1A4-04

Dually Extract Semantic from the Web

Haibo Li[*1] Yutaka Matsuo[*2] Mitsuru Ishizuka[*1]

[*1] Department of Creative Informatics
[*2] Department of Technology Management for Innovation
University of Tokyo

Traditional relation extraction requires pre-defined relations and many human annotated training data. Meanwhile, open relation extraction demands a set of heuristic rules to extract all potential relations from text. These requirements reduce the practicability and robustness of information extraction system. In this paper, we propose a *Relation Expansion* framework, which uses a few seed sentences marked up with two entities to expand a set of sentences containing target relations. During the expansion process, label propagation algorithm is used to select the most confident entity pairs and context patterns. We test the proposed framework with four relationships, the results show that the label propagation is quite competitive comparing with existing methods.

1. Introduction

There are two kinds of methods to extract relations from documents: traditional relation extraction and open relation extraction. For the traditional relation extraction system [Chen et al. 06, Aron and Jeffrey 04], the user is usually required to provide a large amount of annotated texts to identify the interesting relation. On the other hand, the open information extraction system [Banko et al. 07] uses some generalized patterns or a small set of relation-independent heuristics to extract all potential of relations between name entities.

In this paper, we propose a general framework— *Relation EXpansion* (REX) which uses given seed sentences to bootstrap relevant sentences from the Web. The target relations are "weakly" defined by marking the relation containing entity pair in the given seeds. The returned sentences are also marked with entity pairs containing the target relation. For example, given a sentence : (*Albert Einstein*) *was born in* (*Ulm*), REX returns some relevant sentences, such as: (*Bethlehem*)*, the birthplace of* (*Jesus*); (*Pable Picasso*) *was born in* (*Malaga*) and so on. The proposed framework uses dual expansion model to incrementally discover relevant sentences.

The proposed REX framework regards the pattern or entity pair filtering problem as a semi-supervised learning problem. Since various entity pair and context patterns can be extracted from the Web, we need to find the most similar entity pairs and context patterns for the expansion process. Previous bootstrapping based researches treat entity pair and pattern filtering as a binary classification problem or use a confidence measure to select the instances[Agichtein and Gravano 00, Etzioni et al. 05].

2. Related Work

Bootstrapping strategy based relation extraction can efficiently leverage a large amount of data on the Web. The method is initialized with a seed set and extract relative facts or relations. For example, the Snowball [Agichtein and Gravano 00] extracts entity pairs containing predefined relationship from corpus. The SatSnowball [Zhu et al. 09] extends the Snowball with statistical method and extracts the entity pairs and keywords around the entities. Furthermore, both DIPRE [Brin 98] and SatSnowball use a general form to represent extracted patterns. Although these general form patterns improve the coverage of extracted pattern, they decrease the precision.

Many previous reports have described that properly using unlabeled data to complement a traditional labeled data set can improve the performance of supervised algorithm. For example, a named entity classification algorithm proposed in[Collins and Singer 99], which is based on co-training framework, can reduce the need for supervision to a handful of seed rules; Label propagation [Zhu et al 03] is a graph based semi-supervised learning models in which the entire data set as a weighted graph and the label score is propagated on this graph.

Contact: Haibo Li, University of Tokyo, 7-3-1 Hongo Bunkyo-ku Tokyo 113-8656, lihaibo@mi.ci.i.u-tokyo.ac.jp

3. Framework of Relation Expansion

3.1 Relation Duality and Expression Variety

The relation expansion becomes practicable since the relation duality and expression variety is easily available on the Web. First, a semantic relation between two entities can be represented from two different "aspects": the entity pair itself and the context around it. These two kinds of information are usually called two views in machine learning community. For example, the *Person◇Birthplace* relation can be expressed with a set of entities pairs view, such as (*Albert Einstein, Ulm*),(*Jesus, Bethlehem*) and so on. From lexical patterns view, this relation can also be represented with the contexts:"*A was born in B*", "*B, the birth place of A*" and so on. This relation duality enables us to co-bootstrap the context patterns and word pairs. Second, the co-occurrence relation between a entity pair and a context pattern is many-to-many relation. For example, for the *Person◇ Birthplace* relation, the entity pair (*Albert Einstein, Ulm*) may appear with many different patterns: "*A was born in B*", "*B, the birth place of A*" and so on. Similarly, a context pattern may also be used together with many different *Person◇ Birthplace* entity pairs. This expression variety enables more and more word pairs or context patterns can be extracted.

