Unsupervised Relation Extraction by Mining Wikipedia Texts supported with Web Redundancy Information

Yulan Yan[*1] Mitsuru Ishizuka[*1] Yutaka Matsuo[*1]

[*1] The University of Tokyo

In this study, we propose an unsupervised relation extraction approach based on a combination of two types of patterns. On one hand, surface patterns are generated from the Web corpus to provide redundancy information for relation extraction. On the other hand, dependency patterns are generated to abstract away from different surface realizations of semantic relations. Wikipedia which widely used as a corpus for information extraction is used as a local corpus and the Web is used as a global corpus. From the experimental analysis, we conclude that dependency patterns have the properties of being more accurate and less spam-prone than surface patterns from the Web corpus, while the redundancy information can be used to ease the data sparseness problem.

1. Introduction

The machine learning approaches for relation extraction task require substantial human effort, particularly when applied to the broad range of documents, entities, and relations existing in the Web. Even with semi-supervised approaches which use a large unlabeled corpus, manual construction of a small set of seeds known true instances of the target entity or relation is susceptible to arbitrary human decisions. Hence, there is a need for developing semantic information retrieval algorithms that are as unsupervised as possible.

Currently the leading methods in unsupervised information extraction are based on collecting redundancy information from a local corpus or use the Web as corpus (Banko et al., 2007; Bollegala et al., 2007; Fan et al., 2008; Davidov and Rappoport, 2008). The standard process is to scan or search the corpus to collect co-appearances of word pairs with strings between them, then calculate term co-occurrence or generate textual patterns. 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 exponential 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.

Linguistic analysis is another effective technology for semantic relation extraction (see e.g., (Kambhatla, 2004; Bunescu et al., 2005; Harabagiu et al., 2005; Nguyen et l., 2007)). Currently, linguistic approaches for semantic relation extraction are almost exclusively supervised, relying on pre-specification of the desired relationship or hand-coding initial seed words or patterns. The common process is to generate linguistic patterns based on analysis of the syntactic, dependency or shallow semantic structure of text, then train to identify entity pairs which assume a relationship and classify them into pre-defined relationships. The advantage of these methods is using linguistic technologies to learn semantic information from different surface expressions.

Contact: 7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan. yulan@mi.ci.i.u-tokyo.ac.jp

In this paper, we consider of integrating linguistic analysis with redundancy Web information to improve the performance of unsupervised relation extraction. As some work (Banko et al., 2007) claimed, "heavy" linguistic technology runs into problems when applied to the heterogeneous text found on the Web. Therefore, we do not plan to parse information from the Web corpus, but from good-written texts. In particular, we focus on natural occurring texts of Wikipedia articles. We are interested in extracting relations from Wikipedia articles. A fundamental type of Wikipedia resource is that of concepts (represented by Wikipedia articles) and relations between concepts. We propose our approach in which concept pairs are clustered into a number of clusters based on the similarity of their contexts. Contexts are collected as two kinds of patterns: dependency patterns from dependency analysis of sentences in Wikipedia, surface patterns generated from Web information through a Web search engine.

The remainder of the paper is organized as follows. In section 2 we will present out the overview of our approach and describe it in detail. In section 3 we will report on our exploratory experimental results. Finally, in section 4 we will conclude the paper.

2. Pattern Combination Approach for Relation Extraction

We propose a solution in the following way. The tuition idea is the combination of linguistic features and Web features. On one hand we apply linguistic technologies on high-quality text in Wikipedia, on the other hand, we apply web mining technologies on large scaled Web corpus. In this section, we firstly provide the overview of our approach along with the function of the main modules. Secondly, we explain each module in the approach in details.

2.1 Overview of the Approach

Our approach requires a set of Wikipedia articles as input; for each article, outputs a list of concept pairs with relation labels.

As shown in Figure 2, there are four primary modules of our approach:

- **Text Preprocessor and Relation Candidate Generation**, the module which preprocesses Wikipdia articles to split text and filter sentences. For each article, it collects a set of concept pairs as relation candidates.
- **Web Context Collector**, the module which collects context information from the Web corpus. For each relation candidate, it generates a set of ranked relational terms and a set of surface patterns.
- **Dependency Pattern Modeling**, the module which generates dependency patterns for each relation candidates from corresponding sentences in Wikipedia articles.
- **Linear Clustering Algorithm**, the module which clusters relation candidates based on their context. It contains two sub-modules:
  - **Local Clustering**, which merges instances using only dependency patterns generated from Dependency Pattern Modeling;
  - **Global Clustering**, which clusters instances using only textural patterns generated from Web Context Collector, based on the resulted clusters of local clustering. The aim is to merge more instances into existed clusters with surface patterns to improve the coverage of clusters.

