3D2-2

Relation Extraction using Relational Duality

Danushka Bollegala*1 Mitsuru Ishizuka*1

*1The University of Tokyo

A relation can be expressed *extensionally* by stating all the instances of that relation or *intensionally* by defining all the paraphrases of that relation. For example, consider the ACQUISITION relation between two companies. An extensional definition of ACQUISITION contains all pairs of companies in which one company is acquired by another (e.g. (*YouTube*, *Google*) or (*Powerset*, *Microsoft*)). On the other hand we can intensionally define ACQUISITION as the relation described by lexical patterns such as ***X** is acquired by **Y***, or ***Y** purchased **X***, where **X** and **Y** denote two companies. We use this dual representation of semantic relations to propose a novel sequential co-clustering algorithm that can extract numerous relations efficiently from unlabeled data and evaluate its performance on measuring relational similarity.

1. Introduction

Correctly identifying the relationships between entities is an important step for numerous tasks on the Web such as information retrieval and information extraction. A semantic relation that exists between two given objects (e.g., concepts, words, or named-entities) can be defined in two ways [8]: *extensionally* or *intensionally*. An extensional definition of a concept formulates its meaning by specifying *every object* that falls under the definition of the concept. On the other hand, an intensional definition of a concept formulates its meaning by specifying *all the properties* that are necessary to reach that definition. For example, consider the ACQUISITION relation between two companies. An extensional definition of the ACQUISITION relation enumerates all pairs of entities between which an ACQUISITION relation holds (e.g. (*You Tube*,*Google*), (*Powerset*, *Microsoft*), etc.) Alternatively, we can express the ACQUISITION relation intensionally by stating the different ways that we can express an acquisition between two companies **X** and **Y** such as ***X** is acquired by **Y***, ***X** is purchased by **Y***, or ***X** is bought by **Y***. Because both extensional and intensional definitions describe the same semantic relation, there exist a *duality* between the two definitions. We exploit this duality in semantic relations to propose a co-clustering algorithm to extract relations from a given text corpus.

Despite its numerous applications, extracting semantic relations among entities at Web scale is challenging for several reasons. First, a single semantic relation can be expressed using multiple lexical patterns. For example, aside from the pattern ***X** acquired **Y***, an acquisition between two companies **X** and **Y** can be expressed using patterns such as ***X** purchased **Y***, ***X** completed its acquisition of **Y***, etc. Second, there might exist more than one semantic relation between a pair of entities. For example, before an ACQUISITION relation is established between two companies, those companies can have a COMPETITOR relation. A relation extraction system must discover the different relations that hold between a pair of entities. Third, the entities themselves might have variants. For example, Microsoft Corp. is often designated as *the Redmond software giant*. Manually specifying all different name variants of an entity is not feasible. Moreover, the scale and the heterogeneity of Web text prohibit the use of time-consuming, domain-specific approaches that require deep language processing techniques. Supervised approaches to relation extraction that require manual annotation of all relations to be extracted are costly and impossible to execute on a Web scale because we do not know in advance the number or the types of relations that we must extract from the Web. Semi-supervised approaches to relation extraction require some seed instances (i.e., a few pairs of entities between which the desired relation exist) or extraction patterns (either domain specific or independent) to be provided by a human. Unfortunately, the quality of the extracted relations depends heavily on the initial seeds given to the system. Moreover, it is not clear how many seeds are necessary to extract a particular relation correctly beforehand.

連絡先: 7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan.
danushka@iba.t.u-tokyo.ac.jp

2. Related Work

Traditionally, relation extraction is framed as a binary classification problem: Given a sentence S and a relation R, does S assert R between two entities in S? Supervised classification methods such as support vector machines (SVMs) with language-oriented kernels have been used to learn binary classifiers [7, 11, 12, 17, 18]. Roth and Yih [14] present a classification-based framework in which they jointly learn to identify named entities and relations. Culotta et al. [9] model the problem of relation extraction as a one of sequence labeling and used conditional random fields to identify the relations in a given document. Specifically, they perform relation extraction on biographical text in which the topic of each document is known in advance. Then, for each entity found in a document, their goal is to predict the relation between that entity and the topic of the document from a finite set of pre-defined relations. In our setting however, we do not know the relations that must be extracted beforehand. Moreover, the need for manually annotated training data by these supervised relation extraction systems makes it difficult to apply them to large-scale heterogeneous relation extraction tasks such as relation ex-

traction from the web.

