2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology

Relations Expansion: Extracting Relationship Instances from the Web

Haibo Li
Dept. of Creative Informatics
University of Tokyo
Tokyo, Japan
Email:lihaibo@mi.ci.i.u-tokyo.ac.jp

Yutaka Matsuo
Dept. of Technology Management for Innovation
University of Tokyo
Tokyo, Japan
Email:matsuo@biz-model.t.u-tokyo.ac.jp

Mitsuru Ishizuka
Dept. of Creative Informatics
University of Tokyo
Tokyo, Japan
Email: ishizuka@i.u-tokyo.ac.jp

***Abstract*—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. The label propagation algorithm is a graph based semi-supervised learning method which models the entire data set as a weighted graph and the label score is propagated on this graph. We test the proposed framework with four relationships, the results show that the label propagation is quite competitive comparing with existing methods.**

***Keywords*-relation extraction; semi-supervised learning;**

I. Introduction

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.

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[1], [2]. The proposed REX framework regards the pattern or entity pair filtering problem as a semi-supervised ranking problem. As described herein, we use label propagation algorithm, a graph based semi-supervised algorithm, to filter out the irrelevant instances.

II. 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 [1] extracts entity pairs containing predefined relationship from corpus. The SatSnowball [3] extends the Snowball with statistical method and extracts the entity pairs and keywords around the entities. Furthermore, both DIPRE [4] and SatSnowball use a general form to represent extracted patterns.

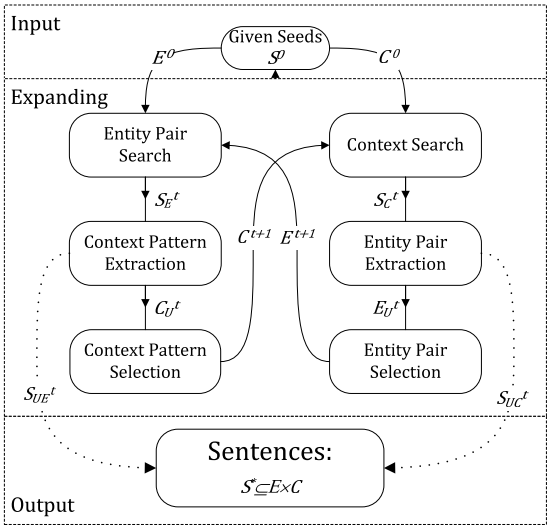

Figure 1. The framework of relation expansion, with three components —Input, Expanding and Output.

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[5], which is based on co-training framework, can reduce the need for supervision to a handful of seed rules; Label propagation [6] 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.

III. Framework of Relation Expansion

In this section, we give an overview of Relation EXpansion framework. The Figure 1 shows the architecture of REX framework. 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*

978-0-7695-4191-4/10 $26.00 © 2010 IEEE
DOI 10.1109/WI-IAT.2010.269

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 list of relation tuples. REX distends S^0 to construct a potential target sentence tuple set S^*.

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 Recognition 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^t_U. The REX selects some entity pairs $E^{t+1} \subseteq E^t_U$ 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^t_{UC}. Similarly, the context pattern C^t_U are extracted and some context patterns $C^{t+1} \subseteq C^t_U$ are selected for *t*+1 round of entity pair expansion. The corresponding sentence tuples are also added to S^t_{UE}.

IV. Dual Expansion

In the entity pair search step, REX uses an entity pair $e_i = (e_{ia}, e_{ib}) \in E^t$ to generate 10 queries, such as "e_{ia} * e_{ib}", "e_{ib} * e_{ia}" and so on. The wildcard is add between two entities. We add 5 wildcards operator "*" in the query at most. For the context search step, given a context pattern $c_j \in C^t$, only one type of query is constructed:"* c_j *".

We submit these queries to the Web search engine. The search engine returns web pages in which the query entities co-occur with a maximum contexts of five tokens. The quotation marks around a query to ensure that the two entities appear in the specified order (e.g. e_{ia} before e_{ib} in text retrieved by the query "e_{ia} * e_{ib}"). All the crawled web pages are segmented into sentences. The sentences set S^t_E and S^t_C is constructed respectively.

From sentences set S^t_E, two types of patterns are extracted: *A*∼*B* pattern and *B*∼*A* pattern. For example, given entity pair (*Steve Jobs, Apple*) and the sentence "*Steve Jobs is CEO of Apple.*", we can extract the pattern "*A is CEO of B*" pattern. For the sentence "*Apple's CEO, Steve Jobs* ⋯", we can extract the pattern "*B's CEO, A*" pattern. The two sentences express the same relation between *Steve Jobs* and *Apple*.

V. Entity Pair and Context Pattern Filtering

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.

A. Graph Based Filtering Method

We use the label propagation algorithm[6] to filter out the irrelevant entity pairs. In the algorithm, E^t is labeled data and E^t_U is unlabeled data. The algorithm models entire data set $E^t_U \cup E^t$ 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 [7], we construct a full connected undirected graph $G^E < E, L >$. The nodes $E = E^t \cup E^t_U$ 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^t_U. 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^0_i 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:

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

In this paper, we simply set the parameter $\mu = 1$. The entity pairs are sorted according to their relevance in decreasing order and top l entity pairs in E^t_U are selected as E^{t+1} for next iteration.

