \leftarrow V \cup \nu_i$
            for each query predicate $y_k | y_k \in U_t$ from subtree $T_{\nu_i}$
                compute $\eta(y_k)$
        end for
        $y_{ij} \leftarrow$ select the query predicate from $U_t$ with $\max(\{\eta(y_k)\})$
        label $y_{ij}$
        $U_t \leftarrow U_t \setminus y_{ij}$
        $L_t^* \leftarrow L_t^* \cup y_{ij}$
    end for
    for each $\nu_i \in V$
        $C' \leftarrow$ the pruning having scores $s(\nu_i)$
        $C \leftarrow C \setminus V \cup C'$
        for each $\nu_k \in C'$
            $L(\nu_k) \leftarrow \underset{l}{\operatorname{argmax}}\ \mu(\nu_k, l)$
        end for
    end for
end for
for each $\nu_k \in C$
    for each query predicate $y_k | y_k \in \nu_k$
        $y_k \leftarrow L(\nu_k)$
        $L_t^* \leftarrow L_t^* \cup y_k$
    end for
end for
```

Figure 6.2: The Cluster-based Active Learning Approach.

where y'_k is the predicted truth value of the k-th query predicate y_k with the source domain MLN, MLN_s, $\xi(y'_k = L^*(\nu_i)) = 1$ if the truth value y'_k is equal to the majority label, $L^*(\nu_i)$, of node ν_i.

Then, in Steps 7 and 8, an unlabeled query predicate y_j is selected by Equation 6.18.

$$\eta(y_j) = (1 - max_l(P(y'_j = l|\text{MLN}_s, U_t))) \tag{6.18}$$

where y'_j is the predicted truth value of the query predicate y_j with MLN_s. $P(y'_j = l|\text{MLN}_s, U_t)$ is the probability of the predicted truth values. $1 - max_l(P(y'_j = l|\text{MLN}_s, U_t)$ is the uncertainty of the prediction made by MLN_s.

Regarding giving manual annotation to some selected query predicates, a pruning is updated and selected by accessing the improvement in the estimated error of majority labeling based on $s(\nu_i)$ in Equation 6.19.

$$s(\nu_i) = \frac{1}{w_{\nu_i}} \sum_{\nu_j \in \nu_i} w_{\nu_j} (1 - \frac{N_{\nu_l}^{L^*(\nu_j)}}{N_{\nu_j}}) \tag{6.19}$$

Finally, the pruning is updated and each query predicate in the cluster in the current pruning is assigned its majority truth value of the actively labeled query predicates.

6.3.2 Pattern Discovery and Filtering

The estimated truth values of the query predicates together with the observations, that are the evidential predicates, in the target domain D_t, we discover the potential patterns. Potential patterns are extracted with the candidate pattern identification component as presented Chapter 5.2.1. We extract connected paths containing the ground query predicates. Next, we filter the un-informative patterns. The rationale of our filtering approach is that a candidate pattern can belong to one of the following cases given the clustering structure and the estimated truth values of the target domain data:

1. the pattern appears in many clusters and its error rate is high;
2. the pattern appears in many clusters and its error rate is low;
3. the pattern appears in only one or a few clusters and the error rate is high;
4. the pattern appears in only one or a few clusters and the error rate is low.

Candidate patterns of case 2 are the patterns that would be beneficial to the target domain. Hence, in our approach, we would like to retain patterns of case 2 and filtered out the others. Hence, one way to solve the situation is to measure the error rate of a pattern by Equation 6.20. Assuming the pattern could be regarded as a formula in form of $p \rightarrow q$:

$$\text{error}(f) = N(f)/N(p) \tag{6.20}$$

where $N(f)$ is the number of true groundings of f given the truth values of predicates in formula f and $N(p)$ is the number of true groundings of p. More specifically, we computed the upper bound of the error rate, $\text{error}_{UB}(f)$, defined in Equation 6.21. to obtain the pessimistic uncertainty of the formulae, such that only the most confident formulae will be retained.

$$\text{error}(f) + 1.96\sqrt{\frac{\text{error}(f)(1 - \text{error}(f))}{n}} \tag{6.21}$$

where $n = N(f)$. The filtering process is described in Figure 6.3.

In Steps 7 to 12, we select the formulae such that more generalized rules are preferred unless the relatively more specific formula is more accurate. Finally, the selected set of formula are added to MLN_t' and weights are learned to construct the refined target domain MLN, MLN_t.

