Information Processing and Management 000 (2017) 1–21

Contents lists available at ScienceDirect

Information Processing and Management

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

Self-training on refined clause patterns for relation extraction

Duc-Thuan Vo*, Ebrahim Bagheri

Laboratory of Systems, Software and Semantics (LS³), Ryerson University, Toronto, ON, Canada

ARTICLE INFO

Article history:
Received 11 August 2016
Revised 14 February 2017
Accepted 21 February 2017
Available online xxx

Keywords:
Relation extraction
Open information extraction
Self-training algorithm
Syntactic parsing
Dependency parsing

ABSTRACT

Within the context of Information Extraction (IE), relation extraction is oriented towards identifying a variety of relation phrases and their arguments in arbitrary sentences. In this paper, we present a clause-based framework for information extraction in textual documents. Our framework focuses on two important challenges in information extraction: 1) Open Information Extraction and (OIE), and 2) Relation Extraction (RE). In the plethora of research that focus on the use of syntactic and dependency parsing for the purposes of detecting relations, there has been increasing evidence of incoherent and uninformative extractions. The extracted relations may even be erroneous at times and fail to provide a meaningful interpretation. In our work, we use the English clause structure and clause types in an effort to generate propositions that can be deemed as extractable relations. Moreover, we propose refinements to the grammatical structure of syntactic and dependency parsing that help reduce the number of incoherent and uninformative extractions from clauses. In our experiments both in the open information extraction and relation extraction domains, we carefully evaluate our system on various benchmark datasets and compare the performance of our work against existing state-of-the-art information extraction systems. Our work shows improved performance compared to the state-of-the-art techniques.

1. Introduction

Relation Extraction (RE) is one of the important tasks in natural language processing, enabling information extraction and knowledge discovery from text. It aims at organizing relevant segments of unstructured text in relation triples that represent the relationship between two arguments through a relation. As part of an effort to infer more complex relational structures, relation extraction techniques aim to steer the extraction process away from the ambiguous extractions of semantic relations. Representing a particular set of relationships between two or more entities in text can be used for querying and automated reasoning. To infer complex relations, several approaches have been proposed, involving supervised learning (Abacha & Zweigenbaum, 2016; Bunescu & Mooney, 2005; Kambhatla, 2004; Ravichandran & Hovy, 2002; Zhou, Qian, & Fan, 2010), semi-supervised learning (Agichtein & Gravano, 2000; Batista, Martins, & Silva, 2015; Pantel & Pennacchiotti, 2006; Vo & Bagheri, 2015), and unsupervised learning methods (Akbik, Visengeriyeva, Herger, Hemsen, & Loser, 2012; Rosenfeld & Feldman, 2007; Turney, 2008; Yao, Riedel, & McCallum, 2012).

Among the supervised methods, Bunescu and Mooney (2005), Kambhatla (2004), Ravichandran and Hovy (2002), and Zhou et al. (2010) have focused on performing language analysis for semantic relation extraction. A running theme among

* Corresponding author.
E-mail addresses: thuanvd@ryerson.ca, thuanvd@gmail.com (D.-T. Vo), bagheri@ryerson.ca, ebrahim.bagheri@gmai.com (E. Bagheri).

http://dx.doi.org/10.1016/j.ipm.2017.02.009
0306-4573/

these techniques is the capacity to generate linguistic features based on syntactic, dependency, or shallow semantic structures of the text. Espousing these features, the models are subsequently trained to identify instances of entities that are related through relations. Once the identification process is underway, the extractions are classified based on pre-defined relation types. This is a laborious and time-consuming undertaking on the part of these approaches, involving the analysis of vast quantities of sample data.

