

A Method of Recognizing Entity and Relation

Xinghua Fan^{1,2} and Maosong Sun¹

¹ State Key Laboratory of Intelligent Technology and Systems,
Tsinghua University, Beijing 100084, China

fanxh@tsinghua.org.cn, sms@mail.tsinghua.edu.cn

² State Intellectual Property Office of P.R. China, Beijing, 100088, China

Abstract. The entity and relation recognition, i.e. (1) assigning semantic classes to entities in a sentence, and (2) determining the relations held between entities, is an important task in areas such as information extraction. Subtasks (1) and (2) are typically carried out sequentially, but this approach is problematic: the errors made in subtask (1) are propagated to subtask (2) with an accumulative effect; and, the information available only in subtask (2) cannot be used in subtask (1). To address this problem, we propose a method that allows subtasks (1) and (2) to be associated more closely with each other. The process is performed in three stages: firstly, employing two classifiers to do subtasks (1) and (2) independently; secondly, recognizing an entity by taking all the entities and relations into account, using a model called the Entity Relation Propagation Diagram; thirdly, recognizing a relation based on the results of the preceding stage. The experiments show that the proposed method can improve the entity and relation recognition in some degree.

1 Introduction

The entity and relation recognition, i.e. assigning semantic classes (e.g., person, organization and location) to entities in a sentence and determining the relations (e.g., born-in and employee-of) that hold between entities, is an important task in areas such as information extraction (IE) [1] [2] [3] [4], question answering (QA) [5] and story comprehension [6]. In a QA system, many questions concern the specific entities in some relations. For example, the question that “Where was Poe born?” in TREC-9 asks for the location entity in which Poe was born. In a typical IE task in constructing a job database from unstructured texts, the system are required to extract many meaningful entities like titles and salary from the texts and to determine how these entities are associated with job positions.

The task of recognizing entity and relation is usually treated as two separate subtasks carried out sequentially: (1) to recognize entities using an entity recognizer, and (2) to determine the relations held between them. This approach has two shortcomings. Firstly, the errors made in subtask (1) will be propagated to subtask (2) with an accumulative effect, leading to a loss in performance of relation recognition. For example, if “Boston” is mislabeled as a person, it will never have chance to be classified as the location of Poe’s birthplace. Secondly, the information available only in

subtask (2) cannot be used for subtask (1). For example, if we feel difficult to determine whether the entity X is a person or not, but we can determine that there exists a relation born-in between X and China easily, it is obvious that we can claim that X must be a person.

To address the problems described above, this paper presents a novel approach which allows subtasks (1) and (2) to be linked more closely together. The process is separated into three stages. Firstly, employing two classifiers to perform subtasks (1) and (2) independently. Secondly, recognizing an entity by taking all the entities and relations into account using a particularly designed model called the Entity Relation Propagation Diagram. And, thirdly, recognizing a relation based on the results of the preceding step.

The rest of the paper is organized as follows. Section 2 defines the problem of entity and relation recognition in a formal way. Section 3 describes the proposed method of recognizing entity and relation. Section 4 gives the experimental results. Section 5 is the related work and comparison. Section 6 is conclusions.

2 The Problem of Entity and Relation Recognition

Conceptually, the entities and relations in a sentence can be viewed, while taking account of the mutual dependencies among them, as a labeled graph in Fig. 1.

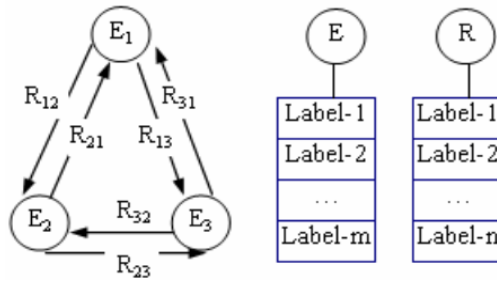


Fig. 1. Concept view of the entities and relations among them

In Fig.1, a node represents an entity and a link denotes the relation held between two entities. The arrowhead of a link represents the direction of the relation. Each entity or relation has several attributes, which are structured as a list of the node or the edge. These attributes can be classified into two classes. Some of them that are easy to acquire, such as words in an entity and parts of speech of words in a context, are called local attributes; the others that are difficult to acquire, such as semantic classes of phrases and relations among them, are called decision attributes. The issue of entity and relation recognition is to determine a unique value for each decision attribute of all entities and relations, by considering the local attributes of them. To describe the problem in a formal way, we first give some basic definitions as follows.