6.3.3 Experiments

We have conducted experiments on two different tasks, namely, pronoun resolution and segmentation of citations. We used the same datasets described in Chapter 5.3 for experiments. For each task, we prepared a source domain training dataset from which the source domain MLN is obtained. For the target domain, an unlabeled training dataset is prepared for adapting the source domain MLN to the target domain. Finally, a separate target domain testing dataset is used for evaluating the performance. Our approaches are

```
# Pattern Filtering

INPUT:

    $MLN_s$: An MLN for $D_s$

    $E_t$: A set of evidential predicates in target domain $D_t$

    $Y_t$: A set of predicted truth values of the query predicates in target domain $D_t$

    $F$: a set of candidate formulae

    $\alpha$: the threshold of error rate

OUTPUT:

    $MLN_t$: An MLN for $D_t$

ALGORITHM:

for each $f_i \in F$
    compute $\text{error}_{UB}(f_i)$
    if $(\text{error}_{UB}(f_i) < \alpha)$
        $F' \leftarrow F' \cup f_i$
end for
```

```
for each $f_i \in F'$
    for each $f_j \in F'$
        if $(f_i \in f_j)$ and $(\text{error}_{UB}(f_i) < \text{error}_{UB}(f_i))$
                $F' \leftarrow F' \setminus f_j$
    end for
end for
$\text{MLN}'_t \leftarrow \text{MLN}_s \cup F'$
$\text{MLN}_t \leftarrow$ perform formula weight adaptation(Chapter 5) on $\text{MLN}'_t$
```

Figure 6.3: The Pattern Filtering Process.

also implemented based on the Alchemy system [34]. For pronoun resolution, documents are preprocessed and source domain MLN is constructed as described in Chapter 5.3.1. For segmentation of citations, documents are preprocessed and source domain MLN used are described in Chapter 5.3.2. Similar to experiments in Chapter 5.3, resolution etiquette is used as the evaluation metric for the task of pronoun resolution, and F_1 measure is used for the task of segmentation of citations.

We first compare the error-driven active learning approach with a random sampling approach. In the random sampling approach, query predicates from the unlabeled target domain data is selected randomly for annotation. The labeled target domain data and the predicted labels of the target domain data using the source domain MLN, MLN_s, are used to learn the weights of the target domain MLN, MLN_t. We also implemented the error-driven approach that randomly selects a limited amount of query predicates instead of using the cluster-based active learning component.

The same amount of selected labeled target domain data are used in all the experiments. For both the error-driven approach and the error-driven active learning approach, we use the penalty-based formula weight adaptation to learn the weights for the target domain MLN, MLN_t.

For both tasks, the amount of selected query predicates is within 5% of the total number of query predicates in the target domain. In both error-driven approach and the error-driven active learning approach, α is set to the value of 0.1 to filter out candidate formulae with estimated error rates above 0.1.

Pronoun Resolution

Table 6.4 depicts the experimental results. The improvement of the error-driven active learning approach over the error-driven approach demonstrates that the estimated labels of the cluster-based active learning component is beneficial towards formula filtering. They enable the selection of a broader but still fairly accurate formulae for the target domain.

System	Accuracy
Random Sampling	16.2%
Error-driven	32.0%
Error-driven Active Learning	37.0%

Table 6.4: Experimental Results of Error-driven Active Learning Approach for Pronoun Resolution Using Limited Amount of Labeled Data

Segmentation of Citations

Table 6.5 shows our average refinement performance of the 12 datasets for the segmentation task. The error-driven active learning approach shows consistent improvements over random sampling and error-driven approach. In average, the error-driven active learning approach outperforms the error-driven approach by 1.2%. Among 11 of 12 datasets, the error-driven active learning approach outperforms the error-driven approach. We conducted statistical significance test using McNemar's paired t-tests over the 12 datasets and the p-value is 0.004. The error-driven active learning approach is found to be statistically better that error-driven approach (p-value < 0.05 with a 95% confidence interval).

System	Average F-measure	p-value
random sampling	74.5%	–
error-driven	76.7%	–
error-driven active learning	78.0%	0.004 (compared with error-driven)

Table 6.5: Experimental Results of Error-driven Active Learning Approach for Segmentation of Citation Records Using Limited Amount of Labeled Data

The detailed results of each of the 12 datasets are shown in Table 6.6.