3.2 Relation Expansion

In this section, we give an overview of Relation EXpansion framework. The two main components are explained in incoming sections. The Figure 1 shows the architecture of REX framework.

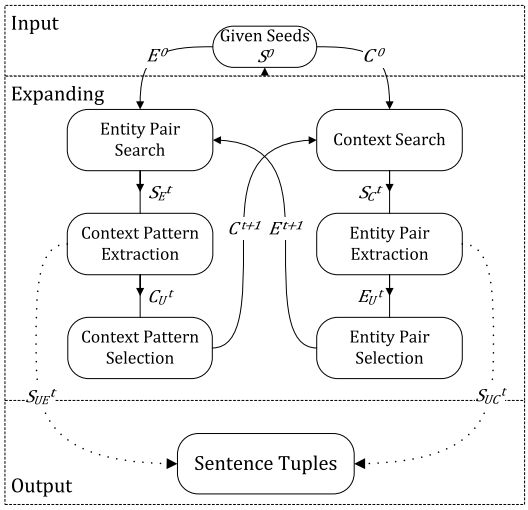


Figure 1: The framework of relation expansion, with four components —Input, Expanding and Output.

In the framework, the sentences containing target relation are represented as a tuple: (e, c) where $e = (e_a, e_b)$ is entity pair and c is context pattern. The *input* of REX is a small relation tuple set $S^0 = \{(e_i, c_j)|i = 1, 2, \cdots, n; j = 1, 2, \cdots, m; \}$; The *output* of REX is a ranked list of relation tuples. REX distends S^0 to construct a potential target sentence tuple set.

The *Expanding* part uses a dually extraction model. Let's note $E^0 = \{e_i|(e_i, c.) \in S^0\}$ and $C^0 = \{c_j|(e., c_j) \in S^0\}$. E^0 and C^0 is the entity pairs and context patterns in S^0 respectively. In *t*-th expansion iteration, we submit some queries generated from entity pairs E^t and context patterns C^t (at the beginning $t = 0$) to a Web search engine respectively. Specifically, the context patterns C^t are used in the Context Search part and the entity pairs E^t are used in the Entity Pair Search part. We crawl the top 100 web pages returned by the Web search engine. The texts in these web pages are split into sentences. In the Entity Pair Extraction step, a Named Entity Recognization tool*1 is used to label the named entities in each sentence. All the entity pair candidates are extracted and added to the candidate set E_U^t. The REX selects some entity pairs $E^{t+1} \subseteq E_U^t$ for *t*+1 round of expansion. At the same time, the corresponding sentences containing candidate entity pairs are added to the sentence tuple candidate set S_{UC}^t. Similarly, the context pattern C_U^t are extracted and some context patterns $C^{t+1} \subseteq C_U^t$ are selected for *t*+1 round of entity pair expansion. The corresponding sentence tuples are also added to S_{UE}^t.

4. Entity Pair and Context Pattern Filtering

It is also because of the many-to many relation between entity pair and context pattern, extracted entity pairs and context patterns are not all applicable to the next round of expanding. Therefore, for an entity pair, some context pattern that represent different types of relations may be extracted. For example, using the entity pair (*Albert Einstein, Ulm*), we can extract two types of context patterns:"*A was born in B*" and "*A's stay in B*". The two context patterns have totally different semantic relation. Therefore, the context pattern and entity pair filtering is necessary. In this section, we take the entity pair filtering for instance to illustrate our method.

∗1 http://nlp.stanford.edu/software/CRF-NER.shtml

4.1 Graph Based Filtering Method

We use the label propagation algorithm to filter out the irrelevant entity pairs. In the algorithm, E^t is labeled data and E_U^t is unlabeled data. The algorithm models entire data set $E_U^t \cup E^t$ as a weighted graph and propagates label scores through the graph along its high-density areas. Each node in the graph receives a relevance score after propagation. According to this score, h nodes with the highest score are selected as E^{t+1} .