Definition 1 (Entity). An entity can be a single word or a set of consecutive words with a predefined boundary. A sentence is a linked list, which consists of words and entities. Entities in a sentence are denoted as $E_1, E_2 \dots$ according to their order, with values ranging over a set of entity class C^E . For example, the sentence in Fig. 2 has three entities: $E_1=$ “Dole”, $E_2=$ “Elizabeth” and $E_3=$ “Salisbury, N.C.”. Note that it is not easy to determine the entity boundaries [7]. Here we assume that it has been solved and its output serves as the input to our model.

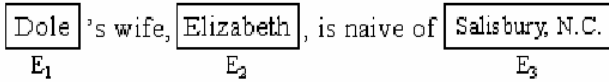


Fig. 2. A sentence that have three entities

Definition 2 (Relation). In this paper, we only consider the relation between two entities. An entity pair (E_i, E_j) represents a relation R_{ij} from entity E_i and E_j , where E_i is the first argument and E_j is the second argument. Relation R_{ij} takes its value that ranges over a set of relation class C^R . Note that (E_i, E_j) is an ordered pair, and there exist two relations $R_{ij}=(E_i, E_j)$ and $R_{ji}=(E_j, E_i)$ between entities E_i and E_j .

Definition 3 (Class). The class of an entity or relation is its decision attribute, which is one of the predefined class set and is unknown before being recognized. We denote the sets of predefined entity class and relation class as C^E and C^R respectively. C^E has one special element other-ent, which represents any unlisted entity class. For algorithmic reasons, we suppose all elements in C^E are mutually exclusive. Similarly, C^R also has one special element other-rel, which represents that the two involved entities are irrelevant or their relation class is undefined. For algorithmic reasons, we suppose all elements in C^R are mutually exclusive. In fact, because the class of an entity or a relation is only a label that we want to predict, if an entity or a relation have more than one labels simultaneously, to satisfy the constraint that all elements in C^E or C^R are mutually exclusive, we can separate it into several cases and construct several predefined entity class sets and relation class sets.

The classes of entities and relations in a sentence must satisfy some constraints. For example, if the class of entity E_1 , which is the first argument of relation R_{12} , is a location, then the class of relation R_{12} cannot be born-in because the class of the first argument in relation R_{12} has to be a person.

Definition 4 (Constraint). A constraint is a 5-tuple $(R, \varepsilon^1, \varepsilon^2, \alpha_R, \alpha_\varepsilon)$. The symbols are defined as follows. $R \in C^R$ represents the class of relation R . $\varepsilon^1, \varepsilon^2 \in C^E$ represents the classes of the first argument E_i and the second argument E_j in the relation R respectively. $\alpha_R \in [0,1]$ is a real number that represents a joint conditional probability distribution $\alpha_R = \Pr\{\varepsilon^1, \varepsilon^2 \mid R\}$. $\alpha_\varepsilon \in [0,1]$ is a real number that represents a conditional probability distribution $\alpha_\varepsilon = \Pr\{R \mid \varepsilon^1, \varepsilon^2\}$. Note that α_R and α_ε need not to be specified manually and can be learned from an annotated training dataset easily.

Definition 5 (Observation). We denote the observations of an entity and a relation in a sentence as O^E and O^R respectively. O^E or O^R represent all the “known” local attributes of an entity or a relation, e.g., the spelling of a word, parts of speech, and semantic related attributes acquired from external resources such as WordNet. The observations O^E and O^R can be viewed as a random event, and $\Pr\{O^E\} = \Pr\{O^R\} \equiv 1$ because O^E and O^R in a sentence are known.