Bootstrapping based pattern matching approaches have been employed by various researchers (Agichtein & Gravano, 2000; Brin, 1998; Greenwood & Stevenson, 2006; Pantel & Pennacchiotti, 2006) to extract patterns from seed relations. These approaches exploit the concept of information redundancy and hypothesize that similar relations tend to appear in uniform contexts. The work conducted by Batista et al. (2015) showed that semi-supervised bootstrapping techniques could be used for extracting semantic relations from text by iteratively expanding a set of initial seed relationships. In an effort to find similar relationships, these researchers investigated the effectiveness of bootstrapping for relationship extraction using word embeddings. Their model involves the use of a Named Entity Recognition (NER) module along with weak entity linking by matching entity names with Freebase concepts. In Xu, Uszkoreit, and Li (2007) and Xu, Uszkoreit, Krause, and Hong Li (2010), the authors' goal of extracting relations of various complexities is accomplished through bootstrapping with the ability to automatically learn pattern rules from parsed data. These researchers use dependency trees as the input for pattern extraction and work with trees or sub-trees that contain seed arguments. Despite their eagerness to maintain high accuracy, it is difficult to claim with certainty that the identified patterns are indeed accurate. In lieu of this, there is a probability that faulty seeds could potentially be injected into the bootstrapping process.

The presence of Open Information Extraction (OIE) (Banko, Cafarella, Soderland, Broadhead, & Etzioni, 2007; Etzioni, Fader, Christensen, Soderland, & Mausam, 2011; Fader, Soderland, & Etzioni, 2011; Nebot & Berlanga, 2014; Yahya, Whang, Gupta, & Halevy, 2014; Vo & Bagheri, 2016) offers a more nuanced approach that relies minimally on background knowledge and manually labeled training data. In this respect, various types of relations are taken into consideration without the need to restrict the search for pre-specified semantic relations. Banko et al. (2007), Wu and Weld (2010), and Fader et al. (2011) propose to use shallow syntactic representations of natural language text in the form of verbs or verbal phrases and their arguments. There has also been a more intense interest in approaches that employ robust and efficient dependency parsing for relation extraction (Akbik et al., 2012; Corro & Gemulla, 2013; Garcia & Gamallo, 2011; Mausam, Bart, & Soderland, 2012). Various heuristics are utilized to determine relevant segments of information based on shallow semantic representation or dependency parsing analysis by identifying factors that draw attention to whether two chunks of the original sentence exhibit connection, disconnection, or dependence on one another. Nonetheless, one of the serious drawbacks of techniques that are restricted to shallow syntactic and dependency analysis is detecting relations that display no connection between the verb or verbal phrases in the sentence. Existing state-of-the-art OIE systems the like of ReVerb (Fader et al., 2011) and ClausIE (Corro & Gemulla, 2013) extract relations that are mediated by verbs or verbal phrases based on dependency parsing. Despite key advantages to this approach, the failure to extract all potential relations beyond a pre-defined set of relations including syntactic entities such as nouns and adjectives along with a whole range of verbal structures can be problematic. For instance, consider the following sentence, as shown in Fig. 1, '*Maxus Energy Corp. discovered a new oil field in the southeast Sumatra area of Indonesia.*'. In this sentence, the relation between "*southeast Sumatra area*" and "*Indonesia*" cannot be determined by any type of verbs or verbal phrases through either syntactic or dependency parsing.

To address such limitations, we propose a clause-based framework with refinements to the grammatical structure. We use the English clause structure and clause types in an effort to generate propositions that can be deemed as extractable relations. The framework offers a unique advantage in that it is designed to address some of the more pressing limitations inherent in previous OIE systems through the reformation of the grammatical structure obtained from Syntactic Parsing (SP) and Dependency Parsing (DP). Moreover, an initial seed set generated by multiple high-confidence clause patterns is used for later integration into a bootstrapping process for extracting specified relations. Through the iterative expansion of the original seed set, our work allows for an increasing number of seeds to be identified that can ultimately lead to higher confidence relation extraction patterns. In this paper, our most significant contributions are as follows:

- We demonstrate that a clause-based approach with grammatical structure reformation can be a suitable method for open information extraction to address the following limitations:(1) Identifying relations that previous OIE systems have been oblivious to or overlooked altogether, e.g., the relation between "*southeast Sumatra area*" and "*Indonesia*" in the earlier example and (2) Reducing the number of erroneous relation extractions, e.g., the erroneous identification of '*there*' as a subject of a relation in the following sentence: "*In today's meeting, there were four CEOs*".
- We show that our framework is a suitable method of bootstrapping for relation extraction. It automatically builds an initial seed set based on high confidence clause patterns. Through the iterative expansion of the original seed set, the proposed bootstrapping method allows for an increasing number of seeds to be identified that can ultimately lead to higher confidence relation extraction patterns.

