2010 12th International Asia-Pacific Web Conference

Multi-View Clustering with Web and Linguistic Features for Relation Extraction

Yulan Yan, Haibo Li, Yutaka Matsuo, Zhenglu Yang, Mitsuru Ishizuka

The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, JAPAN
yulan@mi.ci.i.u-tokyo.ac.jp
lihaibo@mi.ci.i.u-tokyo.ac.jp
matsuo@biz-model.t.utokyo.ac.jp
yangzl@tkl.iis.u-tokyo.ac.jp
ishizuka@i.u-tokyo.ac.jp

***Abstract*— Binary semantic relation extraction is particularly useful for various NLP and Web applications. Currently Web-based methods and Linguistic-based methods are two types of leading methods for semantic relation extraction task. With a novel view on integrating linguistic analysis on local text with Web frequent information, we propose a multi-view co-clustering approach for semantic relation extraction. One is feature clustering by automatically learning clustering functions for Web features, linguistic features simultaneously based on a subset of entity pairs. The other is relation clustering, using the feature clustering functions to define learning function for relation extraction. Our experiments demonstrate the superiority of our clustering approach comparing with several state-of-the-art clustering methods.**

I. Introduction

Recent attention to automatically harvesting semantic resources has encouraged Data Mining and Natural Language Processing researchers to develop algorithms for it. Many efforts have also focused on extracting semantic relations between entities, such as *birth_date* relation, *CEO* relation, and other relations. Semantic relation extraction is also becoming an important component in various applications of Web mining [18] and NLP.

Currently one type of the leading methods in relation extraction are based on collecting redundancy information from a local corpus or use the Web as corpus [19]; [1]; [2]; [8]. Let us call them Web mining-based methods. The standard process is to scan or search the corpus to collect co-occurrences of word pairs with strings between them, then calculate term co-occurrence or generate textual patterns. In order to clearly distinguish from linguistic features below, let us call them Web features. For example, given an entity pair x, y with $Spouse$ relation, string "*x is married to y*" is a Web feature example. The method is used widely, however, even when patterns are generated from good-written texts, frequent pattern mining is non-trivial since the number of unique patterns is loose but many are non-discriminative and correlated. One of the main challenges and research interest for frequent pattern mining is how to abstract away from different surface realizations of semantic relations to discover discriminative patterns efficiently.

Another type of leading methods are using linguistic analysis for semantic relation extraction (see e.g., [14]; [3]; [12]; [17]). Let us call them linguistic-based methods. Currently, linguistic-based methods for semantic relation extraction are almost all supervised or semi-supervised, relying on pre-specification of the desired relationship or hand-coding initial seed words or features. The main process is to generate linguistic features based on the analysis of the syntactic, dependency or shallow semantic structure of text, then through training to identify entity pairs which assume a relationship and classify them into pre-defined relationships. For example, given an entity pair x, y and the sentence "*x is the wife of y*", syntactic, dependency features will be generated by analysis of the sentence. The advantage of these methods is using linguistic technologies to learn semantic information from different surface expressions.

Different from these relation extraction methods, in this paper, we address a novel view of relation extraction task, where we integrate linguistic features and Web features of entity pairs to enhance the clustering performance of extracting relations. In our problem, we do not have any labeled data or pairwise supervisory constraint knowledge. From Web view, a clustering operation on the target data can be performed using Web-based methods; on the other hand, from linguistic view, a clustering operation on the target data can be performed using linguistic-based method. The challenge is how to make use of both views to improve the performance.

Our solution for this two-view clustering problem is to perform two learning tasks through co-clustering. One is to merge features into clusters by perform co-clustering between Web features and linguistic features. The other is to cluster entity pairs by co-clustering between entity pairs and feature (Web&linguistic) spaces. We extend two co-clustering algorithms for our solution. One is the information theoretic co-clustering algorithm [11] which minimizes loss in mutual information before and after clustering. The other is self-taught clustering algorithm [6] which performs clustering on a set of target data with auxiliary data simultaneously to allow the feature representation from the auxiliary data to influence the target data through a common set of features. Separate

978-0-7695-4012-2/10 $26.00 © 2010 IEEE
DOI 10.1109/APWeb.2010.64

from those two works, we introduce a multi-view co-clustering approach which consists of two steps, we call it dual co-clustering. In the first step it automatically learns clustering functions for Web features, linguistic features and entity pairs simultaneously. Then in the second step, the feature clustering functions are used to learn a relation clustering as the final objective function. Our experiments on a dataset from Wikipedia corpus demonstrate the superiority of our clustering approach comparing with several state-of-the-art clustering methods.

The main contributions of this paper are as follows:

- We propose a multi-view co-clustering algorithm. One is learning clustering functions for Web features and linguistic features simultaneously. The other is learning a clustering function for entity pairs based on feature clustering functions.
- Based on these algorithms, we construct an integrated framework for relation extraction task combining with Web features and linguistic features. The whole workflow is an instance of multi-view unsupervised learning. To the best of our knowledge, our approach is novel for various machine learning applications, especially for semantic relation extraction task.
- Our study suggests an example to bridge the gap between Web mining technology and "deep" linguistic technology for information extraction tasks. It shows how deep linguistic features can be combined with features from the whole Web corpus to improve the performance of information extraction tasks.