Let $E = \{e_1, e_2, \cdots, e_l, \cdots, e_{n+l}\}$ denote the set of entity pairs to be filtered. Similarity to [Zhu et al 03], we construct a full connected undirected graph $G^E < E, L >$. The nodes $E = E^t \cup E_U^t$ correspond to the $n+l$ entity pairs, and L is the edge set. This graph is represented as an $(n+l) \times (n+l)$ similarity matrix T, in which T_{ij} corresponds to the similarity of e_i and e_j. Let Y denote a $n+l$ column vector in which the first l elements $Y_i (i \leq l)$ correspond to the entity pairs in E^t and the remaining points $Y_u (l+1 \leq u \leq n+l)$ are candidates in E_U^t. Let's denote D is a diagonal matrix:$D_{ii} = \sum_{j=1}^{n+l} T_{ij}$. For the convergence of label propagation algorithm the matrix T is normalized symmetrically as: $W = D^{-\frac{1}{2}} T D^{-\frac{1}{2}}$.

Formally, the label propagation can be formulated as a cost function $Q(Y, W)$ in a joint regularization framework,

$$Q(Y, W) = \frac{1}{2} \sum_{i,j=1}^{n+l} W_{ij} \left\| \frac{Y_i}{\sqrt{D_{ii}}} - \frac{Y_j}{\sqrt{D_{jj}}} \right\|^2 + \mu \sum_{i=1}^{n+l} \left\| Y_i - Y_i^0 \right\|^2$$

where $\mu > 0$ controls the trade-off between the first term and the second term. Y_i^0 is the initial relevance score of the entity pair e_i with respect to the given seeds. Y_i is the propagated relevance score.

The final relevance score vector is:

$$Y^* = arg \min_{Y^T \in R^{n+l}} Q(Y, W).$$

After differentiating and simplifying, a closed-form solution can be derived as:

$$Y^* = \alpha (I - \beta W)^{-1} Y^0, \alpha = \frac{1}{1+\mu}, \beta = \frac{\mu}{1+\mu}$$

where I is a identity matrix. Given the initial ranking scores Y^0 and the matrix W, the final relevance score Y^* can be computed directly with the equation 4.1. In this paper, we eliminate the parameter μ. The entity pairs are sorted according to their relevance in decreasing order and top l entity pairs in E_U^t are selected as E^{t+1} for next iteration.

4.2 Similarity Graph Generation

In order to build the similarity matrix T, the co-occurrence matrix M is used. First, we built two sets: $E = E_U^t \cup E^t$ and $C = C_U^t \cup C^t$, where $|E| = n + l$, $|C| = m + l$. Then REX constructs the occurrence matrix $M_{ij} = (M_{ij}, \quad i = 1, 2, \ldots, n+l; j = 1, 2, \ldots, m+l;)$ using the Web search engine. For the entity pair $e_i = \{e_{ia}, e_{ib}\} \in E$ and context pattern $c_j \in C$, we generate the query q_{ij}: "e_{ia} c_j e_{ib}" or "e_{ib} c_j e_{ia}". The page count M_{ij} of query q_{ij} is approximately treated as the co-occurrence frequency of e_i and c_j.

Table 1: Performance of Label Propagation and Baselines on Context Pattern and Word Pair Filtering Task (P@20).

Relation Type		VSM	Snowball	KnowItAll	LRA	REX-Dice	REX-Cos	REX-Jaccard
C*◇*O	E	0.75	0.80	0.80	0.80	0.85	0.85	0.80
	C	0.80	0.80	0.80	0.80	0.85	0.80	0.80
A*◇*A	E	0.90	0.80	0.80	0.90	0.85	0.90	0.85
	C	0.80	0.90	0.80	0.85	0.80	0.85	0.85
P*◇*B	E	0.85	0.80	0.85	0.90	0.85	0.80	0.90
	C	0.85	0.85	0.80	0.85	0.85	0.80	0.85
O*◇*H	E	0.75	0.80	0.75	0.85	0.85	0.80	0.85
	C	0.70	0.65	0.70	0.75	0.75	0.70	0.70
Average	E	0.81	0.80	0.80	**0.86**	0.85	0.84	0.85
	C	0.79	0.80	0.78	0.82	**0.83**	0.79	0.80

For the co-occurrence matrix M, the row of $M_{i\cdot}$ can be treat as a context pattern expression of entity pair e_i. Correspondingly, different measures can be used to measure the similarity between the vector $M_{i\cdot}$ and $M_{j\cdot}$. Then the edge of G^E is weighted by the heat kernel as follow:

$$T_{ij} = \exp\left(-\frac{sim(M_{i\cdot}, M_{j\cdot})}{2\sigma^2}\right) \quad (1)$$

where σ is a parameter for the heat kernel and $sim(M_{i\cdot}, M_{j\cdot})$ is the similarity of e_i and e_j .

