Knowledge-Based Systems 83 (2015) 128–137

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Construction of semantic bootstrapping models for relation extraction

Chunyun Zhang [a,*], Weiran Xu [a], Zhanyu Ma [a], Sheng Gao [a], Qun Li [b], Jun Guo [a]

[a] *Pattern Recognition and Intelligent System Laboratory, Beijing University of Posts and Telecommunications, Beijing, China*
[b] *School of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing, China*

ARTICLE INFO

Article history:
Received 14 July 2014
Received in revised form 16 March 2015
Accepted 17 March 2015
Available online 25 March 2015

Keywords:
Relation extraction
Bootstrapping
Trigger word
Kernel
Pattern learning

ABSTRACT

Traditionally, pattern-based relation extraction methods are usually based on iterative bootstrapping model which generally implies semantic drift or low recall problem. In this paper, we present a novel semantic bootstrapping framework that uses semantic information of patterns and flexible match method to address such problem. We introduce formalization for this class of bootstrapping models, which allows semantic constraint to guide learning iterations and use flexible bottom-up kernel to compare patterns. To obtain the insights of reliability and applicability of our framework, we applied it to the English Slot Filling (ESF) task of Knowledge Based Population (KBP) at Text Analysis Conference (TAC). Experimental results show that our framework obtains performance superior to the state of the art.

1. Introduction

Relation extraction (RE) is an important but unsolved problem in information extraction (IE). It focuses on extracting structured relations from unstructured sources such as documents or webs, which can potentially benefit a wide range of natural language processing (NLP) tasks such as question answering, ontology learning, and summarization [1].

To solve the RE problem, a number of machine learning approaches have been recently applied. One common paradigm is the usage of bootstrapping [2] to learn relation patterns. The popularity of this framework lies in its ability to learn sufficient patterns and instances simply by iterations starting from a small number of seeds. Its central assumption is the pattern-relation duality principle [3] that good seed samples lead to good patterns, while good patterns help to extract good instances. Here, good patterns are usually referred to patterns that have high coverage (high recall) and low error rate (high precision), and good instances are instances that are realized by good patterns. Systems such as DIPRE [3], Snowball [4], and ExDisco [5] took a small set of domain-specific examples as seeds and an unannotated corpus as input. The seed examples can be either target relation instances or sample linguistic patterns in which the linguistic arguments correspond to the target relation arguments. New instances or new patterns will be found in the documents where the seed is located. The new instances or patterns will be used as new seed for the next iteration. However, Komachi' analysis in [6] showed that semantic drift is an inherent property of iterative bootstrapping algorithms and, therefore, poses a fundamental problem. Hence, these systems without semantic constraint are greatly troubled by the problem of semantic drift.

Relation patterns are defined as the structured features of the context of the entity and its attribute value (e.g. *Bill Gates* and *Microsoft* of the relation *org:founded_by* of organization entity) in a target relation mentioning [7]. Consequently, how well the system performs largely depends on how well patterns are represented. However, most existing patterns are with inflexible representation or without semantic constraint. Patterns in [3,4, 7–9] using shallow syntactic features have poor performances in the extraction of the relations that are ambiguous or lexically distant in their expression. Dependency patterns [10–15] have been shown to perform better, since they are more informative for relation extraction. The shortest dependency pattern (SDP) and the subject–verb–object (SVO) pattern, among other dependency patterns, are two commonly used patterns [10,1,12,13]. However, due to less semantic constraint, they gain the generality at the cost of lacking specific information and thus may produce semantic drift in bootstrapping iterations.

Similarity method, a measure which determines whether a pattern or instance derived from a new sentence is relation oriented or not, is another important key method for bootstrapping model. Unfortunately, the existing similarity methods are rigid or unsuitable for extracting relations expressed in complex structure patterns, since they cannot weigh the relative importance of different features of patterns only by using exact match method [3,7,8] or cosine-like method [12,4]. Kernel methods [16,10,11,17,15] have been proven to be effective in measuring

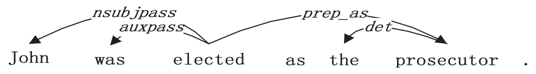
* Corresponding author.
E-mail address: zhangchunyun1009@126.com (C. Zhang).

http://dx.doi.org/10.1016/j.knosys.2015.03.017
0950-7051/© 2015 Elsevier B.V. All rights reserved.

Relation: *per:title*

Seed instance: <*John, prosecutor*>

Dependency tree of the corresponding seed sentence:



The shortest dependency path from entity to its attribute value:

John *nsubjpass* elected *prep_as* **prosecutor.**

The derived SDP:

<PERSON> *nsubjpass* **elected** *prep_as* **<title>.**

Fig. 1. Dependency tree of the seed sentence.

the similarity of two complex relation patterns. Most existing kernels [11,16,18,19] compared two patterns by following their structures from the root node to its child nodes through the syntax trees. However, these methods still have limitations in measuring different kinds of patterns, which degrade the performance of new relation extraction. For example, "*Bolin's son, Yorke B. Mizelle, is a good boy*" and "*Bolin is survived by her son, Yorke*" are two example sentences of relation *per:children* of person entity. The derived shortest dependency patterns [10] (shown in Fig. 3) that involve the respective root nodes of the two sentences son and survived may not be identified as similar by the kernel in [11,16]. It means that these kernels have poor performance in comparing weak relations that are not expressed by the main semantics of the sentences as the two sentences mentioned above.