The remainder of the paper is organized as follows. In section 2 we will consider related work of this work. In section 3 we define the problem formulation and present out our solution. In section 4 we will report on our experimental results. Finally, in section 5 we will conclude the paper.

II. Related Work

In this section, we review several past research works that are related to our work, including, Web-based clustering, linguistic-based clustering and multi-view clustering.

The field of Unsupervised Relation Identification (URI) - the task of automatically discovering interesting relations between entities in a large text corpora was introduced by [13]. In [20] they showed that the clusters discovered by URI can be used for seeding a semi-supervised relation extraction system. To compare different clustering algorithm, feature extraction and selection method, the authors in [21] presented a URI system which used two kinds of surface patterns: patterns that test two entities together and patterns that test only one entity each. [7] proposed a method for unsupervised discovery of concept specific relations, requiring initial word seeds. They used pattern clusters to define general relationships, specific to a given concept. [8] presented an approach to discover and represent general relationships present in an arbitrary corpus. They presented a fully unsupervised algorithm for pattern cluster discovery, which searches, clusters and merges high frequency words-based patterns around randomly selected concepts.

Although linguistic-based relation extraction approaches for semantic relation extraction are almost supervised or semi-supervised, [4] presented an application of spectral clustering technique to unsupervised relation extraction problem, making use of various lexical and syntactic features from the contexts. [22] used simple predicate-argument patterns around the entities of candidate pairs. Their system worked on news articles, and improves its accuracy by looking at multiple news sources describing the same event. [16] built lexically-specific features by looking for verbs, prepositions, and coordinating conjunctions that can help make explicit the hidden relations between the target nouns.

Another related field is multi-view clustering[10]; [11]; [9]; [15]. Multiple view unsupervised learning is a fairly new topic. There is several work on multiple view clustering. Co-clustering techniques, which aim to cluster different types of data simultaneously by making efficient use of the relationship information, are proposed. [10] proposed a Bipartite Spectral Graph Partitioning approach to co-cluster words and documents. [11] presented the information theoretic co-clustering algorithm. With their information theoretic co-clustering, the objective function of co-clustering is defined as minimizing loss in mutual information between entity pairs and features, before and after co-clustering. [9] also assumes two independent views for a multiple view data set and proposes a spectral clustering algorithm which creates a bipartite graph and is based on the minimizing-disagreement idea.

In this study, we propose a multi-view co-clustering approach for relation extraction task based on a combination of two types of features. On the one hand, Web features are generated from the Web information to provide frequency information. On the other hand, linguistic features are generated from local sentences by linguistic analysis to abstract information away from surface realizations of texts.

III. Dual Co-clustering Approach for Multi-view Learning

In this section, we present a dual co-clustering approach for relation extraction task based on two kinds of generated features: Web features and linguistic features.

A. Problem Formulation and Outline of the Proposed Approach

We define the multi-view relation clustering task. The task is that given a target dataset of entity pairs such as "Bill Gates & Microsoft", first we generate Web features and linguistic features from contexts of each entity pair, then cluster all the entity pairs into groups based on these features, each group represents a relationship, such as "CEO". Let X_{all} be a discrete random variable, taking values from the target data set $\{x_1, ...x_l\}$ which contains all entity pairs to be labeled with their relation types. We are interested in clustering X_{all} into L clusters, each of which represents one relation type. Let Y and Z be two discrete random variables, taking values from two value set $\{y_1, ...y_m\}$ and $\{z_1, ...z_n\}$, that respectively corresponds to two different feature spaces of

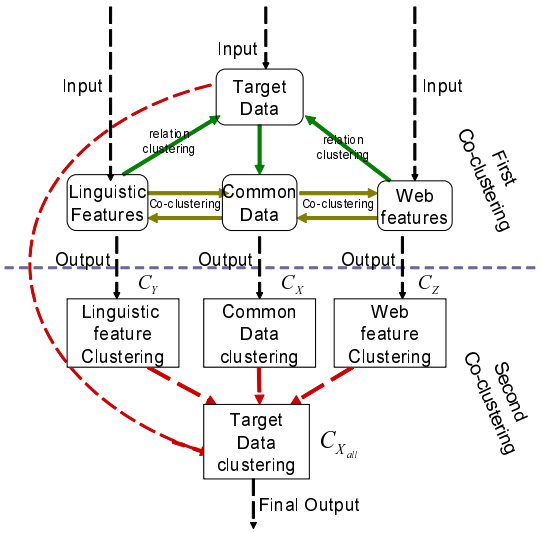

Fig. 1. Outline of the proposed multi-view co-clustering approach.

