Duality based Expansion for Relation of Entity Pair

Haibo Li Yutaka Matsuo Mitsuru Ishizuka
The University of Tokyo
lihaibo@mi.ci.i.u-tokyo.ac.jp, matsuo@biz-model.t.u-tokyo.ac.jp, ishizuka@i.u-tokyo.ac.jp

Abstract

Traditional relation extraction requires pre-defined relations and many human annotated training data. 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 ranked list of sentences containing target relation between entity pair. The proposed framework uses dual expansion model to incrementally discover relevant sentences. Then these extracted relation instances are ranked according their relevance to the given seeds. The proposed framework is tested with four frequently used relationships; the results show the effectiveness of relation expansion framework.

Introduction

There are two kinds of method to extract relations from documents: traditional relation extraction and open relation extraction. For the traditional relation extraction system [1, 6], the user is usually required to provide a large amount of annotated texts to identify the target relation. On the other hand, the open information extraction system [3, 4] uses some generalized patterns or a small set of relation-independent heuristics to extract all potential relations between name entities.

In this paper, we propose a general framework: *Relation EXpansion* (REX), which is different from existing work in two aspects. First, the proposed REX framework uses a small set of sentences with marked entities as seeds to "weakly define" the target relation type. These sentences are used as seeds for the REX framework to extract more relevant sentences from the Web. Second, the REX framework returns a ranked tuple list of extracted sentences. In this way, the user can easily handle the returned results.

The relation expansion becomes practicable because of the relation duality and expression diversity of texts on the Web. First, a relationship between entities can be represented with two different "kinds" of information: the entity pair itself and the context around it. This relation duality enables us to co-bootstrap the context patterns and word pairs. Second, an entity pair containing a relation often co-occurs with various context patterns. Similarly, a context pattern may also be used together with many different entity pairs. This expression diversity enables us to extract more and more word pairs or context patterns.

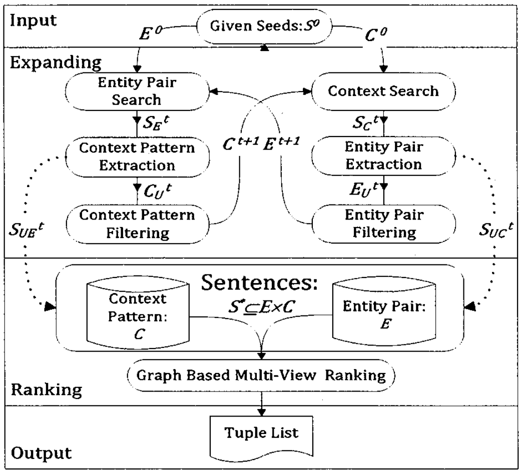

Figure 1. The Framework of Relation Expansion

Related Work

Bootstrapping strategy based relation extraction can efficiently leverage a large amount of data. The Snowball [2] extracts entity pairs containing predefined relationship from corpus; the SatSnowball [8] extends the Snowball with statistical method and extracts the entity pairs and keywords around the entities. Both DIPRE [5] 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. Moreover, general form pattern cannot be directly used as a query for the Web search engine which is an efficient tool to retrieve texts on the Web. Therefore, the REX directly uses the context pattern.

Framework of Relation Expansion

The proposed REX framework uses a Web search engine to bootstrap the sentences containing similar relations to the given seeds. Figure 1. shows the architecture of REX. The *input* of REX is a small relation tuple set $S^0 = \{(e_i, c_j) \mid i=1, 2,\ldots, n; j=1,2,\ldots, m;\}$, where $e_i = (e_{ia}, e_{ib})$ is entity pair and c_j is context pattern. The *output* of REX is a ranked list of sentence tuples.

The *Expanding* part uses a dually extraction model. Let's note $E^0=\{e_i|(e_i, c_x)\in S^0\}$ and $C^0=\{c_j|(e_x, c_j)\in S^0\}$. E^0 and C^0 are the entity pairs and context patterns which are used for expansion in S^0. In t-th expansion iteration (at the beginning t=0), the context patterns C^t are used to generate some queries for the Context Search module and the entity pairs E^t are used to generate some queries for the Entity Pair Search.