Source Domain	Target Domain	Random Sampling	Error-driven	Error-driven active learning
reasoning	constraint+face	76.2%	77.9%	78.7%
reinforcement	constraint+face	72.4%	74.1%	76.2%
face	constraint+reasoning	76.4%	79.4%	77.6%
reinforcement	constraint+reasoning	74.6%	75.6%	77.1%
face	constraint+reinforcement	76.0%	79.3%	79.7%
reasoning	constraint+reinforcement	79.4%	81.6%	83.1%
constraint	face+reasoning	70.6%	72.3%	74.6%
reinforcement	face+reasoning	73.8%	74.7%	76.4%
constraint	face+reinforcement	70.1%	72.5%	74.7%
reasoning	face+reinforcement	78.3%	80.7%	81.5%
constraint	reasoning+reinforcement	71.4%	73.8%	77.1%
face	reasoning+reinforcement	75.3%	78.2%	79.1%

Table 6.6: Experimental Results of Error-driven Active Learning Approach for All Datasets of Segmentation of Citations Using Limited Amount of Labeled Data

6.4 Analytical Study for the Error-driven Approach

We perform an analysis on the error of the estimated label for the query predicates from the cluster-based active learning approach. Let ϵ be the error of the estimated label and $D(\nu_i)$ represent the probability distribution of a node to be selected.

$$D(\nu_i) = \frac{w_{\nu_i}(1 - P_{\nu_i,L^*(\nu_i)})(1 - R_{\nu_i,L^*(\nu_i)})}{\sum_{\nu_k \in C} w_{\nu_k}(1 - P_{\nu_k,L^*(\nu_k)})(1 - R_{\nu_i,L^*(\nu_i)})} \tag{6.22}$$

where $P_{\nu_i,L^*(\nu_i)}$ represents the ratio of labeled query predicates in node ν_i having the truth values $L^*(\nu_i)$, w_{ν_k} is the ratio of labeled query predicates in node ν_i. and $R_{\nu_i,L^*(\nu_i)}$ is defined in Equation 6.17. Essentially, $R_{\nu_i,L^*(\nu_i)}$ can be interpreted as the agreement of the majority label between the current estimated labels and the results of inference using the source domain MLN, MLN_s.

Since the truth value for each query predicate can either be positive (+) or negative (-), the error of the estimated label for the query predicate in iteration j can be as follows: A query predicate whose truth value is actually negative (or positive) but the node it belongs is not selected for manual annotation and the majority label of the node is positive (or negative). This error is denoted by ϵ^+ (or ϵ^-).

Let $\epsilon^{l}_{\nu_k}$ is the expected error of the source domain MLN_s in predicting the truth value of the query predicates in node ν_k to be the truth value l, and $\epsilon^{\bar{l}}_{\nu_k}$ is the expected error of the source domain MLN, MLN_s, in predicting the truth value of the query predicates in node ν_k not to be l. Then, $(1-\epsilon_{\nu_i}^{\overline{L^{\bullet}(\nu_j)}}) = (1-P^{UB}_{\nu_i,L^{\bullet}(\nu_i)})(1-R_{\nu_i,L^{\bullet}(\nu_i)})$ represent the probability that the truth value of a query predicate is estimated to be opposite to $L^{*}(\nu_j)$ and the predicated truth value by MLN_s agrees with the truth value. ϵ^{+} can be expressed as follows:

$$\begin{aligned}
\epsilon^{+} &= P_{\nu_i,-}(1-D(\nu_i)) \\
&= P_{\nu_i,-}(1-\frac{w_{\nu_i}(1-P_{\nu_i,-})(1-R_{\nu_i,-})}{\sum_{\nu_k\in C} w_{\nu_k}(1-P_{\nu_k,L^{\bullet}(\nu_k)})(1-R_{\nu_i,L^{\bullet}(\nu_k)})}) \\
&= P_{\nu_i,-}(\frac{\sum_{\nu_k\in C\backslash\nu_i} w_{\nu_k}(1-P_{\nu_k,L^{\bullet}(\nu_k)})(1-R_{\nu_i,L^{\bullet}(\nu_k)})}{\sum_{\nu_k\in C} w_{\nu_k}(1-P_{\nu_k,L^{\bullet}(\nu_k)})(1-R_{\nu_i,L^{\bullet}(\nu_k)})}) \\
&= P_{\nu_i,-}[\frac{\sum_{\nu_k\in C\backslash\nu_i} w_{\nu_k}(1-\epsilon_{\nu_k}^{\overline{L^{\bullet}(\nu_k)}})}{\sum_{\nu_k\in C} w_{\nu_k}(1-\epsilon_{\nu_k}^{\overline{L^{\bullet}(\nu_k)}})}] \qquad (6.23)
\end{aligned}$$

Hence, the probability that MLN_s correctly predicts the truth value for the query predicates in node ν_j to be the label l is $(1-\epsilon^{l}_{\nu_j})$. Therefore, if MLN_s does not agree with the majority label of the current clustering C except ν_i, $\sum_{\nu_k\in C\backslash\nu_i} w_{\nu_i}(1-\epsilon_{\nu_j}^{\overline{L^{\bullet}(\nu_j)}})$ increases, the error ϵ^{+} increases.

