

Multi-view Bootstrapping for Relation Extraction by Exploring Web Features and Linguistic Features

Yulan Yan, Haibo Li, Yutaka Matsuo, and 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

ishizuka@i.u-tokyo.ac.jp

Abstract. Binary semantic relation extraction from Wikipedia is particularly useful for various NLP and Web applications. Currently frequent pattern mining-based methods and syntactic analysis-based methods are two types of leading methods for semantic relation extraction task. With a novel view on integrating syntactic analysis on Wikipedia text with redundancy information from the Web, we propose a multi-view learning approach for bootstrapping relationships between entities with the complementary between the Web view and linguistic view. On the one hand, from the linguistic view, linguistic features are generated from linguistic parsing on Wikipedia texts by abstracting away from different surface realizations of semantic relations. On the other hand, Web features are extracted from the Web corpus to provide frequency information for relation extraction. Experimental evaluation on a relational dataset demonstrates that linguistic analysis on Wikipedia texts and Web collective information reveal different aspects of the nature of entity-related semantic relationships. It also shows that our multi-view learning method considerably boosts the performance comparing to learning with only one view of features, with the weaknesses of one view complement the strengths of the other.

1 Introduction

Recent attention to automatically harvesting semantic resources from Wikipedia has encouraged Data Mining researchers to develop algorithms for it. Many efforts have been focused on extracting semantic relations between entities, such as *birth_date* relation, *CEO* relation, and other relations.

Currently one type of the leading methods in semantic relation extraction are based on collecting relational frequency patterns or terms from a local corpus or use the Web as corpus [17]; [15]; [2]; [11]; [4]. Let us call them frequent pattern 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 from collective strings 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 $\langle x, y \rangle$ with *Spouse* relation, string “*x is married to y*” is a textual pattern example. The method is used widely, however, 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 from well-written texts (see e.g., [14]; [5]; [22]). Let us call them syntactic analysis-based methods. Currently, syntactic analysis-based methods for semantic relation extraction are almost all supervised, relying on pre-specification of the desired relationship or hand-coding initial training data. 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 $\langle x, y \rangle$ and the sentence “*x is the wife of y*”, syntactic, dependency features will be generated by analysis of the sentence. One of the main disadvantages is that semantic relations maybe expressed in different dependency/syntactic structures. Moreover, for the heterogeneous text found on the Web, it often runs into problems to apply “deep” linguistic technology.

Syntactic analysis-based methods extract relation instances with similar linguistic features to abstract away from different surface realizations of semantic relations, while frequent pattern mining-based methods group different surface patterns for one relation instance from redundancy Web information are expected to address the data sparseness problem. Wikipedia, unlike the whole Web corpus, as an earlier report [12] explained, Wikipedia articles are much cleaner than typical Web pages, we can use “deep” linguistic technologies, such as syntactic or dependency parsing. Considering the complementary of the strengths and the weaknesses of both two views, we propose a multi-view learning approach for relation extraction from Wikipedia with view disagreement detection which can be advantageous when compared to learning with only a single view. To decide whether two relation instances share the same relationship, a common assumption in multi-view learning is that the samples from each view always belong to the same class. In realistic settings, linguistic-view and Web-view are often corrupted by noise. For example, it happens that dependency parsing for some long sentences will be erroneous. Thus we also consider filtering view corruption which is a source of view disagreement.

In this paper we present a method for performing multi-view learning by filtering view disagreement between linguistic features and Web features. We learn a classifier in a bootstrapping way for each relation type from confident trained instances with view disagreement detected by exploiting the joint view statistics.

The main contributions of this paper are as follows:

- With a novel view on integrating linguistic analysis on Wikipedia text with redundancy information from the Web, we propose a multi-view learning approach for bootstrapping relationships between entities with the complementary between the Web view and linguistic view. From the Web view, related information between entity pairs are collected from the whole Web. From linguistic view, syntactic and dependency information are generated from appropriate Wikipedia sentences.