X_{all}. Y represent features from Web frequency information, Z represent features from linguistic analysis. Respectively with only Web features or with only linguistic features, X_{all} will be clustered into L clusters in two different ways. However, Web and linguistic features usually represent two aspects of the meaningful of the same relations, thus there must be some deep connection between them. The main task of this work is that given X_{all} with its feature spaces Y and Z, how to learn the connection between Web features and linguistic features to help perform clustering for relation extraction.

Our novel idea is to use another variable X, which works as an intermediate variable, to explore the deep connection between Web and linguistic features. It is used for information transformation among Web features, linguistic features and the target data set. Let X be a discrete random variable, taking values from value sets $\{x_1, ..., x_p\}$, which we call common data, corresponding to the shared entity pairs after perform relation clustering over Web features and linguistic features separately. The common data is a subset of the target data. Section 3.2 will explain how to obtain the common data in detail.

Figure 1 shows the outline of our solution. The proposed approach consists of two co-clustering steps: co-clustering learning for feature clusterings and co-clustering learning for relation clustering. In the first step, we are interested in simultaneously clustering X into L clusters, Y into (at most) M clusters, and Z into (at most) N clusters. In other words, we are interested in finding clustering functions C_X, C_Y and C_Z. The second step is to reach our objective which is to find a good clustering function $C_{X_{all}}$ for the whole target data, with the support of clustering functions C_X, C_Y and C_Z from the previous step. For brevity, in the following, we will use $\tilde{X_{all}}$, $\tilde{X}$, $\tilde{Y}$ and $\tilde{Z}$ to denote $C_{X_{all}}(X_{all})$, $C_X(X)$, $C_Y(Y)$ and $C_Z(Z)$, respectively. In other words, we are interested in firstly finding maps C_X, C_Y and C_Z and then finding map $C_{X_{all}}$:

$$C_X : \{x_1, ..., x_p\} \rightarrow \{\tilde{x}_1, \tilde{x}_2, ...\tilde{x}_L\} \quad (1)$$

$$C_Y : \{y_1, ..., y_m\} \rightarrow \{\tilde{y}_1, \tilde{y}_2, ...\tilde{y}_M\} \quad (2)$$

$$C_Z : \{z_1, ..., z_n\} \rightarrow \{\tilde{z}_1, \tilde{z}_2, ...\tilde{z}_N\} \quad (3)$$

$$C_{X_{all}} : \{x_1, ..., x_l\} \rightarrow \{\tilde{x}_{a1}, \tilde{x}_{a2}, ...\tilde{x}_{aL}\} \quad (4)$$

B. Initialization of Common Data

Algorithm 1: The Common Data Initialization Algorithm

Input: $\tilde{X_1} = \{\tilde{x}_{11}, \tilde{x}_{12}, ..., \tilde{x}_{1L}\}$(clustering target data based on only Web features)
$\tilde{X_2} = \{\tilde{x}_{21}, \tilde{x}_{22}, ..., \tilde{x}_{2L}\}$(clustering target data based on only linguistic features)
Output: common data clustering $\tilde{X}$

1 define a $L \times L$ similarity matrix A: $\{A_{ij} = |(\tilde{x_{1i}} \bigcap \tilde{x_{2j}})|$ 1≤i≤L; 1≤j≤L};
2 $\tilde{X} = \phi$
3 **for** L *times* **do**
4 (a, b) = $\text{argmax}_{0<i,j<L} A_{ij}$;
5 $\tilde{X}$ = $\tilde{X}$ + $(\tilde{x_{1a}} \bigcap \tilde{x_{2b}})$;
6 A_{a*} = 0; A_{*b} = 0;
7 return $\tilde{X}$

Fig. 2. Common data initialization

The common data set is important for information connection between Web feature space and linguistic feature space. We initialize common data X and clustering function C_X on X by three steps:

- Step 1: perform clustering operations on the target data over Web features and linguistic features separately;

$$C_{XY} : \{x_1, ..., x_l\} \rightarrow \{\tilde{x}_{11}, \tilde{x}_{12}, ...\tilde{x}_{1L}\}$$
$$C_{XZ} : \{x_1, ..., x_l\} \rightarrow \{\tilde{x}_{21}, \tilde{x}_{22}, ...\tilde{x}_{2L}\}$$

- Step 2: take two above clustering results as input for the common data initialization algorithm in Figure 2, we get the output which is a set of relation clusters.
Algorithm 1 details the process involved in this initialization. The input is two sets of relation clusters $\tilde{X_1}$ and $\tilde{X_2}$ resulting from Step 1. The algorithm starts with defining a similarity matrix by counting the shared number of entity pairs between each pair of clusters from $\tilde{X_1}$ and $\tilde{X_2}$. The main loop then starts at line 3 and iterates L times. In each iteration, the entry A_{ab} with the largest value is chosen. The common entity pairs of ath cluster from $\tilde{X_1}$ and $\tilde{X_2}$ will form a new relation cluster, and then be added into the common cluster set $\tilde{X}$.

$$C_{XY} \wedge C_{XZ} : \{x_1, ..., x_l\} \rightarrow \{\tilde{x}_1, \tilde{x}_2, ...\tilde{x}_L\} \quad (5)$$