Based on the above definitions, the issue of entity and relation recognition can be described in a formal way as follows. Suppose in a sentence, the set of entity is $\{E_1, E_2 \dots E_n\}$, the set of relation is $\{R_{12}, R_{21}, R_{13}, R_{31}, \dots, R_{1n}, R_{n1}, \dots, R_{n-1,n}, R_{n,n-1}\}$, the predefined sets of entity class and relation class are $C^E = \{e_1, e_2, \dots e_m\}$ and $C^R = \{r_1, r_2, \dots r_k\}$ respectively, the observation of entity E_i is O_i^E , and the observation of relation R_{ij} is O_{ij}^R . n , m and k represent the number of entity, the number of the predefined entity class and the number of the predefined relation class respectively. The problem is to search the most probable class assignment for each entity and each relation of interest, given the observations of all entities and relations. In other words, the problem is to solve the following two equations, using two kinds of constraint knowledge α_R, α_e and the interaction among entities and relations.

$$e = \arg \max_d \Pr\{E_i = e_d \mid O_1^E, O_2^E, \dots, O_n^E, O_{12}^R, O_{21}^R, \dots, O_{1n}^R, O_{n1}^R, \dots, O_{n-1,n}^R, O_{n,n-1}^R\} \quad (1)$$

$$r = \arg \max_d \Pr\{R_{ij} = r_d \mid O_1^E, O_2^E, \dots, O_n^E, O_{12}^R, O_{21}^R, \dots, O_{1n}^R, O_{n1}^R, \dots, O_{n-1,n}^R, O_{n,n-1}^R\} \quad (2)$$

In (1), $d=1, 2, \dots, m$, and in (2), $d=1, 2, \dots, k$.

3 The Proposed Method

Because the class assignment of a single entity or relation depends not only on local attributes itself, but also on those of all other entities and relations, the equations (1) and equation (2) cannot be solved directly. To simplify the problem, we present the following method consisting of three stages. Firstly, employ two classifiers to perform entity recognition and relation recognition independently. Their outputs are the conditional probability distributions $\Pr\{E \mid O^E\}$ and $\Pr\{R \mid O^R\}$, given the corresponding observations. Secondly, recognize an entity by taking account of all entities and relations, as computed in the previous step. This is achieved by using the model Entity Relation Propagation Diagram (ERPD). And, recognize a relation based on the results of the second step at last.

In this paper, we concentrate on the processes at the second and the third stages, assuming that the process at the first stage is solved and its output are given to us as input. At the second stage, the aim of introducing ERPD is to estimate the conditional probability distribution $\Pr\{E \mid \text{ERPD}\}$ given the constraint α_R in Definition 5 and the sets $\{\Pr\{E_i \mid O^{E_i}\}\}$ and $\{\Pr\{R_{ij} \mid O^{R_{ij}}\}\}$ ($i, j=1, \dots, n$), as computed at the first stage. For the readability, suppose $\Pr\{E \mid \text{ERPD}\}$ is given, the entity recognition equation (1) becomes the equation (3).

$$e = \begin{cases} \arg \max_d \Pr\{E_i = e_d \mid O^{E_i}\} & RV > \theta \\ \arg \max_d \Pr\{E_i = e_d \mid ERPD\} & RV \leq \theta \end{cases} \quad (3)$$

where θ is a threshold determined by the experiment. $RV \in [0, 1]$ is a real number, called the reliable value, representing the belief degree of the output of the entity recognizer at the first stage. Suppose the maximum value of the conditional probability distribution $\Pr\{E \mid O^E\}$ is V_m and the second value is V_s , RV is defined as:

$$RV = \frac{V_m - V_s}{V_m + V_s} \quad (4)$$

The reason of introducing RV is due to a fact that only for ambiguous entities, it is effective by taking the classes of all entities in a sentence into account. ‘‘Reliable Value’’ measures whether an entity is ambiguous.

At the third stage, the basic idea of recognizing a relation is to search the probable relation given its observation, under a condition of satisfying the constraints imposed by the results of entity recognition at the second stage. The relation recognition equation (2) becomes the equation (5).

$$r = \arg \max_k \Pr\{R = r^k \mid O_R\} \times W_R \quad (5)$$

$$W_R = \begin{cases} 1 & \text{if } \Pr\{r \mid \varepsilon^1, \varepsilon^2\} > 0 \\ 0 & \text{if } \Pr\{r \mid \varepsilon^1, \varepsilon^2\} = 0 \end{cases}$$

where $\varepsilon^1, \varepsilon^2$ is the results of entity recognition at the second stage, $\Pr\{r \mid \varepsilon^1, \varepsilon^2\}$ is constraint knowledge α_e in Definition 4, and W_R is the weight of the constraint knowledge.