- Different from traditional multi-view learning approaches for relation extraction task, we filter view disagreement to deal with view corruption between linguistic features and Web features, only confident instances without view disagreement are used to bootstrap learning relations.
- 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. And we conclude that learning with linguistic features and Web features is advantageous comparing to only one view of features.

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

2 Related Work

In this section, we review several past research works that are related to our work, including, frequency pattern mining-based relation extraction, syntactic analysis-based relation extraction and multi-view bootstrapping methods.

The World Wide Web is a vast resource for information. Snowball[1] introduced strategies for generating patterns and extracting tuples from plain-text documents that required only a handful of training examples from users. At each iteration of the extraction process, Snowball evaluated the quality of these patterns and tuples without human intervention, and kept only the most reliable ones for the next iteration. [15] extracted underlying relations among entities from social networks (e.g., person-person, person-location network). They obtained a local context in which two entities co-occur on the Web, and accumulated the context of the entity pair in different Web pages. They defined the context model as a vector of terms surrounding the entity pair. [4] proposed a relational similarity measure, using a Web search engine, to compute the similarity between semantic relations implied by two pairs of words. They represented various semantic relations that exist between a pair of words using automatically extracted lexical patterns. The extracted lexical patterns were then clustered to identify the different patterns that expressed a particular semantic relation. In this paper, motivated by the work of [15] and [4], we extract relational terms and textual pattern from Web contexts as Web view.

Currently syntactic analysis-based relation extraction approaches for semantic relation extraction are almost supervised. Many methods, such as feature-based [14]; [23], tree kernel-based ([20]; [9]) and composite kernel-based ([21]; [22], have been proposed in literature. Zhang et al. (2006)[22] presented a composition kernel to extract relations between entities with both entity kernel and a convolution parse tree kernel. As show in their paper, composition of entity features and structured features outperforms using only one kinds of features. Their work also suggests that structured syntactic information has good predication power for relation extraction and the structured syntactic information can be well captured by the tree kernel. This indicates that the flat and the structured features are complementary and the composite of features is effective

for relation extraction. Motivated by the work of (Zhang et al., 2006), we here generate entity features and tree sub-structure features as linguistic view.

Multi-view learning approaches form a class of semi-supervised learning techniques that use multiple views to effectively learn from partially labeled data. [3] introduced co-training which bootstraps a set of classifiers from high confidence labels. [8] proposed a co-boost approach that optimizes an objective that explicitly maximizes the agreement between each classifier, while [18] defined a co-regularization method that learns a multi-view classifier from partially labeled data using a view consensus-based regularization term. [7] have reported a filtering approach to handle view disagreement, and developed a model suitable for the case where the view corruption is due to a background class.

In this study, we propose a multi-view bootstrapping approach for relation extraction from linguistic and Web views. On the one hand, from the Web view, Web features are generated from the Web redundancy information to provide frequency information. On the other hand, from the linguistic view, syntactic features are generated from Wikipedia sentences by linguistic analysis to abstract information away from surface realizations of texts. Our approach bootstrap learns a classifier for each relation type from confident trained instances by applying Christoudias et al. [7]'s view disagreement detection strategy.

3 Multi-view Bootstrapping

We propose a multi-view bootstrapping approach for relation extraction from Wikipedia based on two views of features - Web features and linguistic features - with view agreement detection strategy.

3.1 Outline of the Proposed Method

The proposed method consists of three steps. In this section, we give a brief overview of each of those steps. The subsequent sections will explain the steps in detail.

Let us assume that we are given a set of entity pairs (X, Y) , the task is to classify all entity pairs into several groups, each of which represent a pre-specified semantic relationship. We first query a Web search engine to find the contexts in which the two entity words co-occur, and extract Web features that express semantic relations between the entity pair. Then we select sentences containing both entity words from Wikipedia articles, generate linguistic features such as dependency sub-structures by parsing the selected sentences using a linguistic parser. Next, since there can be more than one features that express the same semantic relation, we cluster the features to identify the ones that express a particular semantic relation. Finally, we present a multi-view bootstrapping method that learns from confident instances with view disagreement detection.