- Step 3: simply from Equation 5, release all the entity pairs from the common cluster set to collect the common data in Equation 6. The initial clustering function for the common data is formulated in Equation 7.

$$\{\tilde{x}_1, \tilde{x}_2, ...\tilde{x}_L\} \Rightarrow \{x_{c1}, ..., x_{cp}\} \quad (6)$$

$$C_X^0 : \{x_{c1}, ..., x_{cp}\} \rightarrow \{\tilde{x}_1, \tilde{x}_2, ...\tilde{x}_L\} \quad (7)$$

C. Objective Function for Clustering Algorithm

We extend the information theoretic co-clustering [11] and self-taught clustering [6] to model our dual co-clustering learning algorithm. In the information theoretic co-clustering, the objective function of co-clustering is defined as minimizing loss in mutual information between entity pairs and features, before and after co-clustering. Formally, using the target data X and their feature space Y for illustration, the objective function can be expressed as:

$$I(X,Y) - I(\tilde{X},\tilde{Y}) \tag{8}$$

where $I(.,.)$ denotes the mutual information between two random variables [5] that $I(X,Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}$. Moreover, $I(\tilde{X},\tilde{Y})$ corresponds to the joint probability distribution $p(\tilde{X},\tilde{Y})$ which is defined as:

$$p(\tilde{x},\tilde{y}) = \sum_{x\in\tilde{x}}\sum_{y\in\tilde{y}} p(x,y) \tag{9}$$

[6] extended the information theoretic co-clustering [11] to model a self-taught clustering algorithm. They model their self-taught clustering algorithm as performing co-clustering operations on the target data X and auxiliary data Y, simultaneously, while the two co-clusters share the same features clustering $\tilde{Z}$ on the feature set Z. Their objective function is formulated as:

$$I(X,Z) - I(\tilde{X},\tilde{Z}) + \lambda[I(Y,Z) - I(\tilde{Y},\tilde{Z})] \tag{10}$$

λ is a trade-off parameter to balance the influence between the target data and the auxiliary data. Z is used as the bridge to connect the knowledge between the target and auxiliary data.

In this work, we model our multi-view co-clustering learning algorithm in a two-step of clustering process: feature clustering and relation clustering.

- Learning Feature Clustering Functions

In the first step, we model our feature clustering as performing co-clustering operations on the common data X, feature set Y and feature set Z, simultaneously, while the two clusterings on Y and Z share a common relation clustering $\tilde{X}$ on the target data. The objective function for feature clustering defined as minimizing loss in mutual information between entity pairs and features can be formulated as:

$$I(X,Y) - I(\tilde{X},\tilde{Y}) + \lambda[I(X,Z) - I(\tilde{X},\tilde{Z})] \tag{11}$$

In Equation 11, $I(X,Y) - I(\tilde{X},\tilde{Y})$ is computed on the clustering over only Web feature space Y on the common data, while $I(X,Z) - I(\tilde{X},\tilde{Z})$ is computed over only linguistic feature space Z. We also use λ as a trade-off parameter to balance the contribution between Web features and linguistic features, which we will test in our experiments. The objective is to find maps C_Y and C_Z towards a common relation clustering C_X. Intuitively, in an ideal way, targeting on the common data, the clustering function C_Y over Web features and C_Z over linguistic features will lead to the same clustering result. This restriction enables us to build a "bridge" to connect the knowledge between two feature spaces.

Our remaining task is to minimize the value of the objective function in Equation 11. Equation 11 is different from Equation 10 in this way: in Equation 10, the shared feature set Z is the bridge connecting the target data and auxiliary data; while in Equation 11, a subset of target data is the bridge connecting features. We apply the self-taught clustering algorithm in this task to minimize Equation 11 through optimizing this objective function into the form of Kullback-Leibler divergence [5] (KL divergence), and then minimize the reformulated objective function.

Finally, if we iteratively choose the best cluster $\tilde{y}$ for each y to minimize $D(p(X|y)||\tilde{p}(X|\tilde{y}))$, the objective function 11 will be minimized monotonically. Formally,

$$C_Y(y) = \arg\min_{\tilde{y}\in\tilde{Y}} D(p(X|y)||\tilde{p}(X|\tilde{y})) \tag{12}$$

Using a similar argument on Z and X, we have

$$C_Z(z) = \arg\min_{\tilde{z}\in\tilde{Z}} D(q(X|z)||\tilde{q}(X|\tilde{z})) \tag{13}$$

$$\begin{aligned} C_X(x) = \arg\min_{\tilde{x}\in\tilde{X}} & p(x)D(p(Y|x)||\tilde{p}(Y|\tilde{x})) \\ & +\lambda q(x)D(q(Z|x)||\tilde{q}(Z|\tilde{x})) \end{aligned} \tag{14}$$

In each iteration, the optimization algorithm minimizes the objective function by choosing the best $\tilde{y}$, $\tilde{z}$ and $\tilde{x}$ for each y, z and x based on Equation 12, 13 and 14, respectively. As discussed in [6], this can reduce the value of the global objective function in Equation 11.