In the following sections, we present ERPD and two algorithms to estimate the conditional probability distribution $\Pr\{E \mid ERPD\}$.

3.1 The Entity Relation Propagation Diagram

To represent the mutual dependencies among entities and relations, a model named the Entity Relation Propagation Diagram that can deal with cycles, similar to the Causality Diagram [8][9] for the complex system fault diagnosis, is developed for entity and relation recognition.

The classes of any two entities are dependent on each other through the relations between them, while taking account of the relations in between. For example, the class of entity E_i in Fig. 3 (a) depends on the classes of relations R_{ji} between entities E_i and E_j , and the classes of relations R_{ij} and R_{ji} depend on the classes of entities E_i and E_j . This means that we can predict the class of a target entity according to the class of its neighboring entity, making use of the relations between them. We further introduce the relation reaction intensity to describe the prediction ability of this kind.

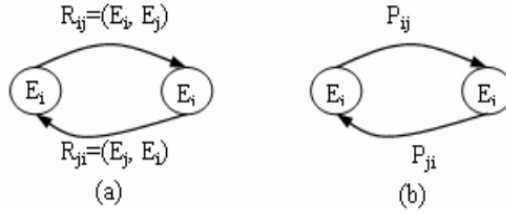


Fig. 3. Illustration of relation reaction

Definition 6 (Relation Reaction Intensity). We denote the relation reaction intensity from entity E_i to entity E_j as P_{ij} , which represents the ability that we guess the class of E_j if we know the class of its neighboring entity E_i and the relation R_{ij} between them. The relation reaction intensity could be modeled using a condition probability distribution $P_{ij}=\Pr \{E_j | E_i\}$.

The element p_{ij}^{kl} of P_{ij} represents the conditional probability $\Pr \{E_j=e_l | E_i=e_k\}$:

$$p_{ij}^{kl} = \Pr\{E_j = e_l | E_i = e_k\} = \sum_{r=1}^N \frac{\Pr\{R_{ij} = r_t\} \Pr\{E_i = e_k, E_j = e_l | R_{ij} = r_t\}}{\Pr\{E_i = e_k\}}$$

according to Definition 5:

$$\Pr\{R_{ij} = r_t\} = \Pr\{R_{ij} = r_t | O_{ij}^R\}, \Pr\{E_i = e_k\} = \Pr\{E_i = e_k | O_i^E\}$$

Then, we have:

$$p_{ij}^{kl} = \sum_{r=1}^N \frac{\Pr\{R_{ij} = r_t | O_{ij}^R\} \Pr\{E_i = e_k, E_j = e_l | R_{ij} = r_t\}}{\Pr\{E_i = e_k | O_i^E\}} \tag{6}$$

where $r_t \in C^R$, N is the number of relations in relation class set. In equation (6), $\Pr\{E_i = e_k, E_j = e_l | R_{ij} = r_t\}$ represents the constraint knowledge α_R among entities and relations. $\Pr\{R_{ij} = r_t | O_{ij}^R\}$ and $\Pr\{E_i = e_k | O_i^E\}$ represent the outputs at the first stage.

Definition 7 (Observation Reaction Intensity). We denote the observation reaction intensity as the conditional probability distribution $\Pr\{E | O^E\}$ of an entity class, given the observation, which is the output at the first stage.

The Entity Relation Propagation Diagram (ERPD). is a directed diagram that allows cycles. As illustrated in Fig. 4, the symbols used in the ERPD are defined as follows. A circle node represents an event variable that can be any one from a set of mutually exclusive events, which all together cover the whole sample space. Here, an event variable represents an entity, an event represents a predefined entity class, and the whole sample space represents the set of predefined entity classes. Box node represents a basic event which is one of the independent sources of the associated event variable. Here, a basic event represents the observation of an entity. Directed arc represents a linkage event variable that may or may not enable an input event to cause the corresponding output event. The linkage event variable from an event

variable to another event variable represents the relation reaction intensity in Definition 6. And, the linkage event variable from a basic event to the corresponding event variable represents the observation reaction intensity in Definition 7. All arcs pointing to a node are in a logical OR relationship.