The approach consists of three steps:

- **Step 1:** Feature Acquisition. For each entity pair, generates linguistic features from corresponding Wikipedia texts using linguistic analysis and extracts Web features from context information by searching the Web.

- **Step 2:** Feature Clustering. Clusters Web feature and linguistic features respectively to identify the ones that express a particular semantic relation. We cluster features to avoid computing the similarities of features during the bootstrapping.
- **Step 3:** Multi-View Bootstrapping. For each relation type, learns a classifier which initially trained from a seed set. During bootstrapping, confidently classified samples in each view are used to label instances in the other views.

3.2 Feature Acquisition

For each entity pair, we generate two kinds of features: linguistic features from Wikipedia texts through linguistic analysis and Web features by searching context information from the Web.

Web Feature Generation. Querying an entity pair using a search engine (e.g. Yahoo!), we characterize the semantic relation between the pair by leveraging the vast size of the Web. Our hypothesis is that there exist some key terms and patterns that provide clues to the relations between entity pairs. From the snippets retrieved by the search engine, we extract relational information of two kinds: ranked relational terms as keywords and surface patterns.

- Relational Terms Collection

To collect relational terms as indicators for each entity pair, we look for verbs and nouns from qualified sentences in the snippets instead of simply finding verbs. Using only verbs as relational terms might engender the loss of various important relations, e.g. noun relations “CEO”, “founder” between a person and a company. Therefore, for each concept pair, a list of relational terms is collected. Then all the collected terms of all concept pairs are combined and ranked using an entropy-based algorithm which is described in [6]. With their algorithm, the importance of terms can be assessed using the entropy criterion, which is based on the assumption that a term is irrelevant if its presence obscures the separability of the dataset. After the ranking, we obtain a global ranked list of relational terms T_{all} for the whole dataset (all the entity pairs). For each entity pair, a local list of relational terms T_{ep} is sorted according to the terms’ order in T_{all} . Then from the relational term list T_{ep} , a keyword t_{ep} is selected for each entity pair ep as the first term appearing in the term list T_{ep} . t_{ep} will be used to generate surface patterns below.

- Surface Pattern Generation

Because simply taking the entire string between two entity words captures an excess of extraneous and incoherent information, we use T_{ep} of each entity pair as a key for surface pattern generation. We classified words into Content Words (CWs) and Functional Words (FWs). From each snippet sentence, two entity words and the keyword t_{ep} is considered to be a Content Word (CW). Our idea of obtaining FWs is to look for verbs, nouns, prepositions, and coordinating conjunctions that can help make explicit the hidden relations between the target nouns.

Table 1. Surface pattern samples for an entity pair

Pattern	Pattern
<i>ep</i> ceo <i>es</i>	<i>es</i> found <i>ep</i>
ceo <i>es</i> found <i>ep</i>	<i>es</i> succeed as ceo of <i>ep</i>
<i>es</i> be ceo of <i>ep</i>	<i>ep</i> ceo of <i>es</i>
<i>ep</i> assign <i>es</i> as ceo	<i>ep</i> found by ceo <i>es</i>
ceo of <i>ep</i> <i>es</i>	<i>ep</i> found in by <i>es</i>

Surface patterns have the following general form.

$$[\text{CW1}] \text{Infix}_1 [\text{CW2}] \text{Infix}_2 [\text{CW3}] \quad (1)$$

Therein, Infix_1 and Infix_2 respectively contain only any number of FWs. A pattern example is “*ep* assign *ep* as ceo (*keyword*)”. All generated patterns are sorted by their frequency, and all occurrences of the principle entity and the second entity are replaced with “*ep*” and “*es*”, respectively for pattern matching of different entity pairs.

Table 1 presents examples of surface patterns for a sample entity pair. Pattern windows are bounded by CWs to obtain patterns more precisely because 1) if we use only the string between two entity words, it may not contain some important relational information, such as “ceo *ep* resign *es*” in Table 1; 2) if we generate patterns by setting a windows surrounding two entity words, the number of unique patterns is often exponential.