One the other hand, if MLN_s agrees with the the majority label of the current clustering C except ν_i, $\sum_{\nu_k\in C\backslash\nu_i} w_{\nu_i}(1-\epsilon_{\nu_j}^{\overline{L^{\bullet}(\nu_j)}})$ decreases, the error ϵ^{+} decreases.

In our experiments, the probability distribution of the purity ratio of the labeled data can be described by:

$$P_{\nu_i,L^\bullet(\nu_i)}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} exp^{-(x-\mu)^2/2\sigma^2} \tag{6.24}$$

where $\mu = \frac{N_{\nu_i}^-}{N_{\nu_i}}$ and $\sigma = \sqrt{\frac{(N_{\nu_i}^{L^\bullet(\nu_i)}/N_{\nu_i})(1-(N_{\nu_i}^{L^\bullet(\nu_i)}/N_{\nu_i})}{|\nu_i|}}$. We employed the upper bound $P^{UB}_{\nu_i,L^\bullet(\nu_i)}$ to replace $P_{\nu_i,L^\bullet(\nu_i)}$ in $D(\nu_i)$, i.e. $D^\bullet(\nu_i)$ shown in Equation 6.22. This can prevent the cluster-based active learning approach in Figure 6.2 from labeling small clusters and leaf nodes.

Chapter 7

Conclusions

In this thesis, we investigate techniques for refining an existing knowledge base, in particular, MLN, from a source domain to a target domain. We tackle MLN refinement under two situations, namely, using unlabeled target domain data, and using a limited amount of labeled target data. For the situation of using unlabeled target domain data, our proposed framework consists of two components, namely, formula weight adaptation, and logic formula refinement. For formula weight adaptation, we design two approaches to jointly maximizing the likelihood of the target domain observations and capturing the differences of the source and the target domains. The first approach captures the difference by measuring the distribution divergence between the two domains. The second approach incorporates a penalized degree of difference between the source and the target domain data. For logic formula refinement, we discover new logic formulae specific to the target domain. For the situation of using a limited amount of labeled target domain data, we develop two approaches to actively select target domain unlabeled data for annotation to refine the existing MLN. In the first approach, a pool-based active learning selects data for annotation based on the difference between the source and the target domains. A theoretical analysis on the assignment error of the data selection process is conducted. In the second approach, the error-driven active learning estimates the labels for the target domain data and performs logic formula selection. Experimental results on two text mining tasks showing consistent improvements demonstrate that the refined knowledge base can capture the differences between the target domain and the source domain. The refined MLN can better characterize the target domain in either situations of using unlabeled target domain data and using a limited amount of labeled target data.

One future direction is to apply our framework over different applications such as sequence labeling. We would like to investigate and evaluate our model in richer domains, and situations when more than one source domain are provided. We also plan to develop more advanced methods in utilizing the limited amount of labeled data in the refinement process. At present, we exploit the actively labeled data for discovering target specific queries and hence new formulae. We would also like to investigate the possible hidden information behind the small amount of labeled data for revising existing formulae and also further enhancing the refined MLN for the target domain.

Bibliography

[1] S. Bergsma and D. Lin. Bootstrapping path-based pronoun resolution. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 33–40, 2006.

[2] A. Beygelzimer, S. Dasgupta, and J. Langford. Importance weighted active learning. In *Proceedings of the 26th International Conference on Machine Learning*, pages 49–56, 2009.

[3] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondance learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 120–128, 2006.

[4] R. Castro and R. Nowak. Chapter 8: Active learning and sampling. *Foundations and Applications of Sensor Management*, 2007.

[5] K. Chan and W. Lam. Extracting causation knowledge from natural language texts. *International Journal of Intelligent Systems*, 20(3):327 – 358, 2005.