To address the aforementioned problem, this paper proposes a general framework for semantic bootstrapping with a novel bottom-up kernel method. The framework represents relations with a semantic dependency pattern where trigger words are used as the semantic anchor. The usage of trigger words allows semantic constraint to guide the learning iterations. Furthermore, a novel flexible similarity method is proposed to compare similarities of patterns. We introduce a formalization for this class of models and illustrate how this model classes can be constructed. Our guiding hypothesis is that relation-mentioning will share similar structures in their dependency trees. We thus model relations by quantifying the degree to which relations are attested in similar dependency patterns. The expressive power of our framework stems from four parameters which guide model construction. The first parameter extracts trigger words of target relations. The second parameter determines what type of features contribute towards the representation of relation pattern. The third parameter allows us to weigh the relative importance of new derived patterns. Finally, the fourth parameter determines which patterns can be added as the seed patterns of next iteration.

We evaluate our framework on the English Slot Filling (ESF) [20] task of Knowledge Based Population (KBP) at Text Analysis Conference 2013 (TAC2013). The performances of methods can be evaluated by micro-average or macro-average. In this paper, to compare all evaluated methods in the same metric, we computed the micro-average precision, recall, and F1 value by using the metric defined in [21]. Because we have large enough samples in the experiment, we simply divided the corpus into the training set and the testing set to evaluate the performance. If the samples are limited, one can consider to use a cross-validation method [22]. The final experimental results show that our new model consistently outperforms former bootstrapping models yielding results superior to the state-of-the-art methods.

Our contributions are threefold: a novel framework for bootstrapping model of RE that incorporates semantic constraint information, uses a novel bottom-up kernel method to compare patterns, and generalizes existing bootstrapping models of RE; an application of this framework to the English slot filling task; and an empirical comparison of our semantic bootstrapping models against state-of-the-art bootstrapping models.

The rest of this paper is organized as follows: In Section 2, we give a brief overview of existing bootstrapping models of relation extraction. In Section 3, we present our modeling framework. Section 4 details experiments of parameters of semantic bootstrapping model and English slot filling. Discussion of our future work concludes the article in Section 5.

2. Overview of bootstrapping model of RE

Bootstrapping model has been proven to be useful framework for variety of information extraction tasks in natural language processing (NLP), such as named entity recognition [23–25], relation extraction [3,4,8] and question and answering [7]. This section will give a brief overview of existing bootstrapping models for RE.

The bootstrapping framework of RE was originally introduced in DIPRE system [3], which describes a duality principle that drives the bootstrapping process. However, it cannot avoid producing noisy and wrong patterns because it does not have a good mechanism to evaluating patterns and seeds. The Snowball system [4] developed the bootstrapping framework of DIPRE system with a three-tuple pattern representation and a new strategy for evaluating patterns and relation instances. It forms a standard bootstrapping framework of relation extraction which is still been used in many derived bootstrapping frameworks [8,14,26,17,5]. The framework can be formalized as following:

Definition 1. A bootstrapping model is a tuple $\langle R, I, P, rep, s, e\rangle$. R is the set of target relation. I and P are the set of seed instance and seed pattern of the target relation, which can achieve dual learning and can be expanded in iterations. rep is the pattern representation method. s is the similarity method, which maps relation mentioning sentences to seed instances. e is the evaluation method of newfound instances or patterns, and determines which one can be added as new seed of next iteration.

To illustrate the framework, we construct a framework of bootstrapping for the target relation *per:title*, using training corpus as the following sentences: "*Tomas was elected as the defense chief.*", using $\langle John, prosecutor\rangle$ as a seed instance and its corresponding seed sentence is "*John was elected as the prosecutor*". Fig. 1 shows the Stanford dependency analysis of the seed sentence. The dependency tree of the sentence is represented as a graph. The sentence head is the main verb *elected* which is modified by its passive nominal subject *John*, its passive auxiliary *was* and the *as* prepositional modifier prosecutor. The *as* prepositional modifier is modified by the determiner *the*. Next we will describe the existing corresponding methods of bootstrapping models in detail.

2.1. Pattern representation method

One of challenges in bootstrapping framework is how to learn selective patterns which have high coverage to represent relations. How well the system performs largely depends on how well the patterns are represented. An ideal relation pattern should be abstract over surface word orders and can mirror semantic relations as clearly as possible.

Traditional bootstrapping models, such as the Snowball system [3], Question and Answer system [7], and Espresso system [8] made use of named entity (NE) tags, surface strings, and their linear orders as components in the pattern representation:

$$rep(s) = \{p | p = left, tag_1, middle, tag_2, right\}, \qquad (1)$$