Linguistic Feature Extraction. We select sentences from Wikipedia articles containing both entities. We define the composite feature vector with flat and the structured features generated from these sentences by using a syntactic parser.

– Flat Features

Using a syntactic parser (Connexor¹), rich linguistic tags can be extracted as features for each entity in an entity pair. For each pair of entities, we extract the following syntactic features as flat features:

- Morphology Features: tells the details of word forms used in text. Connexor Parser defines 70 morphology tags such as *N*(noun), *NUM* (numeral) .
- Syntax Features: describes both surface syntactic and syntactic function information of words. For example, *%NH* (nominal head) and *%>N* (determiner or pre-modifier of a nominal) are surface syntactic tags, *@SUB* (Subject) and *@F-SUBJ* (Formal subject) are syntactic function tags.
- Structure Features

To obtain structured features for an entity pair, we generate dependency patterns. After preprocessing, selected sentences that contain at least one mention of both entity words are parsed into dependency structures. We define dependency patterns as sub-paths of the shortest dependency path between an entity pair for two reasons. One is

¹ www.connexor.com

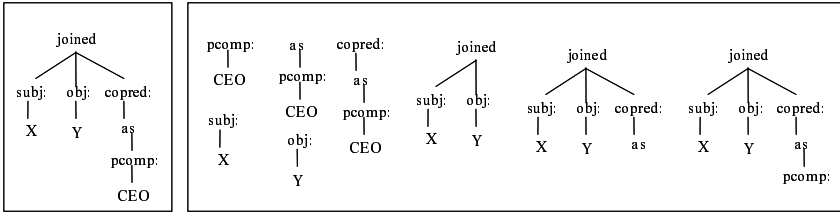


Fig. 1. Example showing how to generate dependency patterns for an entity pair

that the shortest path dependency kernels outperform dependency tree kernels by offering a highly condensed representation of the information needed to assess their relation [5]. The other reason is that embedded structures of the linguistic representation are important for obtaining good coverage of the pattern acquisition, as explained in [9]; [22]. The process of inducing dependency patterns has two steps, as shown in Fig. 1.

1. Shortest dependency path induction. From the original dependency tree structure by parsing the selected sentence for each entity pair, we first induce the shortest dependency path from the Wikipedia sentence with the pair of entity words, as shown in the left side of Fig. 1.

2. Dependency pattern generation. We use a frequent tree-mining algorithm [19] to generate sub-paths as dependency patterns from the shortest dependency path, as shown in the right side of Fig. 1.

3.3 Feature Clustering

A semantic relation can be expressed using more than one pattern. When we compute the relational similarity between two entity pairs, it is important to know whether there is any correspondence between the sets of patterns extracted for each entity pair. If there are many related patterns between two entity pairs, we can expect a high relational similarity. To find semantically related lexical patterns for each view, we apply Sequential pattern clustering algorithm in [4] by using distributional hypothesis [13]. Distributional hypothesis claims that words that occur in the same context have similar meanings.

Given a set P of patterns and a clustering similarity threshold, their algorithm returns clusters (of patterns) that express similar semantic relations. First, their algorithm sorts the patterns into descending order of their total occurrences in all word pairs. Next, it repeatedly takes a pattern p_i from the ordered set P , finds the cluster that is most similar to p_i . To compute the similarity between a pattern and a cluster, first they represent a cluster by the vector sum of all entity pair frequency vectors corresponding to the patterns that belong to that cluster. Next, they compute the cosine of the angle between the vector that represents the cluster (c_j), and the word-pair frequency vector of the pattern (p_i). The sequential nature of their algorithm avoids pairwise comparisons among all patterns. Moreover, sorting the patterns by their total word-pair frequency prior to clustering ensures that the final set of clusters contains the most common relations in the data-set.

3.4 Multi-view Bootstrapping with View Disagreement Detection

In this section we present a multi-view bootstrapping algorithm that uses the idea of view disagreement detection. We apply (Christoudias, et al., 2008)[7]’s conditional view entropy measure to detect and filter entity pairs with view disagreement in a pre-filtering step.