The key to our approach lies in the complementary of Web frequency information from the Web and deep linguistic analysis, i.e., use linguistic information to extract relation instances with good precision and use Web frequency information to improve the coverage of relation instances. Below we focus on the Linear Clustering Algorithm module.

2.2 Linear Clustering Algorithm

In this subsection, we present an unsupervised relation clustering algorithm to merge concept pairs based on two kinds of generated patterns: dependency pattern and surface patterns. We proposed our linear clustering algorithm based on k-means clustering for relation clustering.

The dependency pattern modeling module has the properties of being more accurate, less spam-prone, but Web context has the advantage of containing much more redundant information than the Wikipedia. Our idea of relation instance clustering is: first cluster instance into clusters with good precision using dependency patterns in a local clustering step; then improve the coverage of clusters with more instances by using surface patterns in a global clustering step.

2.2.1 Distance Function and Initial Centroid Selection

The standard k-means algorithm depends on the choice of the seeds and on the number k of clusters, which must be known in advance. However, as we claimed in the introduction section, to extract relations from Wikipedia articles in an unsupervised way, cluster number k is unknown and no good centroids can be predicted. In this paper, we base the selection of centroids on the keyword t_{cp} generated from the Web Context Collector module of each concept pair.

Firstly, all concept pairs are grouped by their keywords t_{cp}. Let $G = \{G_1, G_2, ...G_n\}$ be the resulting groups, where each $G_i = \{cp_{i1}, cp_{i2}, ...\}$ identify a group of concept pairs who share the same keyword t_{cp} (such as "CEO"). With all the groups ranked by their number of instances, we choose the top k groups, and a centroid c_i is selected for each group G_i as follows:

$$s = \arg\max_{1\le s\le |G_i|} |\{cp_{ij}|(\alpha * dis_1(cp_{ij}, cp_{is}) + \beta * dis_2(cp_{ij}, cp_{is})) <= D_z, 1 \le j \le |G_i|\}| \quad (1)$$

$$c_i = cp_{is} \quad (2)$$

where

$$dis_1(cp_{ij}, cp_{is}) = 1 - \frac{|DP_{cp_{ij}} \cap DP_{cp_{is}}|}{\sqrt{(|DP_{cp_{ij}}| * |DP_{cp_{is}}|)}} \quad (3)$$

To compute distance over surface patterns, we implement the distance function $dis_2(cp_{ij}, cp_{is})$ in Figure 3.

```
Algorithm 1: distance function $dis_2(cp_{ij}, cp_{is})$
Input: $SP_1 = \{sp_{11}, ..., sp_{1m}\}$(surface patterns of $cp_{ij}$ )
$SP_2 = \{sp_{21}, ..., sp_{2n}\}$ (surface patterns of $cp_{is}$)
Output: $dis$ (distance between $SP_1$ and $SP_2$)
define a $m \times n$ similarity matrix A: $\{A_{ij} = cost(sp_{1i}, sp_{2j})$
1≤i≤m; 1≤j≤n};
$dis = 1$
for $\min(m, n)$ times do
    (x, y) = $\text{argmax}_{0<i<m;0<j<n} A_{ij}$;
    $dis$ = $dis$ - $A_{xy}$;
    $A_{x*} = 0$; $A_{*y} = 0$;
return $dis$
```

Figure 1: Distance function over surface patterns

We selected the centroid to be the instance which has the most of other instances in the same group that have distance less than D_z with it. D_z is a threshold to avoid noisy instances, we assign it 1/3. dis_1 is the distance function to calculate distance between dependency pattern lists $DP_{cp_{ij}}, DP_{cp_{is}}$ of two concept pairs. The distance is decided by the number of shared dependency patterns. When computing dis_2, for cost function $cost(sp_{1i}, sp_{2j})$ which is used to calculate the similarity of two surface patterns, we use the dynamical programming computing which described in detail in (Rosenfeld and Feldman, 2006). α and β are used to leverage between dependency patterns and surface patterns, in this work we assign them both 1/2.