[6] K. Chan and W. Lam. Pronoun resolution with markov logic networks.

In *Proceedings of the 4th Asia Information Retrieval Symposium*, pages 153–164, 2008.

[7] K. Chan, W. Lam, and T.L. Wong. Accomodating new relations for e-business text mining applications. In *Proceedings of the First International Conference on Networking and Distributed Computing*, 2010.

[8] K. Chan, W. Lam, and T.L. Wong. Knowledge base refinement using limited amount of efforts from experts. *International Journal of Knowledge-Based Organizations*, submitted.

[9] K. Chan, W. Lam, and X. Yu. Coreference resolution using expressive logic models. In *Proceedings of the ACM 17th Conference on Information and Knowledge Management*, pages 1373–1374, 2008.

[10] K. Chan, T.L. Wong, and W. Lam. Adapting relational logic models using unlabeled data. In *Proceedings of the International Conference on Machine Learning and Cybernetics*, 2010.

[11] K. Chan, T.L. Wong, and W. Lam. Discovery of logic relations for text mining adaptation using unlabeled data. In *Proceedings of the 9th international conference on Adaptivity, Personalization and Fusion of Heterogeneous Information*, 2010.

[12] K. Chan, T.L. Wong, and W. Lam. Logic relation refinement using unlabeled data. In *Proceedings of the 2010 International Conference of Computational Intelligence and Intelligent Systems*, 2010.

[13] Y. S. Chan and H. T. Ng. Domain adaptation with active learning for word sense disambiguation. In *Proceedings of 45th Annual Meeting of the Association for Computational Linguistics*, pages 49–56, 2007.

[14] E. Charnial and M. Elsner. Em works for pronoun anaphora resolution. In *Proceedings of the 12th Conference of the European Chapter of the Association of Computational Linguistics*, pages 148–156, 2009.

[15] C. Chelba and A. Acero. Adaptation of maximum entropy capitalizer: Little data can help a lot. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 285–292, 2004.

[16] C. Cherry and S. Bergsma. An expectation maximization approach to pronoun resolution. In *Proceedings of the 9th Conference on Computational Natural Language Learning*, pages 88–95, 2005.

[17] D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, 1994.

[18] C. Cortes and V.N. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[19] S. Dasgupta and D. Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 208–215, 2008.

[20] H. Daumé III. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, 2007.

[21] H. Daumé III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26, 2006.

[22] J. Davis and P. Domingos. Deep transfer via second-order markov logic. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pages 217–224, 2008.

[23] P. Denis and J. Baldridge. A ranking approach to pronoun resolution. In *Proceedings of the 20th International Joint Conference on Artifical intelligence*, pages 1588–1593, 2007.

[24] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel, and R. Weischedel. The Automatic Content Extraction (ACE) Program–Tasks, Data, and Evaluation. *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 837–840, 2004.

[25] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 363–370, 2005.

[26] J. R. Finkel and C. D. Manning. Hierarchical bayesian domain adaptation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 602–610, 2009.

[27] H. L. Guo, H. Zhu, Z. Guo, X. Zhang, X. Wu, and Z. Su. Domain adaptation with latent semantic association for named entity recognition. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 281–289, 2009.

[28] S. C. H. Hoi, R. Jin, and M. R. Lyu. Large-scale text categorization by batch mode active learning. In *Proceedings of the International Conference on the World Wide Web*, pages 633–642, 2006.

[29] E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel. Ontonotes: The 90% solution. In *Proceedings of Human Language Tech-*

nology Conference/North American Chapter of the Association for Computational Linguistics Annual Meeting, Companion Volume: Tutorial Abstracts, pages 11–12, 2006.

[30] J. Jiang and C. Zhai. Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 264–271, 2007.

[31] R.D. King, K.E. Whelan, F.M. Jones, P.G. Reiser, C.H. Bryant, S.H. Muggleton, D.B. Kell, and S.G. Oliver. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature*, 427(6971):247–252, 2004.

[32] S. Kok and P. Domingos. Learning the structure of markov logic networks. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 441–448, 2005.

[33] S. Kok and P. Domingos. Learning markov logic network structure via hypergraph lifting. In *Proceedings of the 26th International Conference on Machine Learning*, pages 505–512, 2009.