Multi-view learning can be advantageous when compared to learning with only a single view especially when the weaknesses of one view complement the strengths of the other. A common assumption in multi-view learning is that the samples from each view always belong to the same class. In realistic settings, datasets are often corrupted by noise. Thus we need to consider view disagreement caused by view corruption. We apply the method in (Christoudias, et al., 2008)[7] for Multi-view Bootstrapping by learning a classifier in one view from the labels provided by a classifier from another view with a view disagreement strategy. Their Method consists of two steps:

- **Step 1:** View disagreement detection. Detect and filter entity pairs with view disagreement using an information theoretic measure based on conditional view entropy.
- **Step 2:** Multi-view Bootstrapping. Mutually train a set of classifiers, on an unlabeled dataset by iteratively evaluating each classifier and re-training from confidently classified entity pairs.

Firstly, to detect view disagreement, they use conditional entropy $H(x|y)$ which is a measure of the uncertainty in x given that we have observed y . In the multi-view setting, the conditional entropy between views, $H(x_i|x_j)$, can be used as a measure of agreement that indicates whether the views of a sample belong to the same class or event. Under the assumptions the conditional view entropy is expected to be larger when conditioning on entity pairs with disagreement compared to those without disagreement. When computing the conditional entropy between views, we use the pattern clusters to replace features when measuring the conditional entropy between views so we can avoid computing the distance between two similar patterns.

Secondly, with the conditional entropy measure, we mutually train a set of classifiers for each relation type, on an unlabeled dataset iteratively evaluating each classifier and re-training from confidently classified samples. In the presence of view disagreement, we detect classified samples which are not in view disagreement. Only those detected classified samples are used to train classifiers iteratively. During bootstrapping, confidently classified samples in each view are used to label corresponding samples in the other views.

4 Experiments

In this section, we evaluate our multi-view bootstrapping approach on the relation extraction from Wikipedia, and show the effectiveness of the proposed approach.

4.1 Experimental Setup

We conduct our experiments on relation extraction task using the dataset that was created for evaluating relation extraction from Wikipedia in [10]. This data contains

Table 2. Overview of the dataset

relation	#of Instances	Instance samples for each relation type
job_title	216	(Charles Darwin, naturalist), (Jack Kerouac, novelist)
birth_year	157	(Hillary Clinton, 1947), (George H. W. Bush, 1924)
education	106	(James Bowdoin, Harvard), (Franklin Schaffner, Columbia University)
death_year	104	(Abraham Lincoln, 1865), (James Bowdoin, 1790)

Wikipedia pages for which links between pages have been annotated with a relation type, e.g. *birth_year*, *education*, *nationality*, etc. We evaluate on a subset which consists of four relation types *job_title*, *birth_year*, *education*, *death_year*. For each relation type, in Table 2, we show some of the instances and the total number of entity pairs. Each entity pair in the dataset has one accompanying sentence from a Wikipedia article.

We build three baseline systems on the dataset. One baseline system is built by semi-supervised learning from only the linguistic view which shows the performance of learning with only linguistic features. Another system is built by learning from only the Web view which shows the performance of learning with Web features. We also evaluate on bootstrap learning from the linguistic view and Web view without view disagreement detection in a traditional way.

To evaluate the performance of our approach, we run the feature generation algorithm described in section 3.2 for each entity pair in our dataset to extract Web features and linguistic features. We collect Web features through querying with each pair of entity words by a search engine (We use Yahoo, the top 1000 snippets are downloaded as collective context). We collect relational terms and textual patterns as Web features by 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 by a linguistic parser. We collect entity features for each entity word and generate dependency patterns as sub-paths from the shortest dependency path containing two entities by making use of a frequent tree-mining algorithm [19].

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.

$$\text{Precision } (P) = p'/p \quad \text{Recall } (R) = p'/n$$

$$F\text{-value } (F) = 2 * P * R / (P + R)$$

4.2 Feature Clusters

We use the clustering algorithm described in Section 3.3 to cluster the extracted Web features and linguistic features respectively.