In our work, we empirically show that our framework is highly practical toward building systems for information extraction. We evaluated the approach by carrying out two sets of experiments on textual corpora in the form of 1) Open Information Extraction and 2) Bootstrapping Relation Extraction. The first set of experiments reveals that the approach utilized in our work improves the performance of leading OIE systems such as ClausIE (Corro & Gemulla, 2013), OLLIE (Mausam et al., 2012) and ReVerb (Fader et al., 2011). In the second set of experiments, we apply our proposed method on the standard and

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

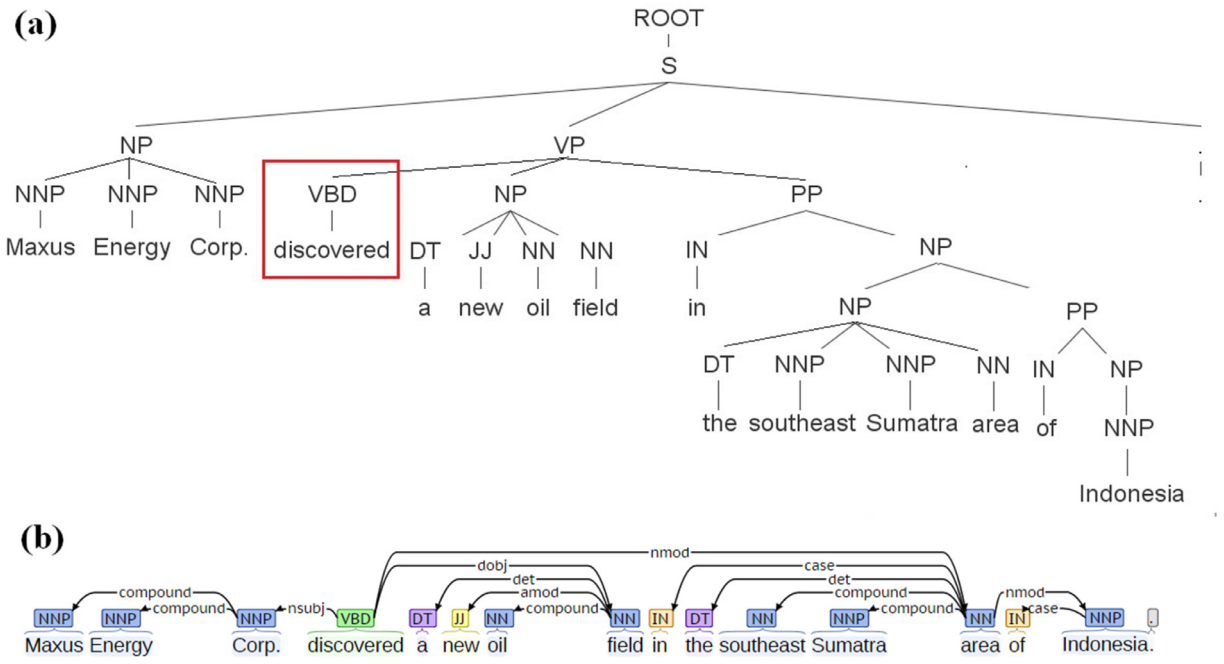

Fig. 1. (a) Syntactic and (b) dependency parsing.

widely used Nobel Prize and MUC-6 corpora. The experiments show that our approach improves upon the performance of the current state-of-the-art systems.

In the sections that follow, we begin by delving into the literature related to the concept of IE that provide further insight into the issues addressed in the current study; followed by a brief foray into the contexts that shapes this work. Section 3 offers a detailed description of our proposed approach. In Section 4, we present the methods used for OIE with regards to grammatical structure reformation. Section 5 offers a detailed description of a clause-based framework for information extraction. This is followed by a detailed description of our proposed method in Section 6 where we put forth methods used for RE with self-training. Section 7 offers an in-depth analysis of our experiments for IE where the results obtained from our proposed approach are compared to the state-of-the-art systems. In the last section, we draw conclusions about the merits of our work and offer ways to advance the literature in the future.