- Learning Relation Clustering Function

Subsequently, with map functions C_X, C_Y and C_Z on X, Y and Z, we are interested in finding map function $C_{X_{all}}$ for the whole target data X_{all}. Let F be a discrete random variable, taking values from the whole feature space $Y \bigcup Z$. Similar to learning functions for feature clustering, the final objective function defined as minimizing loss in mutual information between entity pairs and features can be formulated as

$$\begin{aligned} & I(X_{all},F) - I(\tilde{X_{all}},\tilde{F}) \\ & \quad = D(p(X_{all},F)||\tilde{p}(X_{all},F)) \end{aligned} \tag{15}$$

From the same induction as for the objective loss function for feature clustering, to minimize Equation 15 is to reduce the value of $D(p(X_{all},F)||\tilde{p}(X_{all},F))$.

We have

$$\begin{aligned} & D(p(X_{all},F)||\tilde{p}(X_{all},F)) \\ & \quad = \sum_{\tilde{x}\in\tilde{X_{all}}}\sum_{\tilde{f}\in\tilde{F}}\sum_{x\in\tilde{x}}\sum_{f\in\tilde{f}} p(x,f)log\frac{p(x,f)}{\tilde{p}(x,f)} \end{aligned} \tag{16}$$

Since $\tilde{p}(x,f) = p(x)\frac{p(\tilde{x},\tilde{f})}{p(\tilde{x})}\frac{p(f)}{p(\tilde{f})}$, we have

$$\begin{aligned}
&D(p(X_{all},F)||\tilde{p}(X_{all},F)) \\
&= \sum_{\tilde{x}\in\tilde{X_{all}}}\sum_{\tilde{f}\in\tilde{F}}\sum_{x\in\tilde{x}}\sum_{f\in\tilde{f}} p(x)p(f,x)log\frac{p(x)p(f/x)}{p(x)\tilde{p}(f/\tilde{x})} \\
&= \sum_{\tilde{x}\in\tilde{X_{all}}}\sum_{x\in\tilde{x}} p(x)\sum_{\tilde{f}\in\tilde{F}}\sum_{f\in\tilde{f}} p(f/x)log\frac{p(f/x)}{\tilde{p}(f/\tilde{x})}
\end{aligned}$$

where $\tilde{X_{all}}$ is the objective cluster set, Y and Z are independent, we have

$$\begin{aligned}
&D(p(X_{all},F)||\tilde{p}(X_{all},F)) \\
&= \sum_{\tilde{x}\in\tilde{X_{all}}}\sum_{x\in\tilde{x}} p(x)\{\sum_{\tilde{y}\in\tilde{Y}}\sum_{y\in\tilde{y}} p(y/x)log\frac{p(y/x)}{\tilde{p}(y/\tilde{x})} \\
&\qquad +\lambda\sum_{\tilde{z}\in\tilde{Z}}\sum_{z\in\tilde{z}} p(z/x)log\frac{p(z/x)}{\tilde{p}(z/\tilde{x})}\} \\
&= \sum_{\tilde{x}\in\tilde{X_{all}}}\sum_{x\in\tilde{x}} p(x)\{D(p(Y|x)||p(Y|\tilde{x})) \\
&\qquad +\lambda D(p(Z|x)||p(Z|\tilde{x}))\}
\end{aligned}$$

Since Web features and linguistic features have been clustered, $\tilde{X}$ is used as the seed cluster set, if we choose the best cluster $\tilde{x}$ from $\tilde{X}$ for each x in $X_{all} - X$ to minimize $D(p(X,F)||\tilde{p}(X,F))$, the objective function will be minimized. Formally

$$C_X(x_{all}) = \tilde{x}, x_{all} \in X \& x_{all} \in \tilde{x} \tag{17}$$

$$\begin{aligned}
&C_{X_{all}}(x_{all}) \\
&= \arg\min_{\tilde{x}\in\tilde{X}}\{p(x_{all})D(p(Y|x_{all})||\tilde{p}(Y|(\tilde{x})) \\
&+\lambda q(x_{all})D(q(Z|x_{all})||\tilde{q}(Z|(\tilde{x}))\}, x_{all} \notin X
\end{aligned} \tag{18}$$

Based on Equation 17 and 18, an alternative way to minimize the objective function in Equation 15 is derived. If entity pair x is in the common data X, we simply choose the cluster $\tilde{x}$ that it maps to using map function C_X.

IV. Experiments

In this section, we evaluate our multi-view co-clustering approach on the relation extraction task, and show the effectiveness of the proposed approach.

A. Experimental Setup

We conduct our experiments on relation extraction task using the dataset that was created for evaluating relation extraction from Wikipedia in [17]. The dataset consists of 3833 positive relation instances (entity pairs), for 13 relation types which are the Spouse, President, Vice_Chairman, COO, Director, Chairman, Founder, CEO, Birth_date, Birth_place, Products, Foundation and Location relations. Each relation instance (entity pair) in the dataset has one accompanying sentence from a Wikipedia article.