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

Table 3. Examples of frequent Web features from Web feature clustering

ep was a es	ep was elected es	ep was the es	ep was the leading es
ep was born in es	ep born in es	ep born D es	ep was born on es
es graduate ep	ep graduated from es	ep is a graduate of es	ep graduated from the es
ep died es	ep died in D es	ep who died in es	ep who died in D es

Table 4. Examples of frequent features from linguistic feature clustering

(be(ep))	(be(es))	(mainroot(be(es)))	(be(ep)(es))
(bear(die))	(bear(be)(die))	(mainroot(bear(die)))	(bear(be(ep))(die))
(graduate(ep))	(mainroot(graduate(ep)))	(mainroot(graduate))	(graduate(ep)(from))
(attend(ep))	(attend(ep)(es))	(mainroot(attend(ep)(es)))	(mainroot(attend(ep)))
(bear(es))	(bear(be)(in))	(bear(be(ep)))	(bear(in))

features that express a specific semantic relation. *ep* and *es* 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.

4.3 Empirical Analysis

Table 5 presents the overall evaluation of the comparison of our approach and three baseline systems. The first two columns of results show learning with only one view of features respectively: linguistic view, Web view. It shows that the performance of using Web features is better than using linguistic features. Moreover, by applying traditional bootstrapping method with Web features and linguistic features without view disagreement detection, the performance is even better. It means Web features and linguistic features provide different information for the relation extraction task. The final column shows using multi-view bootstrapping approach with view disagreement detection, the performance is improved over traditional bootstrapping approach. It means that by dealing with view corruption, relations can be learned with better reliability from confident samples.

We also compared the above three baseline systems with our proposed method for the four relation types, shown in Table 6. Using only linguistic features, the performance is much worse than Web views for some relationships, such as “*birth_year*”. A closer look into the features extracted for some entity pairs reveals that some instances which belong to different relation types are often described in the same Wikipedia sentence. This kind of sentences are often hard to be parsed in an appropriate way to generate the correct linguistic features. For Example, “*Aldous Leonard Huxley (July 26, [[1894]] C November 22, [[1963]]) was a British [[writer]]*” is the Wikipedia sentence containing instances of relations *job_title*, *birth_year*, *death_year*.

Table 5. Overall evaluation over different methods

	Single-View Learning		Multi-View Bootstrapping	
	Linguistic Feature	Web Feature	Traditional	Proposed
Pre	46.30	51.80	54.14	68.19
Rec	40.82	47.00	51.03	63.95
F1	43.39	49.28	52.54	66.00

Table 6. Evaluation on each relation type over different methods

Relation	Single-View Learning						Multi-view Bootstrapping					
	Linguistic-View			Web-View			Traditional			Proposed		
	Pre.	Rec.	F-v.	Pre.	Rec.	F-v.	Pre.	Rec.	F-v.	Pre.	Rec.	F-v.
job_title	69.82	54.63	61.30	66.20	21.76	32.75	69.75	52.31	59.79	91.18	57.41	70.45
birth_year	21.43	15.29	17.84	40.00	53.50	45.78	43.38	37.58	40.27	57.71	64.33	60.84
education	56.52	12.26	20.16	52.63	47.17	49.75	42.39	36.79	39.40	69.57	60.38	64.65
death_year	39.52	79.81	52.87	60.78	89.42	72.37	48.19	89.42	62.63	53.33	92.31	67.61
overall	46.30	40.82	43.39	51.80	47.00	49.28	54.14	51.03	52.54	68.19	63.95	66.00

All the experimental results support our idea mainly in three main ways: 1) the combination of Web features and linguistic features is effective in relation extraction task; 2) It has been shown that multi-view bootstrapping is advantageous to learning with only a single view when the weaknesses of one view complement the strengths of the other. 3) the detection and filtering of view disagreement considerably increases the performance of traditional multi-view learning approaches.