2. Related work

Relation extraction in general has become an active research topic during the past decade. The task of relation extraction was first introduced in the Message Understanding Conference (MUC). Since then, a number of techniques have been proposed for this task such as supervised learning (Abacha & Zweigenbaum, 2016; Bunescu & Mooney, 2005; Kambhatla, 2004; Ravichandran & Hovy, 2002; Singhal, Simmons, & Lu, 2016; Zhou et al., 2010), distant supervision (Angeli et al., 2014; Mintz, Bills, Snow, & Jurafsky, 2009, Riedel, Yao, McCallum, & Marlin, 2013; Surdeanu, Tibshirani, Nallapati, & Manning, 2012), deep learning (Santos, Xiang, & Zhou, 2015; Socher, Huval, Manning, & Ng, 2012; Xu et al., 2015; Zeng, Liu, Chen, & Zhao, 2015; Zeng, Liu, Lai, Zhou, & Zhao, 2014), unsupervised learning (Akbik et al., 2012; Etzioni et al., 2005; Oramasa, Espinosa-Ankeb, Sordoc, Saggionb, & Serraa, 2016; Rosenfeld & Feldman, 2007; Turney, 2008; Vlachidis & Tudhope, 2016; Yao et al., 2012) and bootstrapping methods (Agichtein & Gravano, 2000; Batista et al., 2015; Pantel & Pennacchiotti, 2006; Xu et al., 2007). In this milieu, the presence of Open Information Extraction (Corro & Gemulla, 2013; Mausam et al., 2012; Wu & Weld, 2010; Xu, Kim, Quinn, Goebel, & Barbosa, 2013) offers a more nuanced approach that relies minimally on background knowledge and manually labeled training data. In this respect, various types of relations are taken into consideration without the need to restrict the search to pre-specified semantic relations. In this section, we present several studies that are relevant to IE.

2.1. Supervised learning

In these approaches (Abacha & Zweigenbaum, 2016; Bunescu & Mooney, 2005; Choi & Kim, 2013; Kambhatla, 2004; Ravichandran & Hovy, 2002; Singhal et al., 2016; Zhou & Zhang, 2007; Zhou et al., 2010), there is a heavy reliance on hand-crafted datasets for training the extractor with manually pre-labeled training data. The advantage of these approaches is the use of linguistic patterns for learning information from different surface expressions. These approaches rely on pre-specification of desired relations or patterns by performing hand coding. The common strategy of these processes is to

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

generate linguistic features based on the analysis of the syntactic features, dependency features, or shallow semantic structure of text. These systems are trained to identify pairs of entities, and to classify them based on the pre-defined relations. Kambhatla (2004) used textual features such as POS, parsing, and NER to define features which include entities, types of entities (person, location), number of entities, number of words separating the two entities, and paths between the entities in a parse tree. Zhou and Zhang (2007) employed lexical, syntactic and semantic knowledge in feature-based relation extraction using support vector machines. The work by Zhou et al. (2010) illustrated how features can be constructed based on syntactic and semantic information from WordNet. Suchanek, Kasneci, and Weikum (2007) built an ontology by extracting relations from Wikipedia categories using WordNet and heuristic rules. Choi and Kim (2013) proposed a dependency trigram kernel based on Support Vector Machines (SVM) to classify the relationship between two persons' names in order to extract social relations. Reidel et al. (2013) used matrix factorization based on combining surface patterns extracted from OIE and knowledge bases such as Freebase to train latent relations. Abacha and Zweigenbaum (2016) trained an SVM classifier on the i2b2 2010 challenge's corpus. They used a set of lexical, morpho-syntactic and semantic features for each pair of medical entities (E1, E2) in order to be able to classify relations. Singhal et al. (2016) used supervised classifiers such as C4.5, Multilayer Perceptron, and Bayesian logistic regression on various types of features to identify relations of disease-mutation in biomedical text. While such approaches offer high precision and recall, most of them are laborious and expensive in training and face problems when handling large-scale text documents.