Similarly, the column vector of $M_{\cdot j}$ can be regarded as a feature vector of context pattern c_j. Then the context pattern similarity graph can be constructed and the context patterns are filtered in the same method as entity pairs .

5. Experiment

In this section, we report empirical results of REX with different configurations. From previous sections, we can see the key tasks of the REX framework is a ranking task. This task is to rank entity pairs and context patterns. According to the ranking score, the most similar instances can be selected. For the evaluation of the result, we adopt a frequently used measures for ranking quality: *P@n* .

The named entity recognization tool used in this experiment can label four types of entities: Organization, Person, Location and Miscellaneous names. Therefore, we test our method on the following four relation types:*CEO◇Organization* (*C◇O*), Acquirer◇Acquiree (*A◇A*), Person◇Birthplace (*P◇B*) and Company◇Headquarters (*C◇H*). These four relation types cover the first three types of named entities. For each relation type, we give a seed for bootstrapping. The seeds used for expansion are listed as follow:

***C◇O*:** (*Bill Gates*) *is the CEO of* (*Microsoft*).

***A◇A*:** (*Google*) *has acquired* (*YouTube*).

***P◇B*:** (*Albert Einstein*) *was born in* (*Ulm*).

***C◇H*:** (*Microsoft*) *headquarters in* (*Redmond*).

We run the proposed framework described in previous section on the Web, and the YahooBOSS API [*2] is used to search with the given query.

In this experiment, we compare the label propagation algorithm with following methods:

VSM: This method is a vector based method which is proposed by Turney et al.[Turney 06]. Since the co-occurrence matrix of entity pair and context pattern is built as mentioned in previous section, entity pair and context pattern can be used as the vector representing each other. The similarity between entity pairs or context patterns can be computed as the cosine of the two corresponding vectors. Then the entity pairs and context patterns which have the highest similarity score are selected.

Snowball: This is the measure proposed by Agichtein et.al[Agichtein and Gravano 00]. The patterns are measured by the *confidence*, by which the pattern that tend to generate wrong data are filtered. In this experiment, we use the pattern confidence to measure the quantity of context pattern and entity pair.

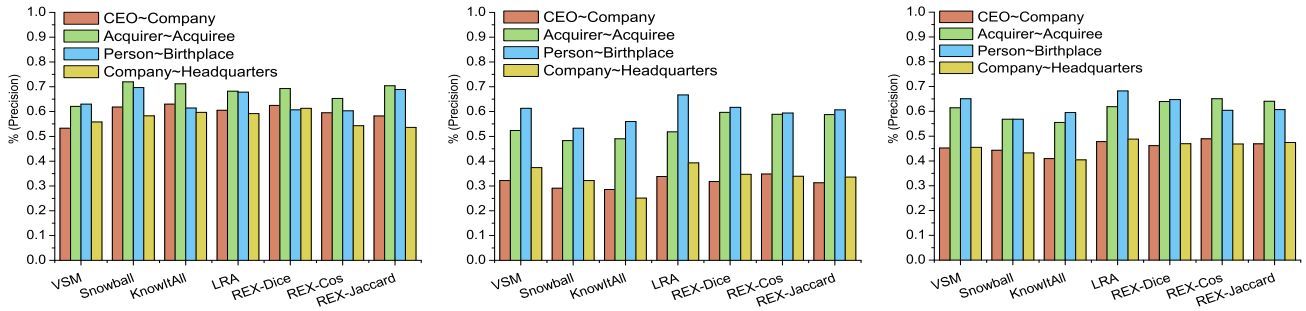
LRA: The Latent Relational Analysis(LRA) is proposed by Turney [Turney 06]. For a matrix M, supposing the rows represent the entity pairs and the columns represent context patterns. Then Singular value decomposition(SVD) is performed on the matrix, in which the matrix toolkit [*3] is used. The relation similarity of entity pair can be measured by the cosine of the angle between the two vector in matrix $U_k\Sigma_k$. Similarly, the relevance of context pattern can be measured using the vector in matrix $\Sigma_k V_k^T$. In our experiment, k is set as 10. LRA is the current state-of-the-art relation similarity measure.