[34] S. Kok, P. Singla, M. Richardson, M. Sumner, and H. Poon. The alchemy system for statistical relational ai. In *http://alchemy.cs.washington.edu/*, 2006.

[35] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probablistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, 2001.

[36] S. Lawrence, K. Bollacker, and C. L. Giles. Autonomous citation matching. In *Proceedings of the 3rd International Conference on Autonomous Agents*, pages 392–393, 1999.

[37] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th ACM International Conference on Research and Development in Information Retrieval*, pages 3–12, 1994.

[38] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

[39] P. Melville and R. Mooney. Diverse ensembles for active learning. In *Proceedings of the 21st International Conference on Machine Learning*, pages 584–591, 2004.

[40] L. Mihalkova, T. Huynh, and R. J. Mooney. Mapping and revising markov logic networks for transfer learning. In *Proceedings of the 22nd National Conference on Artificial Intelligence – Volume 1*, pages 608–614, 2007.

[41] L. Mihalkova and R. J. Mooney. Transfer learning by mapping with minimal target data. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence, Workshop on Transfer Learning for Complex Tasks*, 2008.

[42] R. Mitkov, R. Evans, and C. Orasan. A new, fully automatic version of mitkov's knowledge-poor pronoun resolution method. In *Proceedings of the 3rd International Conference on Computational Linguistics and Intelligent Text Processing*, pages 168–186, 2002.

[43] H. T. Nguyen and A. Smeulders. Active learning using pre-clustering. In

Proceedings of the 21st International Conference on Machine Learning, pages 79–86, 2004.

[44] S. Pan, J. Kwok, and Q. Yang. Transfer learning via dimensionality reduction. In *Proceedings of the 23rd AAAI conference on Artification Intelligence - Volume 2*, pages 677–682, 2008.

[45] H. Poon and P. Domingos. Joint inference in information extraction. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, pages 913–918, 2007.

[46] S. S. Pradhan, L. Ramshaw, R. Weischedel, J. MacBride, and L. Micciulla. Unrestricted coreference: Identifying entities and events in ontonotes. In *Proceedings of the International Conference on Semantic Computing*, pages 446–453, 2007.

[47] H. Raghavan, O. Madani, and R. Jones. Active learning with feedback on both features and instances. *Journal of Machine Learning Resarch*, 7:1655–1686, 2006.

[48] P. Rai, A. Saha, H. Daumé III, and S. Venkatasubramanian. Domain adaptation meets active learning. In *Proceedings of Workshop on Active Learning For NLP (in conjunction with NAACL-HLT)*, 2010.

[49] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th International Conference on Machine Learning*, pages 759–766, 2007.

[50] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006.

[51] N. Roy and A. McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the International Conference on Machine Learning*, pages 441–448, 2001.

[52] B. Settles and M. Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1069–1078, 2008.

[53] H.S. Seung, M. Opper, and H. Sompolinsky. Query by committe. In *Proceedings of the ACM Workshop on Computational Learning Theory*, pages 287–294, 1992.

[54] X. Shi, W. Fan, and J. Ren. Actively transfer domain knowledge. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 324–357, 2008.

[55] P. Singla and P. Domingos. Memory-efficient inference in relational domains. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 488–493, 2006.

[56] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the 9th ACM International Conference on Multimedia*, pages 107–118, 2001.

[57] S. Tong and D. Koller. Support vector machine active learning with appliations to text classification. In *Proceedings of 17th International Conference on Machine Learning*, pages 999–1006, 2000.

[58] K. Toutanova and C. D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of*

Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Copora, pages 63–70, 2000.

[59] C. Zhang and T. Chen. An active learning framework for content-based information retrieval. *IEEE Trans. on Multimedia*, 4:260–268, 2002.

[60] E. Zhong, W. Fan, J. Peng, and K. Zhang. Cross domain distribution adaptation via kernal mapping. In *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining KDD'09*, pages 1027–1036, 2009.

Appendix A

Detailed Experimental Results for Using Unlabeled Target Domain Data for Segmentation of Citation Records

This appendix demonstrates the detailed experimental results of each dataset for Segmentation of Citation Records. Performances for each field, namely, author, title, and venue are shown.