2.2. Unsupervised learning

Unsupervised approaches are usually based on rules or some clustering techniques over a large unlabeled corpus for relation discovery and extractions. Several approaches have been built based on latent relation hypothesis (Akbik et al., 2012; Rosenfeld & Feldman, 2007; Turney, 2008), latent topic assumption (Yao, Haghighi, Riedel, & McCallum, 2011; Yao et al., 2012), low rank assumption (Kok & Domingos, 2008; Takamatsu, Sato, & Nakagawa, 2011) and rule-based methods (Oramasa et al., 2016; Ryu, Jang, & Kim, 2015; Vlachidis & Tudhope, 2016). Turney (2008), Akbik et al. (2012) and Yao et al. (2012) exploit features from the dependency tree for discovering relations by clustering entity pairs. The cluster vector space model is often applied by using the k-mean algorithm and cosine similarity is used to measure distance. In rule-based approaches, Ryu et al. (2015) have defined a set of relationships on named entities such as Person, Location, and Data to support question answering in the Korean language. Similarly, Vlachidis and Tudhope (2016) have defined a set of rules based on syntactic analysis for extracting relation patterns within the archaeology domain. Further, Oramasa et al. (2016) have defined rules based on syntactic and semantic information to extract potential relations between entities, which have been discovered by traversing the dependency tree in the music domain. Since the assumptions largely rely on co-occurrence, previous unsupervised approaches tend to confuse correlated but semantically different phrases during extraction.

2.3. Distant supervision

The core idea of distant supervision is to learn a classifier based on a set of weakly labeled corpora that are often annotated using some heuristics. In the area of relation extraction, the work by Mintz et al. (2009) is among the pioneering works that consider the application of distant supervision techniques. In their work as well as other closely related work such as Surdeanu et al. (2012) and in order to curate the weakly labeled corpus, they use the Freebase knowledge base whereby for each pair of entities that are related to each other using some Freebase relation, they will identify sentences in their corpus where these entities have been seen together. This way they are able to extract features that can help them train a classifier for relation extraction. One of the challenges of distant supervision methods is the noisy labels, which are generated by the heuristics that will eventually lead to poor relation extraction performance. There have been works by Takamatsu, Sato, and Nakagawa (2012) and Min, Grishman, Wan, Wang, and Gondek (2013) among others that propose methods to identify low confidence labels that can be removed or ignored. From a different perspective and in order to augment the work in distant supervision, Riedel, Yao, and McCallum (2010) argue that many of the errors produced by relation extraction techniques are due to the generous interpretation of sentence relevance. In other words, if two entities were related to each other through a Freebase relation, any sentences containing these two entities would be considered related and labeled as such. The authors argue that this might not necessarily be the case, especially for cases when the knowledge base is not fully aligned with the corpus. For this reason, they propose the idea of expressed-at-least-once assumption and use constraint-driven semi-supervision without worrying about exactly which sentence expresses the relation.

2.4. Deep learning

In deep learning, several approaches address the task of extracting relations through the use of two major architectures of neural networks, namely Recursive Neural Network (RNN) (Socher et al., 2012; Xu et al., 2015) and Convolutional Neural Networks (CNN) (Santos et al., 2015; Zeng et al., 2014; Zeng et al., 2015). These approaches learn the hidden and continuous structures of relations on both internal features such as POS, Chunking, and Syntactic and/or external features such as word embeddings. Socher et al. (2012) presented the RNN model to learn compositional vector representations for phrases and sentences of arbitrary syntactic type and length in order to classify semantic relations between nouns using syntactic paths.

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

Xu et al. (2015) proposed an RNN model by exploiting long short-term memory units (LSTM) and shortest dependency path (SDP) to classify relation between two entities in a sentence. In their architecture, SDPs are used to retain most relevant information of the sentence while LSTMs are used as multichannel networks that can effectively integrate information from heterogeneous sources over the dependency paths. Zeng et al. (2014) presented a CNN model for relation classification where sentence-level features are learned through a CNN. In their CNN architecture, they extract lexical and sentence level features without complicated NLP preprocessing and assign pairs of words to targeted relations by encode the distances of the features relative to the position of the target noun pairs. Santos et al. (2015) proposed a Ranking CNN model that learns a distributed vector representation for relation classification. The network generates a distributed vector representation for the relations by using a ranking function in order to produce a score for each relation type.