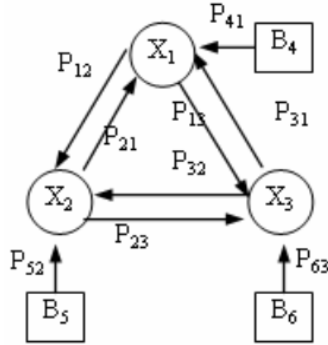


Fig. 4. Illustration of the Entity Relation Propagation Diagram

Now, we present two algorithms to compute the conditional probability distribution $\Pr\{E \mid \text{ERPD}\}$, one is based on the entity relation propagation tree, and the other is the directed iteration algorithm on ERPD.

3.2 The Entity Relation Propagation Tree

The Entity Relation Propagation Tree (ERPT), is a tree decomposed from an ERPD, which represents the relation reaction propagation from all basic events to each event variable logically. Each event variable in the ERPD corresponds to an ERPT. For example, the ERPT of X_1 in Fig. 4 is illustrated in Fig. 5. The symbols used in the ERPT are defined as follows. The root of the tree, denoted as Circle, is an event variable corresponding to the event variable in the ERPD. A leaf of the tree, denoted as Box, is a basic event corresponding to the basic event in the ERPD. The middle node of the tree, denoted as Diamond, is a logical OR gate variable, which is made from an event variable that has been expanded in the ERPD, and, the label in Diamond corresponds to the label of the expanded event variable. The directed arc of the tree corresponds to the linkage event variable in the ERPD. All arcs pointing to a node are in a logical OR relationship. The relation between the directed arc and the node linked to it is in logical AND relationship.

To decompose an ERPD into entity relation propagation trees, firstly we decompose the ERPD into mini node trees. Each event variables in the ERPD corresponds to a mini node tree, in which the root of the mini tree is the event variable in concern at present, and the leaves are composed of all neighboring basic events and event variables that are connected to the linkage event variables pointing to the top event variables. Secondly, expand a mini node tree into an entity relation propagation tree, i.e., the neighboring event variables in the mini node tree are replaced with their corresponding mini trees. During expanding a node event variable, when there are loops, Rule BreakLoop is applied to break down the loops.

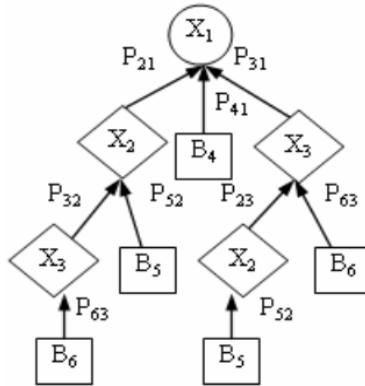


Fig. 5. Illustration of the entity relation propagation tree

Rule BreakLoop. An event variable cannot propagate the relation reaction to itself. Rule 1 is derived from a law commonsense - one can attest that he is sinless. When such a loop is encountered, the descendant event variable, which is same as the head event variable of the loop, is treated as a null event variable, together with its connected linkage event variable to be deleted.

Compute the Conditional Probability Distribution in an ERPT. After an ERPD is decomposed into entity relation propagation trees, the conditional probability distribution $\Pr\{E|ERPD\}$ becomes $\Pr\{E|ERPT\}$. When an event variable X_i has more than one input, these inputs will be in logic OR relationship, as defined in the ERPD. Since these inputs are independent, there exists such a case that one input causes X_i to be an instance X_i^k while another input causes X_i to be an instance X_i^l , this would be impossible because X_i^k and X_i^l are exclusive. In the real world, the mechanism, in which X_i can response to more than one independent input properly, is very complicated and may vary from one case to another. To avoid this difficulty, a basic assumption is introduced.

Assumption. When there is more than one input to X_i , each input will contribute a possibility to X_i . For each input, its contribution to this possibility equals to the probability that it causes X_i directly, as if the other inputs do not exist. The final possibility that X_i occurs is the sum of the possibilities from all inputs.

Suppose an event variable X has m inputs, and the probability distributions of all linkage event variables, linked basic events or event variables are P_i and $\Pr\{X_i\}$ respectively, $i=1,2,\dots,m$. Based on the above assumption, the formula for computing the probability distribution of X can be derived as:

$$\begin{bmatrix} \Pr\{X^1\} \\ \vdots \\ \Pr\{X^n\} \end{bmatrix} = \text{Norm}\left(\sum_{i=1}^m P_i \times \begin{bmatrix} \Pr\{X_i^1\} \\ \vdots \\ \Pr\{X_i^n\} \end{bmatrix}\right) \tag{7}$$

where, $\text{Norm}()$ is a function that normalizes the vector in $\{\}$, and n is the state number of X .