Table 1. The Top 3 Selected Entity Pairs and Context Patterns

Extracted Top 3 Context Patterns

C◊O	A◊A	P◊B	C◊H
X's founder *Y*	*X* bought *Y*	*X* was born in *Y*	*X*'s headquarters in *Y*
X co-founder and chairman *Y*	*X* acquires *Y*	*X* born in *Y*	*X* campus in *Y*
X CEO and Chairman *Y*	*X* acquisition of *Y*	*X*, born in *Y*	*X* is located in *Y*

Extracted Top 3 Word Pairs

C◊O	A◊A	P◊B	C◊H
Steve Jobs::Apple	Facebook::FriendFeed	Obama::Kenya	Apple::Cupertino
Rogers::Duke Energy	Bloomberg::BusinessWeek	Jesus::Bethlehem	Yahoo::Sunnyvale
Raymond::Exxon	Twitter::Summize	Pablo Picasso::Malaga	Dell::Round Rock

We crawl the top 100 web pages returned by the Web search engine. The texts in these pages are split into sentences. In the Entity Pair Extraction module, a Named Entity Reorganization tool[1] is used to classify named entities in each sentence. All the entity pair candidates are extracted and added to the candidate set E_U^t. REX uses the label propagation algorithm [9] to select 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 candidate set S_{UC}^t. In the Context Pattern Extraction module, the contexts between given entity pairs are extracted and added to the context candidate set C_U^t. Some context patterns $C^{t+1} \subseteq C_U^t$ are selected for $t+1$ round of entity pair expansion. Meanwhile, the corresponding sentences are also added to S_{UE}^t.

The third part of REX is *Ranking*. In the expanding part, we get a set of sentence tuples S^* which is naturally represented from two views: entity pair view and context view. In this part, we rank the sentence tuples in S^* from the two views using a graph based multi-view learning algorithm [7]. In this process, at first, we use the search engine to generate the co-occurrence matrix. We combine entity pairs and context patterns together to generate the search queries. For example, the sentence tuple $((e_{ia}, e_{ib}), c_j)$ can generate a corresponding query: "e_{ia} c_j e_{ib}" or "e_{ib} c_j e_{ia}". The page count of each query is treated as the approximate co-occurrence frequency of entity pair and context pattern. Using this frequency, a co-occurrence matrix M is generated. The graph base multi-view learning algorithm uses M to rank all the sentence tuples.

Experiment

Although the REX framework can be used to extract any relation between two named entities, the named entity reorganization is a bottle neck of our framework. Beacuse the recognizer can only label four types of entities: Organization, Person, Location and Miscellaneous. Therefore, we test REX 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 three types of named entity returned by the recognizer. For each relation type, we give one seed for bootstrapping which is 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*).

In this experiment, the YahooBOSS API[2] is used to search with the given queries. Table 1. shows the top 3 extracted context patterns and entity pairs.

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 a given seed set. Experimental results show the effectiveness of relation expansion framework.

Reference

[1] C. Aron and S. Jeffrey. Dependency Tree Kernels for Relation Extraction. In *ACL'04*, pages 423—429, 2004.

[2] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *DL*, 2000.

[3] M. Banko, M. J. Cafarella, S. Soderl, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, pages 2670–2676, 2007.

[4] M. Banko and O. Etzioni. The tradeoffs between open and traditional relation extraction. In *ACL '08*, 2008.

[5] S. Brin. Extracting patterns and relations from the world wide web. In *WebDB Workshop of EDBT98*, 1998.

[6] G. Zhou, S. Jian, Z. Jie, and Z. Min. Exploring various knowledge in relation extraction. In *ACL '05*, 2005.

[7] H. Li, Y. Matsuo, and M. Ishizuka. Graph based multi-view learning for CDL relation classification. In *ICSC'09*, 2009.

[8] J. Zhu, Z. Nie, X. Liu, B. Zhang and J.-R. Wen. Statsnowball: a statistical approach to extracting entity relationships. In *WWW'09*, 2009.

[9] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Technical Report CMU-CALD-02-107, 2002.

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

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