2.5. Open information extraction

The emergence of a pioneering OIE system, called TextRunner, following the seminal work of Banko et al. (2007), brought a myriad of techniques to the fore in recent years. Currently, the majority of OIE systems use a shallow syntactic representation or dependency parsing in the form of verb or verbal phrases and their arguments. TextRunner uses automatically generated training data and syntactic analysis while WOE^{pos} (Xu et al., 2010) trains the corpus automatically by procuring infoboxes from Wikipedia. $\text{WOE}^{\text{parse}}$ (Xu et al., 2010) expands on this and uses automatically generated training data to learn extraction patterns on dependency parsing. ReVerb (Fader et al., 2011) extracts verb phrase-based relations building a set of syntactic and lexical constraints to identify relations based on verb phrases then finds a pair of arguments for each identified relation phrase. Mausam et al. (2012) have presented OLLIE, as an extension of the ReVerb system, which stands for Open Language Learning for IE. In OLLIE various heuristics are implemented to obtain propositions from dependency parsers. OLLIE performs deep analysis on the identified verb-phrase relations and then extracts all relations mediated by verbs, nouns, adjectives, and others. A more recent OIE system, named ClausIE, uses dependency parsing and a small set of domain-independent lexica without any post-processing or training data. At the outset, ClausIE (Corro & Gemulla, 2013) exploits linguistic knowledge about the grammar of the English language to first detect clauses in an input sentence and to subsequently identify each clause type based on the grammatical function of its constituents. As a result, ClausIE is able to generate high-precision extractions and can be flexibly customized to adapt to the underlying application domain. Our work builds on the foundations of ClausIE. We extend ClausIE by proposing to perform novel grammatical structure reformation for addressing the limitation on DP analysis from ClausIE that helps the system determine more accurate clause types for generating high-precision relations.

2.6. Weakly supervised and bootstrapping-based learning

The minimally supervised learning systems (Brin, 1998; Greenwood & Stevenson, 2006; Sudo, Sekine, & Grishman, 2003; Yangarber, Grishman, Tapanainen, & Huttunen, 2000) engender a context within which the concept of information redundancy is used in conjunction with bootstrapping. Through the adoption of different methods, minimally supervised learning systems attempt to estimate the confidence of the learned patterns for relation extraction. Sudo et al. (2003), Yangarber et al. (2000) and Greenwood and Stevenson (2006) are among a myriad of scholars who have opted to calculate domain relevance by relating the frequency of a term in domain relevant documents. Based on bootstrapping, the central goal of prevalent minimally supervised learning systems is to identify relation patterns that can lead to the identification of newer seeds and patterns. Here, the domain relevance of documents is used to discover patterns along with the distribution frequency of said patterns in relevant documents as an indicator of good patterns. Some other bootstrapping approaches (Agichtein & Gravano, 2000; Gupta & Manning, 2014; Xu & Zhang, 2014; Zhang, Xu et al., 2015; Zhang, Zhang et al., 2015) have proven to be effective methods to generate high-precision relation patterns when the set of labeled instances are limited. These works aim to expand an initial “seed” set of instance with new relationship instances. Documents are detected for entities from the seed instances and linguistic patterns connecting them are extracted with a similarity measure between the new patterns and the ones in the seed set.

Xu et al. (2007) presented Domain Adaptive Relation Extraction system (DARE), which is comprised of four major components including linguistic annotation, classifier, rule learning, and relation extraction. The second component, rule learning, is used to identify and extract relations of varying complexities through a seed-driven bottom-up process. Xu et al. use the dependency tree as the input for pattern extraction. In view of the possibility that the DARE model might include faulty seeds in the bootstrapping process, its performance demonstrates weaker results when used against unobserved new domains. This is due to the high probability that DARE extracts incorrect rules from the dependency tree during the bootstrapping process. Xu et al. (2010) extend DARE using supervised learning to build seeds by observing the learning rules. Despite the immense advances brought on by such an approach for improving precision/recall, there is still need for manual semantic annotation in these approaches. In our work, we also present a bootstrapping approach, but in contrast, our work largely avoids such errors by exploiting an initial seed set without the need for manual input for bootstrapping. We automatically build an initial seed set for later iterations based on high confidence patterns from clause patterns extracted from Open IE. To ensure that a seed has high confidence, it is essential for it to be generated by multiple high-confidence patterns.