So, the probability distribution $\text{Pr}\{E|\text{ERPT}\}$ of the variable X in the corresponding ERPT can be computed in the following steps. Firstly, to find the middle node sequence in the corresponding ERPT in the depth-first search; secondly, according to the sequence, for each middle node, equation (7) is applied to compute its probability distribution. In this procedure, the previous results can be used for the latter computation.

3.3 The Directed Iteration Algorithm on ERPD

The idea is to compute the probability distribution of the event variable on the ERPD directly, without decomposing the ERPD to some ERPTs. The aim is to avoid the computational complexity of using ERPT. This is achieved by adopting an iteration strategy, which is the same as that used in the loopy belief network [10].

The Directed Iteration Algorithm. is as follows: Firstly, only take the basic event as input, and initialize each event variable according to formula (7), i.e., assigning an initialized probability distribution to each event variable. Secondly, take the basic event and the probability distributions of all neighboring nodes computed in the previous step as input, and iterate to update the probability distributions of all nodes in ERPD in parallel according to formula (7). Thirdly, if none of the probability distribution of all nodes in ERPD in successive iterations changes larger than a small threshold, the iteration is said to converge and then stops.

4 Experiments

Dataset. The dataset in our experiments is the same as the Roth's dataset "all" [11], which consists of 245 sentences that have the relation kill, 179 sentences that have the relation born-in and 502 sentences that have no relations. The predefined entity classes are other-ent, person and location, and the predefined relation classes are other-rel, kill and born-in. In fact, we use the results at the first stage in our method as the input, which are provided by W. Yih.

Experiment Design. We compare five approaches in the experiments: Basic, Omniscient, ERPD, ERPD* and BN. The Basic approach, which is a baseline, tests the performance of the two classifiers at the first stage, which are learned from their local attributes independently. The Omniscient approach is similar to Basic, the only difference is that the classes of entities are exposed to relation classifier and vice versa. Note that it is certainly impossible to know the true classes of an entity and a relation in advance. The BN is the method based on the belief network, -- we follow the BN method according to the description in [11]. The ERPD is the proposed method based on ERPT, and the ERPD* is the proposed method based on the directed iteration algorithm. The threshold of RV is 0.4.

Results. The experimental results are shown in Table 1. It can be seen from the table that 1) it is very difficult to improve the entity recognition because BN and Omniscient almost do not improve the performance of Basic; 2) the proposed method can

improve the precision, which is thought of being more important than the recall for the task of recognizing entity; 3) the relation recognition can be improved if we can improve the entity recognition, as indicated by the comparisons of Basic, ERPD and Omniscient; 4) the proposed method can improve the relation recognition, and its performance is almost equal to that of BN; 5) the performance of ERPD and ERPD* is almost equal, so the directly iteration algorithm is effective.

Table 1. Experimental results

Approach	Person			Location		
	Rec	Prec	F ₁	Rec	Prec	F ₁
Basic	90.9	89.2	90.0	83.8	83.8	83.6
ERPD	90.9	89.2	90.0	74.7	91.0	81.9
ERPD*	90.7	89.1	89.9	75.3	90.9	82.3
BN	87.7	90.7	89.1	83.4	83.2	83.1
Omniscient	90.9	89.5	90.1	83.8	84.5	84.0
Approach	Kill			Born-in		
	Rec	Prec	F ₁	Rec	Prec	F ₁
Basic	59.1	68.0	62.8	65.1	72.9	68.4
ERPD	57.6	81.4	66.6	64.0	84.9	72.4
ERPD*	56.9	81.2	66.1	64.0	83.9	72.0
BN	53.9	85.5	65.7	63.0	86.6	72.4
Omniscient	59.1	81.3	67.8	65.1	86.1	73.5

5 Related Work and Comparison

Targeting at the problems mentioned above, a method based on the belief network has been presented in [11], in which two subtasks are carried out simultaneously. Its procedure is as follows: firstly, two classifiers are trained for recognizing entities and relations independently and their outputs are treated as the conditional probability distributions for each entity and relation, given the observed data; secondly, this information together with the constraint knowledge among relations and entities are represented in a belief network [12] and are used to make global inferences for all entities and relations of interest. This method is denoted BN in our experiments.