TABLE II
Overall performance

Method	pre.	rec.	f-v.
Linguistic clustering	41.18	31.47	35.09
Web clustering	47.31	45.72	46.50
Proposed clustering	67.74	54.03	60.11

We build two baseline systems on the dataset. One baseline system is built using [21]'s URI method which showed that their algorithm improved over previous work using Web features for unsupervised relation extraction: features that test two entities together and features that test only one slot each. We use this system to represent the performance of Web-based relation extraction methods. The other system is built using [4]'s method, which is demonstrated in their paper, that outperforms other clustering methods by use of various lexical and syntactic features from the contexts. We use it to represent the performance of linguistic-based relation extraction methods.

To evaluate the performance of our approach, we collect Web features through querying with entity pairs by a search engine (Google). Different from simply taking the entire string between two concept words which capture an excess of extraneous and incoherent information, our idea of getting Web features is to look for verbs, nouns, prepositions, and coordinating conjunctions that can help make explicit the hidden relations between the target nouns. To collect linguistic features, for each entity pair, the accompanying sentence is parsed using a linguistic parser. We generate dependency patterns as sub-paths from the shortest dependency path [17] containing two entities by making use of a frequent tree-mining algorithm [23].

In these experiments, we use precision, recall, and F-value to measure the performance of different methods. The following quantities are considered to compute precision, recall, and F-value:

- p = the number of detected entity pairs.
- p' = the number of detected entity pairs which are actual relation instances.
- n = the number of actual relation instances.

Precision (P) = p'/p Recall (R) = p'/n
F-value (F) = $2 * P * R/(P + R)$

B. Empirical Analysis

Table 1 presents the comparison between our approach and two baseline systems. Using our multi-view co-clustering approach, it is effective to integrate Web features and linguistic features by information transformation among Web features, linguistic features and entity pairs, with precision 67.74%, recall 54.03% and F-value 60.11%. From this table, we can see that the performance of the proposed approach is better than both the Web-based method (with precision 47.31%, recall 45.72% and F-value 46.50%) and the linguistic-based method (with precision 41.18%, recall 31.47% and F-value 35.68%) for relation extraction task. Using different feature

TABLE I

PERFORMANCE COMPARISON USING DIFFERENT METHODS

Relation	Linguistic feature clustering			Web feature clustering			Multi-view clustering		
	Pre.	Rec.	F-v.	Pre.	Rec.	F-v.	Pre.	Rec.	F-v.
Spouse	19.02	45.13	26.76	52.31	39.73	45.16	64.23	51.58	57.21
President	14.07	40.00	20.82	19.71	25.00	22.04	22.63	39.51	28.78
Vice_Chairman	67.14	14.81	24.27	20.61	16.67	18.43	45.82	26.64	33.69
COO	100.0	11.17	20.10	14.55	10.88	12.45	25.78	21.42	23.40
Director	87.50	42.31	57.04	40.25	37.69	38.93	55.32	47.57	51.15
Chairman	24.62	21.59	23.01	41.79	43.54	42.65	57.36	46.45	51.33
Founder	72.70	59.43	65.40	28.99	52.61	37.38	67.02	71.49	69.18
CEO	48.89	17.49	25.76	35.96	42.62	39.01	51.85	41.90	46.35
Birth_date	56.67	72.35	63.56	73.80	82.06	77.71	78.62	88.74	83.37
Birth_place	24.93	13.19	17.25	63.19	48.70	55.01	66.31	51.57	58.02
Products	100.0	11.16	20.08	58.67	31.32	40.84	63.51	36.14	46.07
Foundation	72.26	53.42	61.43	61.11	47.83	53.66	84.32	63.86	72.68
Location	72.16	16.97	27.48	63.91	51.82	57.23	74.19	49.86	59.64
overall	41.18	31.47	35.68	47.31	45.72	46.50	67.74	54.03	60.11

TABLE III

MOST FREQUENT WEB FEATURES IN THE CLUSTERS

Spouse	X marry Y	X be married to Y	X wife Y	X husband Y
President	X president of Y	X be president of Y	X president for Y	Y president X
Vice_Chairman	Y vice chairman X	X be vice chairman of Y	X as Vice Chairman of Y	X vice chairman of Y
COO	Y coo X	Y be coo of X	X be chief operating officer of Y	X as coo of Y
Director	X be director of Y	X Y director	X director Y	X director of Y
Chairman	X be chairman of Y	X ceo and chairman of Y	Y chairman of committee X	Y board chairman X
Founder	Y be found by X	X founder Y	X be founder of Y	Y founder X
CEO	X be ceo of Y	Y ceo X	X be chief executive officer of Y	Y ceo X
Birth_date	X be bear on Y	X bear in Y	X bear in Y	bear in Y X
Birth_place	X be bear in Y	X be bear in district of Y	X birth place Y	X birthplace Y
Products	X supplier of product to Y	X deliver Y	X launch Y	X provide Y service
Foundation	X be find in Y	X based in Y	X establish in Y	X foundation Y
Location	X located in Y	X be located in Y	X be headquartered in Y	X site in Y