Bootstrapping methods [1, 6, 10, 13, 19] to relation extraction are attractive because they require markedly fewer training instances than supervised approaches do. Bootstrapping methods are initialized with a few instances (often referred to as seeds) of the target relation [1, 13, 19] or general extraction templates [10]. During subsequent iterations of the bootstrapping process, new extraction patterns are discovered and used to extract new instances. The quality of the extracted relations depends heavily upon the initial seeds provided to the bootstrapping system. If the extracted relations are of low quality, then we must restart with a different set of seeds and re-run the bootstrapping process. It might not be readily apparent to a non-expert user to devise good seeds of the target relation. Moreover, in a setting such as ours, in which we attempt to process a heterogeneous corpus such as Web text, it is not possible to know the target relations in advance, let alone provide seeds or extraction patterns for each relation.

Open Information Extraction (Open IE) [2, 3, 15] is a domain independent information extraction paradigm and has been studied in both the natural language document corpus [15], and the Web environment [2, 3] to extract relation tuples. Open IE systems are initialized with a few manually provided domain independent extraction patterns. To produce training data for the algorithm, dependency parsing is conducted on a text corpus; domain independent extraction patterns are used to identify correct extractions. Using the created training data, a classifier is trained to identify the correct instances of target relations.

3. Method

Given a corpus of text, we split it into sentences using a sentence boundary detection tool[*1]. We then run a part-of-speech (POS) tagger[*2] and annotate each sentence with POS tags. To detect potential entities in sentences, we use a noun phrase chunking tool[*3] and extract noun phrase chunks containing at least one proper noun (NP). We consider all extracted noun phrases to be candidate entities between which a semantic relation might exist. Next, we replace the two entities respectively with two variables **X** and **Y** in a sentence. The entity that occurs first in the sentence is replaced by **X**, whereas the entity that occurs second is replaced by **Y**. Following the approach described in [4] we extract subsequence lexical and POS patterns that describe numerous semantic relations that exist between a given pair of entities.

We represent the dyadic relation between entity pairs and lexical-syntactic patterns as matrix A. Each extracted entity pair is represented as a row in this matrix, whereas each lexical-syntactic pattern is represented as a column. The A_{ij} element of the data matrix denotes the number of times the lexical-syntactic pattern p_j was extracted for the entity pair e_i. Each normalized row vector $\mathbf{e}_i$ in matrix A denotes the distribution of an entity pair e_i over lexical-syntactic patterns. Similarly, each normalized column vector $\mathbf{p_j}$ in matrix A denotes the distribution of a lexical-syntactic pattern p_j over entity pairs.

Algorithm 1 Sequential co-clustering algorithm.

Input: sets E, P, matrix A, thresholds θ, ϕ
Output: row clusters C_E, column clusters C_P

```
1: SORT(E)
2: SORT(P)
3: C_E ← {}, C_P ← {}
4: while E ≠ {} AND P ≠ {} do
5:    p ← POP(P)
6:    ASSIGN(p, C_p, θ)
7:    e ← POP(E)
8:    ASSIGN(e, C_E, φ)
9: end while
10: function ASSIGN(x, C, λ)
11: max ← −∞
12: c* ← null
13: for cluster c_j ∈ C do
14:    sim ← cosine(x, c_j)
15:    if sim > max then
16:       max ← sim
17:       c* ← c_j
18:    end if
19: end for
20: if max > λ then
21:    c* ← c* ⊕ x
22: else
23:    C ← C ∪ {x}
24: end if
```