5 Conclusions

We propose a multi-view learning approach for bootstrapping relationships between entities from Wikipedia with the complementary between the Web view and linguistic view. From Web view, related information for entity pairs are collected from the whole Web. From linguistic Web, analysis information from sentences are generated from Wikipedia sentences. We filter view disagreement to deal with view corruption between linguistic features and Web features, with only confident trained instances used for classifiers. Experimental evaluation on a relational dataset demonstrates that linguistic analysis and Web collective information reveal different aspects of the nature of entity-related semantic relationships. Our multi-view learning method considerably boosts the performance comparing to learning with only one view, with the weaknesses of one view complement the strengths of the other. This study suggests an example to bridge the gap between Web mining technology and “deep” linguistic technology for information extraction tasks.

References

1. Agichtein, E., Gravano, L.: Snowball: Extracting relations from large plain-text collections. In: Proceedings of the Fifth ACM International Conference on Digital Libraries (2000)
2. Banko, M., Cafarella, M.J., Soderland, S., Broadhead, M., Etzioni, O.: Open information extraction from the Web. In: Proceedings of IJCAI 2007 (2007)

3. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of COLT 1998 (1998)
4. Bollegala, D., Matsuo, Y., Ishizuka, M.: An Integrated Approach to Measuring the Similarity between Implicit Semantic Relations from the Web. In: Proceedings of WWW 2009 (2009)
5. Bunescu, R., Mooney, R.: A shortest path dependency kernel for relation extraction. In: Proceedings of HLT/EMLNP 2005 (2005)
6. Chen, J., Ji, D., Tan, C.L., Niu, Z.: Unsupervised Feature Selection for Relation Extraction. In: Proceedings of IJCNLP 2005 (2005)
7. Christoudias, C., Urtasun, R., Darrell, T.: Multi-view learning in the presence of view disagreement. In: Proceedings of UAI 2008 (2008)
8. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (1999)
9. Culotta, A., Sorensen, J.: Dependency tree kernels for relation extraction. In: Proceedings of the ACL 2004 (2004)
10. Culotta, A., McCallum, A., Betz, J.: Integrating probabilistic extraction models and data mining to discover relations and patterns in tex. In: Proceedings of the HLT-NAACL 2006 (2006)
11. Davidov, D., Rappoport, A.: Classification of Semantic Relationships between Nominals Using Pattern Clusters. In: Proceedings of ACL 2008 (2008)
12. Giles, J.: Internet encyclopaedias go head to head. *Nature* 438, 900–901 (2005)
13. Harris, Z.: Distributional structure. *Word* 10, 146–162 (1954)
14. Kambhatla, N.: Combining lexical, syntactic and semantic features with maximum entropy models. In: Proceedings of ACL 2004 (2004)
15. Mori, J., Tsujishita, T., Matsuo, Y., Ishizuka, M.: Extracting Relations in Social Networks from the Web using Similarity between Collective Contexts. In: Proceedings of ISWC 2006 (2006)
16. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of cotraining. In: Workshop on Information and Knowledge Management (2000)
17. Pantel, P., Pennacchiotti, M.: Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In: Proceedings of ACL 2006 (2006)
18. Sindhwani, V., Niyogi, P., Belkin, M.: A coregularization approach to semi-supervised learning with multiple views. In: Proceedings of ICML 2005 (2006)
19. Zaki, M.: Efficiently mining frequent trees in a forest. In: Proceedings of KDD 2002 (2002)
20. Zelenko, D., Aone, C., Richardella, A.: Kernel Methods for Relation Extraction. *Journal of Machine Learning Research* 3, 1083–1106 (2003)
21. Zhao, S., Grishman, R.: Extracting relations with integrated information using kernel methods. In: Proceedings of ACL 2005 (2005)
22. Zhang, M., Zhang, J., Su, J., Zhou, G.: A Composite Kernel to Extract Relations between Entities with both Flat and Structured Features. In: Proceedings of ACL 2006 (2006)
23. Zhou, G., Su, J., Zhang, J., Zhang, M.: Exploring various knowledge in relation extraction. In: Proceedings of ACL 2005 (2005)