As for estimating the number of clusters, in this work we apply the stability-based criterions from (Chen, et al, 2005) to decide the number k of optimal clusters.

2.2.2 Local Dependency Pattern Clustering

The purpose of this stage is that given the initial seed instances and cluster number k, to merge relation instances over dependency patterns into k clusters. Each concept pair cp_{ij} has a set of dependency patterns $DP_{cp_{ij}}$, we calculate distances between two pairs cp_{ij} and cp_{is} with above the function $dis_1(cp_{ij}, cp_{is})$. The clustering algorithm is shown in Figure 4.

```
Algorithm 2: localClustering
Input: $I = \{cp_1, ..., cp_n\}$(all instances)
       $C = \{c_1, ..., c_k\}$ (k initial centroids)
Output: $m_{loc} : I \rightarrow C$ (cluster membership)
        $I_{rest}$ (rest of instances not clustered)
        $C_{loc} = \{c_1, ..., c_k\}$ (recomputed centroids)
for each $cp_i \in I$ do
    if $\min_{s\in 1..k} dis_1(cp_i, c_s) <= D_l$ then
        $m_{loc}(cp_i) = \mathrm{argmin}_{s\in 1..k}\, dis_1(cp_i, c_s)$
    else
        $m_{loc}(cp_i) = 0$; $I_{rest} \leftarrow cp_i$
for each $j \in \{1..k\}$ do
    recompute $c_j$ as the centroid of
        $\{cp_i | m_{loc}(cp_i) = j\}$
```

Figure 2: Clustering with dependency patterns

Since there are many concept pairs which are scattered and actually do not belong to any of the top k clusters, we filter concept pairs that with distance larger than D_l with the seed instances, to make sure the precision of the clustering. The rest instances not clustered are stored in I_{rest}. After this step, relation instances with similar dependency patterns are merged into same clusters.

2.2.3 Global Surface Pattern Clustering

One major problem faced by the local clustering is the fact that instances of the same semantic relationship which are represented in different dependency structures will not be merged into the same cluster.

In this step of clustering, we use surface patterns to merge more instance for each cluster to improve the coverage performance using the algorithm shown in Figure 6.

```
Algorithm 3: globalClustering
Input: $I_{rest}$ (rest of instances)
       $C_{loc} = \{c_1, ..., c_k\}$ (initial centroids)
Output: $m_{glo} : I_{rest} \rightarrow C$ (cluster membership)
        $C = \{c_1, ..., c_k\}$ (final centroids)
for each $cp_i \in I_{rest}$ do
    if $\min_{s\in 1..k} dis_2(cp_i, c_s) <= D_g$ then
        $m_{glo}(cp_i) = \mathrm{argmin}_{s\in 1..k}\, dis_2(cp_i, c_s)$
    else
        $m_{glo}(cp_i) = 0$
for each $j \in 1..k$ do
    recompute $c_j$ as the centroid of cluster
        $\{cp_i | m_{loc}(cp_i) = j \vee m_{glo}(cp_i) = j\}$
return clusters C
```

Figure 3: Clustering with surface patterns

Each concept pair has a set of surface patterns from the Web context collector module, to measure the distance between two instances, we use the distance function dis_2 explained in the above section. Also, we filter concept pairs with distance larger than D_g with the seed instances, to make sure the precision of the clustering.

Finally we have k clusters of instances, each cluster has a centroid instance. To attach a single relationship label each cluster, we use the centroid instance and assign the keyword of it as the relation label.

3. Experiments

In this section we wish to consider the variety of relations that can be generated by our approach from Wikipedia, and to measure the quality of these relations in terms of their precision and coverage. To balance between precision and coverage of clustering, our approach uses several parameters: D_l, D_g, k. For purposes of evaluation, we ran our algorithm on the category - "American chief executives".

The performance of the proposed approach is evaluated on different pattern types: dependency patterns, surface patterns and the combination of both. We compare our approach with (Rosenfeld and Feldman, 2007)'s URI method, which showed that their algorithm improved over previous work using two kinds of surface features for unsupervised relation extraction: features that test two entities together and features that test only one slot each. For the purpose of comparison, we use k-means clustering algorithm and choose the same k as our approach when applying their approach.