TABLE IV

MOST FREQUENT LINGUISTIC FEATURES IN THE CLUSTERS

Spouse	(marry(subj:)(obj:(Y)))	(marry(subj:(X))(obj:(Y)))	(married(v-ch:(to(pcomp:))))	(marry(obj:(Y)))
President	(president(mod:(of)))	(president(mod:))	(president(mod:(of(pcomp:))))	(be(comp:(president)))
Vice_Chairman	(vice-chairman(mod:(of)))	(vice-chairman(mod:(of)))	(be(vice-chairman(mod:)))	(be(comp:(vice-chairman)))
COO	(coo(ha:(of(pcomp:))))	(coo(ha:(of(pcomp:(Y))))	(coo(ha:(of)))	(coo(ha:(of(pcomp:(Y)))))
Director	(be(comp:(director)))	(director(mod:(of(pcomp:))))	(be(subj:)(comp:(director(mod:))))	(be(comp:(director)))
Chairman	(be(comp:(chairman)))	(chairman(mod:(of(pcomp:(Y)))))	(become(subj:)(comp:(chairman)))	(comp:(chairman(mod:(of))))
Founder	(found(agt:(by)))	(found(agt:(by(pcomp:(X)))))	(co-founder(mod:(of(pcomp:))))	(co-founder(mod:(of)))
CEO	(become(comp:(ceo)))	(become(comp:(ceo(mod:(of)))))	(X(attr:(ceo)))	(ceo(attr:(Y)))
Birth_date	(bear(v-ch:)(ha:(Y)))	(bear(v-ch:(be(subj:)))(tmp:(in)))	(bear(v-ch:(be(subj:(X))))	(bear(tmp:(in)))
Birth_place	(bear(loc:))	(bear(v-ch:(be(subj:)))(loc:(in)))	(bear(v-ch:(be))(loc:))	(bear(v-ch:(be(subj:(X)))))
Products	(provide(subj:(X)))	(provide(subj:(X))(obj:(Y)))	(provider(mod:(include(obj:(Y)))))	(release(ha:(for(pcomp:))))
Foundation	(found(loc:(in)))	(form(v-ch:(be(subj:))))	(establish(phr:(in(pcomp:(Y)))))	(found(loc:(in(pcomp:))))
Location	(X(mod:(locate(loc:))))	(locate(loc:(in)))	(base(loc:(in(pcomp:(Y)))))	(cla:(headquarter(loc:)))

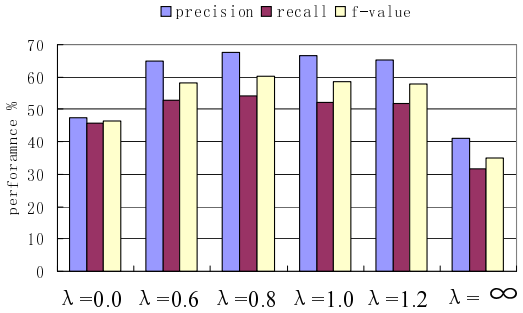

Fig. 3. Performance against trade-off values

sets, the performance is different. It shows that each kind of feature type contributes differently to our task. Another observation is that Web features and linguistic features provide complementary information to relation extraction task, so that by learning the connectivity between them, the performance of relation extraction is boosted. It's worth noting that our multi-view co-clustering approach shows much higher precision than both Web-based and linguistic-based methods, with similar features clustered into small groups and by minimizing loss in mutual information before and after clustering of Web features, linguistic features and entity pairs.

We use the feature clustering function described in section 3.3.1 to cluster Web features and linguistic features, and use relation clustering function described in section 3.3.2 to cluster entity pairs. As described in Equation 11 and 18, a trade-off parameter λ between Web and linguistic features is used to determine the contribution of different features. As shown in Figure 3, we test the dataset against several values of

λ: $\lambda = 0.6$, $\lambda = 0.8$, $\lambda = 1.0$ and $\lambda = 1.2$. $\lambda = 0.0$ means using only Web features, while $\lambda = \propto$ means using only linguistic features. It can be seen that the performance is the best when λ is 0.8. This means that Web features contribute more than linguistic features. The results support our assumptions about Web information and linguistic analysis technologies: 1) Dependency analysis can abstract away from different surface realizations of text. In addition, embedded structures of the dependency representation are important features for relation extraction task. 2) Surface patterns are used to merge concept pairs with relations represented in different dependency structures with redundancy information from the vast size of Web pages. Using surface patterns, more concept pairs are clustered, and the coverage is improved.

For each relation cluster in Table 3, we show top four Web features that occur with the largest frequency. From Table 3, it is clear that each cluster contains different Web features that express a specific semantic relation. X and Y in feature expressions are used to label the first entity and second entity of a relation instance respectively. Similarly, in Table 4, for each relation cluster, we show the top four linguistic features that occur with the largest frequency. We see that linguistic features in different surface expressions are clustered to represent the same semantic relation. Moreover, each cluster contains different linguistic features that express a specific semantic relation. Each linguistic feature denotes one tree transaction represented in strict S-expression. Strict means that all nodes, even leaf nodes, must be bracketed.