B. Similarity Graph Generation

In order to build the similarity matrix T, the co-occurrence matrix M is used. First, we built two sets: $E = E^t_U \cup E^t$ and $C = C^t_U \cup C^t$, where $|E| = n + l$, $|C| = m + l$. Then

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

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.

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:

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

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 .

VI. Experiment

Although the proposed framework can be used to extract any relation type between two named entities, the named entity recognition tool is a bootle neck of our framework. Since the recognizer can only 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 evaluate the label propagation algorithm in the entity pair and context pattern filtering task. Let's take entity pair selection as example. Given a set of entity pairs, the goal is to select 20 entity pairs which are most similar to the given seeds from all the candidates.

For the evaluation of the result, we adopt frequently used measures for ranking quality: P@n. The precision at rank n with respect to the given seed e_l is defined as follow:

$$P@n = \frac{\sum_{i=1}^{n} Rel(e_i, e_l)}{n}$$

where $Rel(e_i, e_l)$ is the relevance between e_i and e_l. $Rel(e_i, e_l)$ is binary, which is set to 1 when e_i is relevant to the seed, otherwise is set to 0. In our experiment, we report the precision of P@20. The label propagation algorithm is compared with following methods.

VSM: This method is a vector based method which is proposed by Turney et al.[8]. 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 representation each other. The similarity between entity pairs or context patterns can be computed as the cosine of the two corresponding vectors and vice versa. Then the entity pairs and context patterns which have the highest similarity score are selected.

CON: This is the measure proposed by Agichtein et.al[1]. The patterns are measured by the *confidence*, by which the context patterns that tend to generate wrong entity pairs are filtered. In this experiment, we use the instance confidence to measure the quantity of context pattern and entity pair.

$$Conf(s) = \frac{s_{positive}}{s_{positive} + s_{negative}}$$

in which $s_{positive}$ is the number of positive matches of entity pair or context pattern. Taking entity pair as example, if entity pair e matches the pattern c which can be found in previous iteration, then this match is considered as a *positive* match. Otherwise, the match is *negative*.

LRA: The Latent Relational Analysis(LRA) is proposed by Turney [8]. 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.

KIA: The *KnowItAll* information extraction system[2] uses the following method to measure the relation between context pattern and entity pair:

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

In order to test the sensitivity of label propagation to the similarity measure, we test three frequently used similarity measures in the naturel language processing community. We use these measures to weight the edges of graphs for label propagation.

Dice: Dice coefficient is a usually used measure in Natural Language Processing community. In this experiment we want to test the sensitivity of label propagation algorithm

[2]http://developer.yahoo.com/search/boss/

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

Table I
PERFORMANCE OF LABEL PROPAGATION AND BASELINES ON CONTEXT PATTERN AND WORD PAIR FILTERING TASK (P@20).

Relation	Type	VSM	CON	KIA	LRA	REX-Dice	REX-Cos	REX-Jac
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

to the similarity measures.The Dice coefficient is used to weight the graph G^E and G^Cwhich is defined as:

$$Sim_{Dic}(M_{i\cdot}, M_{j\cdot}) = 2\frac{|M_{i\cdot} \bigcap M_{j\cdot}|}{|M_{i\cdot}| + |M_{j\cdot}|}$$

Cos: In this method, the cosine similarity is used to weight the graph G^E and G^C. Given co-occurrence matrix of context pattern and entity pair M, we can construct the graph G^E with the following cosine similarity measure:

$$Sim_{Cos}(M_{i\cdot}, M_{j\cdot}) = Cosine(M_{i\cdot}, M_{j\cdot})$$

Similarly, graph G^C can be constructed with cosine similarity measure.

Jac: In this setting, we compute the *Jaccard* score between each entity pairs e_i and e_j, using following equation:

$$Sim_{Jac}(M_{i\cdot}, M_{j\cdot}) = \frac{|M_{i\cdot} \cap M_{j\cdot}|}{|M_{i\cdot} \cup M_{j\cdot}|}$$

These seven methods described above presented for comparison in table I. 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.

VII. 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

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

[2] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates, "Web-scale information extraction in knowitall:(preliminary results)." ACM, 2004, pp. 100–110.

[3] J. Zhu, Z. Nie, X. Liu, B. Zhang, , and J.-R. Wen, "Statsnowball: a statistical approach to extracting entity relationships," 2009, pp. 101–110.

[4] S. Brin, "Extracting patterns and relations from the world wide web," in *WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT98*, 1998, pp. 172–183.

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

[6] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems 16*. MIT Press, 2004, pp. 321–328.

[7] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proc. 20th International Conference on Machine Learning*, 2003, pp. 912–919.

[8] P. D. Turney, "Similarity of semantic relations," *Computational Linguistics*, vol. 32, pp. 379–416, 2006.