3.1 Wikipedia Category: "American chief executives"

In the first series of experiments we build a development phrase to select appropriate D_l (instance filter in local clustering) and D_g(instance filter in global clustering). To balance between precision and coverage, we assign 1/3 for both D_l and D_g.

526 articles in this category are used for developing. We get 7310 concept pairs from the articles as our data set. Top 18 Groups are chosen to get the centroid instances. Of these, 15 binary relations are clearly identifiable relations which are shown in Table 2, where # Ins. represents the number of instances clustered by each method, and *pre* shows the precision of each cluster.

The proposed approach shows the higher precision and better coverage than URI in Table 2. This demonstrates that adding dependency patterns from linguistic analysis contribute greatly to the precision and coverage of the clustering task than using only surface patterns.

For further view of the contribution of dependency pattern, we experiment with dependency pattern separately to compare the results with using only surface patterns or combined patterns. The results are shown in Table 3. The best precision is achieved with dependency patterns, significantly better than surface patterns, though the coverage is the lowest, showing the sparseness of dependency patterns. The coverage is evaluated as the proportion of correctly extracted instances to the whole data set (7310 concept pairs). We use coverage as a relatively evaluation instead of using

Table 1: Results on the category: "American chief executives"

method	Existing method (Rosenfeld et al.)		Proposed method (Our method)	
Relation (sample)	# Ins.	pre	# Ins.	pre
chairman (*x be chairman of y*)	434	63.52	547	68.37
ceo (*x be ceo of y*)	396	73.74	423	77.54
bear (*x be bear in y*)	138	83.33	276	86.96
attend (*x attend y*)	225	67.11	313	70.28
member (*x be member of y*)	14	85.71	175	91.43
receive (*x receive y*)	97	67.97	117	73.53
graduate (*x graduate from y*)	18	83.33	92	88.04
degree (*x obtain y degree*)	5	80.00	78	82.05
marry (*x marry y*)	55	41.67	74	61.25
earn (*x earn y*)	23	86.96	51	88.24
award (*x won y award*)	23	43.47	46	84.78
hold (*x hold y degree*)	5	80.00	37	72.97
become (*x become y*)	35	74.29	37	81.08
director (*x be director of y*)	24	67.35	29	79.31
die (*x die in y*)	18	77.78	19	84.21
all	1510	68.27	2314	75.63

recall as measure, since for unsupervised task, it is difficult to evaluate with the recall. As we claimed, there are many concept pairs which are scattered and actually do not belong to any of the top k clusters, thus the coverage is low.

All the experimental results support our idea mainly in two aspects: 1) dependency analysis can abstract away from different surface realizations of text and embedded structures of the dependency representation are important for obtaining a good coverage of the pattern acquisition. And the precision is better than string surface patterns from various kinds of Web pages; 2) surface patterns are used to merge instances with relations represented in different dependency structures with redundancy information from the vast size of Web pages, by using surface patterns, more instances can be clustered, the coverage is improved.

4. 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 patterns generated from large corpus, and use Web frequency information to easy the sparseness of linguistic information. In particular, we focus on natural occurring texts from Wikipedia articles. Relations are gathered in an unsupervised way over two types of patterns: dependency patterns by parsing sentences in Wikipedia articles using a linguistic parser, and surface patterns from redundancy information from the Web corpus by using a search engine. We report our experimental results comparing to previous work and evaluating over using different patterns. The results show that the performance is the best with the combination of dependency patterns and surface patterns.

Table 2: Performance of different pattern types

Pattern type	#Instance	Precision	Coverage
dependency	1127	84.29	8.63%
surface	1510	68.27	9.39%
Combined	2314	75.63	15.81%

References

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

[Chen et al., 2005] . Chen, D. Ji, C.L. Tan, and Z. Niu: Unsupervised Feature Selection for Relation Extraction, In Proceedings of IJCNLP-2005.

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

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

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

[Dhillon:2003] . S. Dhillon, S. Mallela, and D. S. Modha: Information-theoretic co-clustering, In Proceedings of SIGKDD-2003.

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

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

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

[Rosenfeld:2007] . Rosenfeld and R. Feldman: Clustering for Unsupervised Relation Identification, In Proceedings of CIKM-2007.