We propose the co-clustering algorithm 1 to simultaneously cluster both lexical-syntactic patterns and entity pairs. The algorithm takes as its input E, P, A, and two clustering thresholds: row (entity pair) clustering threshold, ϕ, and column (lexical-syntactic pattern) clustering threshold, θ. The output of the clustering algorithm is the set of row clusters, C_E, and column clusters, C_P. First, in Line 1, we sort the set of entity pairs E in the descending order of total frequency, $\sum_j A_{ij}$, of each entity pair e_i with all lexical-syntactic patterns in P. Similarly, in Line 2, we sort the set of lexical-syntactic patterns P in the descending order of total frequency, $\sum_i A_{ij}$, of each pattern with all entity pairs in E. After sorting, the most common entity pairs and patterns in the corpus appear respectively at the beginning of E and P, whereas rare instances are shifted to the end. In Line 3, we initialize both row and column cluster sets to the empty set. The function, POP(P) in Line 5, returns the first pattern $p \in P$ and removes p from P, thereby reducing the size of P by one. Next, the function, ASSIGN, measures the similarity between the vector $\mathbf{p}$ that corresponds to pattern p and each column cluster $\mathbf{c}_j$ in C_P. Here, $\mathbf{c}_j$ denotes the centroid vector of the j-th column cluster. Similarity between $\mathbf{p}$ and $\mathbf{c}_j$ is measured using cosine similarity. If the similarity between p and the most similar cluster $\mathbf{c}^*$ is greater than the column clustering threshold θ, then we merge $\mathbf{p}$ to $\mathbf{c}^*$. Here, the operator $\oplus$ denotes vector addition. Otherwise, we form a new column cluster that contains $\mathbf{p}$ and append it to C_P. This procedure is re-

*1 http://stp.ling.uu.se/~gustav/java/classes/MXTERMINATOR.html
*2 http://nlp.stanford.edu/software/tagger.shtml
*3 http://chasen.org/~taku/software/yamcha/

peated for entity pairs in Lines 7 and 8. The *while-loop* in Algorithm 1 is repeated until both E and P are empty. It is noteworthy that the operation of merging rows or columns in Line 22 changes the distributions of patterns and entity pairs, thereby directly influencing the subsequent similarity computations. For example, if a pattern p is merged into a column cluster c_j, then, in the next iteration, when we compute cosine similarity between entity pairs, all patterns in cluster c_j will be considered as forming a single dimension. For further details of the sequential co-clustering algorithm and its threshold estimation refer [5].

4. Experiments

We use the pattern clusters produced using the proposed method to measure the relational similarity between entity pairs in the ENT benchmark dataset [4], and compare it to the previously proposed relational similarity measures. Relational similarity between two pairs of words is defined as the correspondence between semantic relations that exist between the two words in each word pair. For example, the two pairs, (*ostrich,bird*) and (*lion,cat*) are considered relationally similar because the relation ***X is a large Y*** holds between the two words **X** and **Y**, in each of those word pairs. Bollegala et al. [4] proposed a *supervised* method (RELSIM) to measure the relational similarity between two word pairs using a set of automatically extracted lexical pattern clusters. We employ the lexical-syntactic pattern clusters extracted using the proposed *unsupervised* method to measure the relational similarity between two word pairs. If the pattern clusters produced by the proposed method are useful to predict the relational similarity between given two word pairs, then it not only justifies the proposed method; it also demonstrates a useful application of it.