KnowItAll: The *KnowItAll* information extraction system[Etzioni et al. 05] uses the following way to measure the relation between context pattern and entity pair:

$$KnowItAll(e_i, c_j) = \frac{|e_{ia}, c_j, e_{ib}|}{|e_{ia}, *, e_{ib}|}$$

∗2 http://developer.yahoo.com/search/boss/

∗3 http://code.google.com/p/matrix-toolkits-java/



(a) Precision of Entity Pairs (b) Precision of Context Patterns (c) Average of Word Pair and Context Pattern

Figure 2: Average precision of instances extraction in the second iteration using different filtering methods.

In order to test the sensitivity of label propagation algorithm to the similarity measure, we test three frequently used spirality measure in the naturel language processing community: 1) Dice coefficient (*REX-Dice*); 2) Cosine similarity (*REX-Cos*); 3) Jaccard measure (*REX-Jaccard*)

These seven methods described above presented for comparison in table 1. The letter 'E' denotes the entity pair filtering and 'C' denotes the context pattern filtering. The results show that label propagation algorithm is very competitive to instance filtering task. For entity pair filtering, we can see the *LRA* get the highest performance. Furthermore, the label propagation based algorithms are also competitive. For context pattern filtering, the *REX-Dice* gets the best performance of P@20 score respectively. Moreover, we also notice the performance of entity pair selection is better than context pattern selection. A close look into the sentences extracted from the Web reveal that an entity pair often contain more than one type of relations. Then when the entity pairs are used to extracted context pattern, many noise also are extracted. On the other hand, a context pattern usually only expresses a "specified" relation and extracted entity pairs are more "centrical". Therefore, the context pattern selection task is more difficult than entity pair selection. For example, given entity pair (*Bill Gates, Microsoft*), we extract two context: "*is the CEO of*" and "*has retired from*". These two context patterns are expressing different relations.

In order to test the efficiency of the REX framework, the precision of the second round of expansion is plotted. We randomly selected 10% of extracted context patterns and entity pairs to manually evaluate. The figure 2(a) and 2(b) shows the precision of expanding result using different selected instances. We can observe that the precision entity pair is higher than precision context pattern. The reason is mentioned previously. The figure 2(c) is the average precision of context pattern extraction and word pair extraction.

6. Conclusions

We proposed a general framework to extract sentences containing certain relationship between an entity pair. We utilized the duality and expression diversity of semantic relation to bootstrap from given seed set. For each expansion iteration, we apply the label propagation algorithm to select the most confident entity pairs and context patterns. Experimental results show that label propagation algorithm works efficiently.

References

[Agichtein and Gravano 00] Agichtein, E., and Gravano, L. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the Fifth ACM International Conference on Digital Libraries* (2000).

[Aron and Jeffrey 04] Aron, C., and Jeffrey, S. Dependency tree kernels for relation extraction. In *ACL04*, 423–429 (2004).

[Banko et al. 07] Banko, M.; Cafarella, M. J.; Soderl, S.; Broadhead, M.; and Etzioni, O. Open information extraction from the web. In *IJCAI-07*, 2670–2676 (2007).

[Brin 98] Brin, S. Extracting patterns and relations from the world wide web. In *WebDB Workshop at EDBT98*, 172–183 (1998).

[Chen et al. 06] Chen, J.; Ji, D.; Tan, C.; and Niu, Z. Relation extraction using label propagation based semi-supervised learning. In *ACL06*, 129–136 (2006).

[Collins and Singer 99] Collins, M., and Singer, Y. Unsupervised models for named entity classification. In *Proc. Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 100–110 (1999).

[Etzioni et al. 05] Etzioni, O.; Cafarella, M.; Downey, D.; Popescu, A.-M.; Shaked, T.; Soderland, S.;Weld, D. S.; and Yates, A. Unsupervised named-entity extraction from the web: an experimental study. In *Artificial Intelligence* 165:91–134 2005.

[Turney 06] Turney, P. D. Similarity of semantic relations. *Computational Linguistics* 32:379–416 (2006).

[Zhu et al. 09] Zhu, J.; Nie, Z.; Liu, X.; Zhang, B.; ; and Wen, J.-R. Statsnowball: a statistical approach to extracting entity relationships. In *WWW'09* 101–110 (2009).

[Zhu et al 03] Zhu, X.; Ghahramani, Z.; and Lafferty, J. Semisupervised learning using gaussian fields and harmonic functions. In *Proc. 20th International Conference on Machine Learning*, 912–919 (2003).