Source	Target	model	overall	author	title	venue
reasoning	constraint+face	SR2LR	77.4%	79.0%	67.8%	82.0%
reasoning	constraint+face	distribution	78.0%	78.9%	68.2%	83.0%
reasoning	constraint+face	penalty-based	78.2%	78.1%	69.7%	82.9%
reasoning	constraint+face	complete model	78.4%	78.6%	70.3%	83.0%
reinforcement	constraint+face	SR2LR	73.2%	72.9%	59.3%	80.6%
reinforcement	constraint+face	distribution	78.0%	78.9%	68.2%	83.0%
reinforcement	constraint+face	penalty-based	73.4%	74.5%	58.6%	80.6%
reinforcement	constraint+face	complete model	74.1%	74.4%	60.2%	81.2%
face	constraint+reasoning	SR2LR	73.8%	62.5%	77.2%	79.0%
face	constraint+reasoning	distribution	76.5%	74.9%	68.7%	81.1%
face	constraint+reasoning	penalty-based	78.5%	79.6%	69.8%	82.6%
face	constraint+reasoning	complete model	78.9%	80.3%	69.6%	83.2%
reinforcement	constraint+reasoning	SR2LR	75.3%	77.2%	61.0%	81.3%
reinforcement	constraint+reasoning	distribution	76.0%	77.4%	61.6%	82.2%
reinforcement	constraint+reasoning	distribution	75.7%	77.2%	60.9%	82.0%
reinforcement	constraint+reasoning	complete model	76.3%	77.4%	62.5%	82.4%
face	constraint+reinforcement	SR2LR	74.3%	61.6%	72.9%	79.1%
face	constraint+reinforcement	distribution	76.6%	73.9%	70.6%	80.9%
face	constraint+reinforcement	penalty-based	79.1%	80.3%	71.3%	83.0%
face	constraint+reinforcement	complete model	79.2%	79.7%	71.8%	83.2%
reasoning	constraint+reinforcement	SR2LR	80.9%	82.0%	75.2%	83.7%
reasoning	constraint+reinforcement	distribution	81.2%	83.2%	75.2%	83.8%
reasoning	constraint+reinforcement	penalty-based	81.4%	82.8%	75.8%	84.1%
reasoning	constraint+reinforcement	complete model	81.4%	82.3%	76.4%	83.9%
constraint	face+reasoning	SR2LR	72.0%	64.4%	61.2%	80.9%
constraint	face+reasoning	distribution	73.4%	68.6%	62.5%	81.2%
constraint	face+reasoning	penalty-based	72.5%	64.6%	62.2%	81.1%
constraint	face+reasoning	complete model	72.6%	65.1%	61.9%	81.4%
reinforcement	face+reasoning	SR2LR	74.4%	76.5%	59.9%	80.4%
reinforcement	face+reasoning	distribution	74.9%	78.2%	59.9%	80.7%
reinforcement	face+reasoning	penalty-based	74.3%	77.1%	59.0%	80.5%
reinforcement	face+reasoning	complete model	74.7%	77.3%	59.8%	80.6%
constraint	face+reinforcement	SR2LR	72.2%	65.9%	61.0%	80.9%
constraint	face+reinforcement	distribution	73.2%	69.4%	61.6%	81.2%
constraint	face+reinforcement	penalty-based	72.5%	66.0%	61.2%	81.4%
constraint	face+reinforcement	complete model	72.8%	66.3%	62.1%	81.6%
reasoning	face+reinforcement	SR2LR	79.8%	81.8%	72.5%	83.2%
reasoning	face+reinforcement	distribution	80.4%	82.9%	73.1%	83.4%
reasoning	face+reinforcement	penalty-based	80.3%	82.4%	73.5%	83.4%
reasoning	face+reinforcement	complete model	80.5%	82.5%	73.4%	83.7%
constraint	reasoning+reinforcement	SR2LR	73.5%	68.9%	61.7%	81.3%
constraint	reasoning+reinforcement	distribution	74.3%	72.6%	62.4%	81.0%
constraint	reasoning+reinforcement	penalty-based	73.8%	68.2%	62.8%	81.4%
constraint	reasoning+reinforcement	complete model	73.9%	69.0%	62.8%	81.4%
face	reasoning+reinforcement	SR2LR	72.8%	63.3%	69.6%	77.4%
face	reasoning+reinforcement	distribution	75.9%	74.5%	68.3%	80.3%
face	reasoning+reinforcement	penalty-based	78.2%	82.0%	68.3%	81.7%
face	reasoning+reinforcement	complete model	78.3%	76.7%	68.8%	82.0%

Table A.1: Detailed Experimental Results in F_1 Measure of Using Unlabeled Target Domain Data for All Datasets of Segmentation of Citation Matching