Following [4], we represent a word pair (a, b) as an n-dimensional vector $\mathbf{f}_{(a,b)}$, in which the k-th element, $\mathbf{f}^k_{(a,b)}$, is set to the total frequency of all lexical-syntactic patterns in a pattern cluster C_k with word pair (a, b). Under this representation, each pattern cluster C_k contributes a single feature to the feature vector for a word pair. Considering the fact that each pattern cluster is expected to represent a unique semantic relation, this feature representation can be regarded as a projection of a word pair over the space defined by semantic relations. We then measure the relational similarity (alternatively the relational distance) between two word pairs (a, b) and (c, d) using Mahalanobis distance between the corresponding feature vectors $\mathbf{f}_{(a,b)}$ and $\mathbf{f}_{(c,d)}$, which is given as

$$(\mathbf{f}_{(a,b)} - \mathbf{f}_{(c,d)})^{\mathrm{T}} \Gamma^{-1} (\mathbf{f}_{(a,b)} - \mathbf{f}_{(c,d)}). \quad (1)$$

Here, Γ^{-1} denotes the inverse of the inter-cluster correlation matrix computed for pattern clusters. The (i, j) element of Γ is computed as the inner product between the centroid vectors of pattern clusters i and j. In contrast to the Euclidean distance, the Mahalanobis distance has shown to be more appropriate for the task of measuring relational similarity [4] because semantic relations are not independent.

We use the ENT dataset [4] as a gold standard of relational similarity. The ENT dataset contains 100 entity pairs describing the five semantic relations: **ACQUISITION** (between two companies, where one company is acquired by the other, e.g. (*Google,YouTube*)), **HEADQUARTERS** (between a company and the location of its headquarters, e.g. (*Microsoft,Redmond*)), **FIELD** (between a person and his field of expertise, e.g. (*Albert Einstein,Physics*)), **CEO** (between a company and its current CEO, e.g. (*Steve Jobs,Apple*)), and **BIRTHPLACE** (between a person and his place of birth, e.g. (*Charlie Chaplin,London*)). For each entity pair (a, b) of relation R in the ENT dataset, we measure the relational similarity between (a, b) and the remaining 99 entity pairs. A good relational similarity measure must assign higher similarity scores to entity pairs with similar semantic relations. Consequently, we evaluate the top k similar pairs to each entity pair in the dataset using average precision given as

$$\text{Average Precision} = \frac{\sum_{r=1}^{k} \text{Pre}(r) \times \text{Rel}(r)}{\text{no. of relevant entity pairs}}. \quad (2)$$

Here, $\text{Rel}(r)$ is a binary valued function that returns 1 if the entity pair at rank r has the same relation (i.e. R) as in (a, b). Furthermore, $\text{Pre}(r)$ is the precision at rank r, which is given as

$$\text{Pre}(r) = \frac{\text{no. of entity pairs with relation } R \text{ in top } r \text{ pairs}}{r}. \quad (3)$$

In fact, the ENT dataset contains 20 entity pairs for each relation. Following previous work, we evaluate using the top 10 ranked results (i.e. $k = 10$).

The pattern extraction algorithm extracts 142,655 lexical patterns from the text snippets provided in the ENT dataset. We then use Algorithm 1 to co-cluster both entity pairs and the extracted patterns. Clustering thresholds θ and ϕ are estimated respectively as 0.67 and 0.83 using the method described in [5]. This process produces 9 row (entity pair) clusters and 139 column (pattern) clusters. Table 1 presents a comparison of the relational similarity measured using the pattern clusters produced using the proposed method (**PROP**) against four others: **VSM** (Vector Space Model-based approach [16]), **LRA** (Latent Relational Analysis [16]), **EUC** (Euclidean distance between feature vectors [4]), and **RELSIM** (Mahalanobis distance between feature vectors [4]). Except for the proposed method, all other figures in Table 1 are obtained from previously published results obtained using the ENT dataset. Overall, PROP shows the highest average precision score (0.76) in Table 1. Moreover, for three out of the five relations in the ENT dataset, PROP outperforms all existing relational similarity measures. It is noteworthy that although RELSIM has a similar average precision score (0.74) to that of PROP, unlike PROP, which is unsupervised, RELSIM is a supervised method that requires labeled data for training. Moreover, both EUC and RELSIM use one-sided sequential clustering in which only patterns are clustered. In contrast, PROP clusters both patterns and entity pairs simultaneously using Algorithm 1, which exploits the dyadic structure in the data more effectively.