All the experimental results support our idea mainly in two main ways: 1) the combination of Web features and linguistic features is effective in relation extraction task; 2) multi-view co-clustering learning which makes use of knowledge gained from feature learning task is feasible to improve the performance of relation clustering task even in an unsupervised way.

V. Conclusions

To discover a range of semantic relationships from large-scale corpus, we present an unsupervised relation extraction approach to use deep linguistic information to alleviate surface and noisy surface features generated from large corpus, and use Web frequency information to ease the sparseness of linguistic information. We propose a multi-view co-clustering approach for semantic relation extraction task. One is learning clustering functions for Web features and linguistic features simultaneously. The other is learning a clustering function for entity pairs based on feature clustering functions. The proposed approach is an instance of unsupervised multi-view clustering. To the best of our knowledge, our approach is novel for various machine learning applications, especially for semantic relation extraction task. We report our experimental results comparing it to previous work and evaluating it over using different features. The results show that the performance of our proposed approach is the best when compared with several existed clustering methods.

References

[1] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead and O. Etzioni. 2007. *Open information extraction from the Web.* In proceedings of IJCAI-2007.

[2] D. Bollegala, Y. Matsuo and M. Ishizuka. 2007. *Measuring Semantic Similarity between Words Using Web Search Engines.* In Proceedings of WWW-2007.

[3] R. Bunescu and R. Mooney. 2005. *A shortest path dependency kernel for relation extraction.* In Proceedings of HLT/EMLNP-2005.

[4] J. Chen, D. Ji, C.L. Tan, and Z. Niu. 2006. *Unsupervised Feature Selection for Relation Extraction.* In Proceedings of EMNLP-2006.

[5] T. M. Cover, J. A. Thomas. 1991. *Elements of information theory.* Wiley-Interscience.

[6] D. Dai, Q. Yang, G. Xue and Y. Yu. 2008. *Self-taught Clustering.* In Proceedings of the ICML-2008.

[7] D. Davidov, A. Rappoport and M. Koppel. 2007. *Fully unsupervised discovery of concept-specific relationships by Web mining.* In Proceedings of ACL-2007.

[8] D. Davidov and A. Rappoport. 2008. *Classification of Semantic Relationships between Nominals Using Pattern Clusters.* In Proceedings of ACL-2008.

[9] V. R. de Sa. 2005. *Spectral clustring with two views.* In ICML Workshop on Learning with Multiple Views, 2005.

[10] I. S. Dhillon. 2001. *Co-clustering documents and words using bipartite spectral graph partitioning.* In Knowledge Discovery and Data Mining, pages 269C274, 2001.

[11] I. S. Dhillon, S. Mallela, and D. S. Modha. 2003. *Information-theoretic co-clustering.* In Proceedings of KDD-2003.

[12] S. Harabagiu, C.A. Bejan and P. Morarescu. 2005. *Shallow semantics for relation extraction.* In Proceedings of IJCAI-2005.

[13] T. Hasegawa, S. Sekine, and R. Grishman. 2004. *Discovering Relations among Named Entities from Large Corpora.* In Proceedings of ACL-2004.

[14] N. Kambhatla. 2004. *Combining lexical, syntactic and semantic features with maximum entropy models.* In Proceedings of ACL-2004.

[15] B. Long, P. S. Yu, and Z. Zhang. 2008. *A general model for multiple view unsupervised learning.* In Proceedings of SDM-2008.

[16] P. Nakov, and M. A. Hearst. 2008. *Solving Relational Similarity Problems Using the Web as a Corpus.* In Proceedings of ACL-2008.

[17] D. P. T. Nguyen, Y. Matsuo and M. Ishizuka. 2007. *Relation extraction from wikipedia using subtree mining.* In Proceedings of AAAI-2007.

[18] X. Ni, Z. Lu, X. Quan, L. Wenyin, and B. Hua. 2009. *Short Text Clustering for Search Results.* In Proceedings of APWeb-WAIM-2009.

[19] P. Pantel and M. Pennacchiotti. 2006. *Espresso: Leveraging generic patterns for automatically harvesting semantic relations.* In Proceedings of ACL-2006.

[20] B. Rosenfeld and R. Feldman. 2006. *URES: an Unsupervised Web Relation Extraction System.* In Proceedings of COLING/ACL-2006.

[21] B. Rosenfeld and R. Feldman. 2007. *Clustering for Unsupervised Relation Identification.* In Proceedings of CIKM-2007.

[22] Y. Shinyama and S. Sekine. 2006. *Preemptive Information Extraction using Unrestricted Relation Discovery* In Proceedings of HLT-NAACL-2006.

[23] M. Zaki. 2002. *Efficiently mining frequent trees in a forest.* In Proceedings of KDD-2002.