Although BN can block the error propagation from the entity recognizer to the relation classifier as well as improve the relation recognition, it cannot make use of the information, which is only available in relation recognition, to help entity recognition. Experiments show that BN cannot improve entity recognition.

Comparing to BN, the proposed method in this paper can overcome the two shortcomings of it. Experiments show that it can not only improve the relation recognition, but also improve the precision of entity recognition. Moreover, the model ERPD could be more expressive enough than the belief network for the task of recognizing

entity and relation. It can represent the mutually dependences between entities and relations by introducing relation reaction intensity, and can deal with a loop without the limitation of directed acyclic diagram (DAG) in the belief network. At the same time, the proposed method can merge two kinds of constraint knowledge (i.e. α_R and α_e in Definition 4), but the method based on belief network can only use α_e . Finally, the proposed method has a high computation efficiency while using the directed iteration algorithm.

6 Conclusions

The subtasks of entity recognition and relation recognition are typically carried out sequentially. This paper proposed an integrated approach that allows the two subtasks to be performed in a much closer way. Experimental results show that this method can improve the entity and relation recognition in some degree.

In addition, the Entity Relation Propagation Diagram (ERPD) is used to figure out the dependencies among entities and relations. It can also merge some constraint knowledge. Regarding to ERPD, two algorithms are further designed, one is based on the entity relation propagation tree, the other is the directed iteration algorithm on ERPD. The latter can be regarded as an approximation of the former with a higher computational efficiency.

Acknowledgements

We would like to express our deepest gratitude to Roth D. and Yih W. for making their dataset available for us. The research is supported in part by the National 863 Project of China under grant number 2001AA114210-03, the National Natural Science Foundation of China under grant number 60321002, and the Tsinghua-ALVIS Project co-sponsored by the National Natural Science Foundation of China under grant number 60520130299 and EU FP6.

References

1. Chinchor, N. MUC-7 Information Extraction Task Definition. In Proceeding of the Seventh Message Understanding Conference (MUC-7), Appendices, 1998.
2. Califf, M. and Mooney, R. Relational Learning of Pattern-match Rules for Information Extraction. In Proceedings of the Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence, 328-334, Orlando, Florida, USA, AAAI Press, 1999.
3. Freitag, D. Machine Learning for Information Extraction in Informal Domains. *Machine learning*, 39(2/3): 169-202, 2000.
4. Roth, D. and Yih, W. Relational Learning via Prepositional Algorithms: An Information Extraction Case Study. In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, 1257-1263, Seattle, Washington, USA, Morgan Kaufmann, 2001.
5. Voorhees, E. Overview of the Trec-9 Question Answering Track. In The Ninth Text Retrieval Conference (TREC-9), 71-80, 2000.

6. Hirschman, L., Light, M., Breck, E. and Burger, J. Deep Read: A Reading Comprehension System. In Proceedings of the 37th Annual Meeting of Association for Computational Linguistics, 1999.
7. Abney, S.P. Parsing by Chunks. In S. P. Abney, R. C. Berwick, and C. Tenny, editors, Principle-based parsing: Computation and Psycholinguistics, 257-278. Kluwer, Dordrecht, 1991.
8. Xinghua Fan. Causality Diagram Theory Research and Applying it to Fault Diagnosis of Complexity System, Ph.D. Dissertation of Chongqing University, P.R. China, 2002.
9. Xinghua Fan, Zhang Qin, Sun Maosong, Huang Xiyue. Reasoning Algorithm in Multi-Valued Causality Diagram, Chinese Journal of Computers, 26(3), 310-322, 2003.
10. Murphy, K., Weiss, Y., and Jordan, M. Loopy Belief Propagation for Approximate Inference: An empirical study. In Proceeding of Uncertainty in AI, 467-475, 1999.
11. Roth, D. and Yih, W. Probability Reasoning for Entity & Relation Recognition. In Proceedings of 20th International Conference on Computational Linguistics (COLING-02), 835-841, 2002.
12. Pearl, J. Probability Reasoning in Intelligence Systems. Morgan Kaufmann, 1988.