表 1: Performance of the proposed method and previous work on relational similarity measures.

Relation	VSM	LRA	EUC	RELSIM	PROP
ACQUISITION	0.92	0.92	0.91	**0.94**	0.89
HEADQUARTERS	0.84	0.82	0.79	0.86	**0.97**
FIELD	0.44	0.43	0.51	**0.57**	0.42
CEO	0.95	0.96	0.90	0.95	**0.99**
BIRTHPLACE	0.27	0.27	0.33	0.36	**0.53**
Overall Average Precision	0.68	0.68	0.69	0.74	**0.76**

5. Conclusion

We proposed a sequential co-clustering algorithm for the task of extracting semantic relations that exist between named entities using two complementary definitions of a relation: intensional and extensional. Relations between entities were represented using lexical and syntactic patterns extracted from the contexts in which those entities co-occur. The proposed co-clustering algorithm was evaluated for its ability to predict the relational similarity between entities in a previously proposed benchmark dataset. Our experimental results show that despite the proposed method being an unsupervised method, its performance is comparable to that of previously supervised approaches for measuring relational similarity. (An extended version of this paper appears in the International World Wide Web Conference 2010 [5].)

参考文献

[1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *ICDL'00*, 2000.

[2] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI'07*, pages 2670–2676, 2007.

[3] M. Banko and O. Etzioni. The tradeoffs between traditional and open relation extraction. In *ACL'08*, pages 28–36, 2008.

[4] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Measuring the similarity between implicit semantic relations from the web. In *WWW'09*, pages 651–660, 2009.

[5] Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Relational duality: Unsupervised extraction of semantic relations between entities on the web. In *WWW'10*, 2010.

[6] S. Brin. Extracting patterns and relations from the world wide web. In *WebDB Workshop at EDBT'98*, pages 172 – 183, 1998.

[7] R. Bunescu and R. Mooney. Subsequence kernels for relation extraction. In *NIPS'06*, pages 171–178, 2006.

[8] I.M. Copi. *Introduction to Logic*. Prentice Hall College Div, 1998.

[9] A. Culotta, A. McCallum, and J. Betz. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *HLT/NAACL'06*, pages 296–303, 2006.

[10] O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderl, D. S. Weld, and E. Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165:91–134, 2005.

[11] C. Giuliano, A. Lavelli, and L. Romano. Exploiting shallow linguistic information for relation extraction from biomedical literature. In *EACL'06*, pages 401–408, 2006.

[12] A. Harabagiu, C. A. Bejan, and P. Mor?arescu. Shallow semantics for relation extraction. In *IJCAI'05*, pages 1061–1066, 2005.

[13] M. Pasca, D. Lin, J. Bigham, A. Lifchits, and A. Jain. Organizing and searching the world wide web of facts - step one: the one-million fact extraction challenge. In *AAAI'06*, pages 1400–1405, 2006.

[14] D. Roth and W. Yih. A linear programming formulation for global inference in natural language tasks. In *CoNLL'04*, pages 1–8, 2004.

[15] Y. Shinyama and S. Sekine. Preemptive information extraction using unrestricted relation discovery. In *HLT-NAACL'06*, pages 304–311, 2006.

[16] P.D. Turney. Measuring semantic similarity by latent relational analysis. In *IJCAI'05*, pages 1136–1141, 2005.

[17] D. Zelenko, C. Aone, and A. Richardella. Kernel methods for relation extraction. *JMLR*, 3:1083–1106, 2003.

[18] G. Zhou, M. Zhang, D. H. Ji, and Q. Zhu. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *EMNLP-CoNLL*, pages 728 – 736, 2005.

[19] Jun Zhu, Zaiqing Nie, Xiaojiang Liu, Bo Zhang, and Ji R. Wen. Statsnowball: a statistical approach to extracting entity relationships. In *WWW'09*, pages 101–110. ACM, 2009.