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

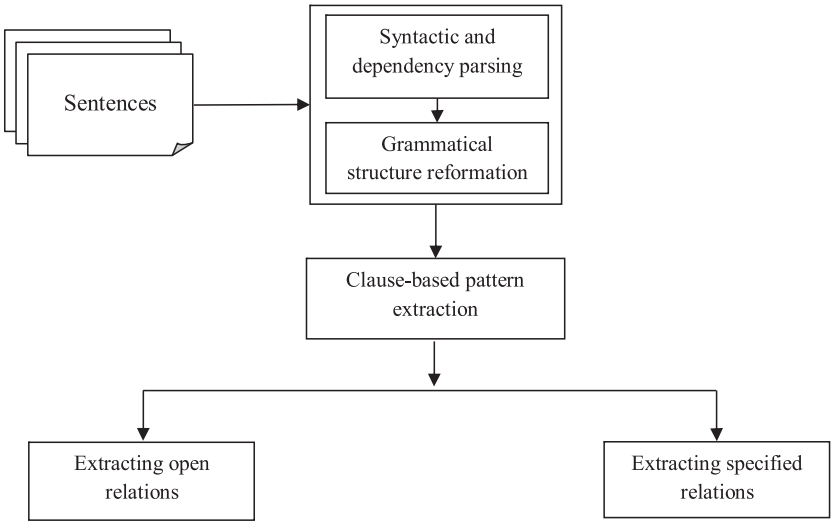

Fig. 2. Overview of the proposed framework.

3. Overview of the proposed framework

Typical work in this area extracts triples in the form of (arg1, rel, arg2), representing basic propositions or assertions from text. In this context, propositions are defined as coherent and non-over-specified pieces of basic information. In this section, we will present a framework for relation extraction shown in Fig. 2. Our framework handles two tasks in RE: (T1) extracting open relations and (T2) extracting specified relations. In our work and inspired by Corro and Gemulla (2013), we focus on the English grammar *clause structure*. The Oxford dictionary defines a clause as "A unit of grammatical organization next below the sentence in rank and in traditional grammar said to consist of a subject and predicate". While the literature (Quirk, Greenbaum, Leech, & Svartvik, 1985) is replete with definitions of clause as a part of a sentence that expresses some coherent piece of information, our approach moves beyond this and refines the tree structure produced from syntactic and dependency parsing. We propose a novel grammatical structure reformation on the product of the syntactic parser to add necessary relation nodes and removing noise nodes in order to derive a set of coherent constituents for generating propositions that can produce correct extractable relations. For the first task (T1), we will extract open relations where the system makes a data driven pass over its clause patterns without requiring background knowledge and manually labeled training data. In this respect, various types of relations are taken into consideration without the need to restrict the search to pre-specified semantic relations. For the second task (T2), with respect to each clause, the corresponding clause type will be determined pursuant to the grammatical function of its coherent constituent. The emergent patterns for the determined clause type will be used to extract specified relations. Subsequently, we propose a self-training algorithm based on bootstrapping that uses the patterns identified in the first step to automatically derive the required seeds. We learn context clues from the learned seeds and use the clues to identify the category of a particular relation. The approach proposed here eliminates the need for a manually prepared seed set at the onset and instead opts to automatically extract the required seeds from high confidence patterns extracted in clause-based extraction. Through the iterative expansion of the original seed set, bootstrapping allows for an increasing number of seeds to be identified that can ultimately lead to higher confidence relation extraction patterns.

4. Grammatical structure reformation

Considering the fact that a relation candidate is surrounded by words before, between, or after the relation pair, as well as the combination of two consecutive positions, the clause structure can be posed as a suitable grammatical structure for identifying relations in a sentence (Corro & Gemulla, 2013; Thenmozhi & Aravindan, 2015). It should be noted that a clause could consist of different components including subject (S), verb (V), indirect object (O), direct object (O), complement (C), and/or one or more adverbials (A). As previously indicated, the use of syntactic and dependency parsing has the potential to reduce precision at higher points due to incoherent information extractions after parsing. Normally relying merely on syntactic or dependency parsing based on verb or verbal phrases to determine relations has a tendency to engender certain problems. This is particularly true in dealing with sentences that do not exhibit sufficient information in order to create a connection between the subject, verb, and object of a relation. For instance, the relation between "*southeast Sumatra area*" and "*Indonesia*" in Fig. 1 cannot be determined in the absence of a verb or verbal phrase that is required to describe the relation. Hence, the grammatical tree structure of "*southeast Sumatra area*" and "*Indonesia*" needs to be essentially refined by adding new relation nodes (e.g., a dummy R relation) with associated links between "*southeast Sumatra area*" and

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

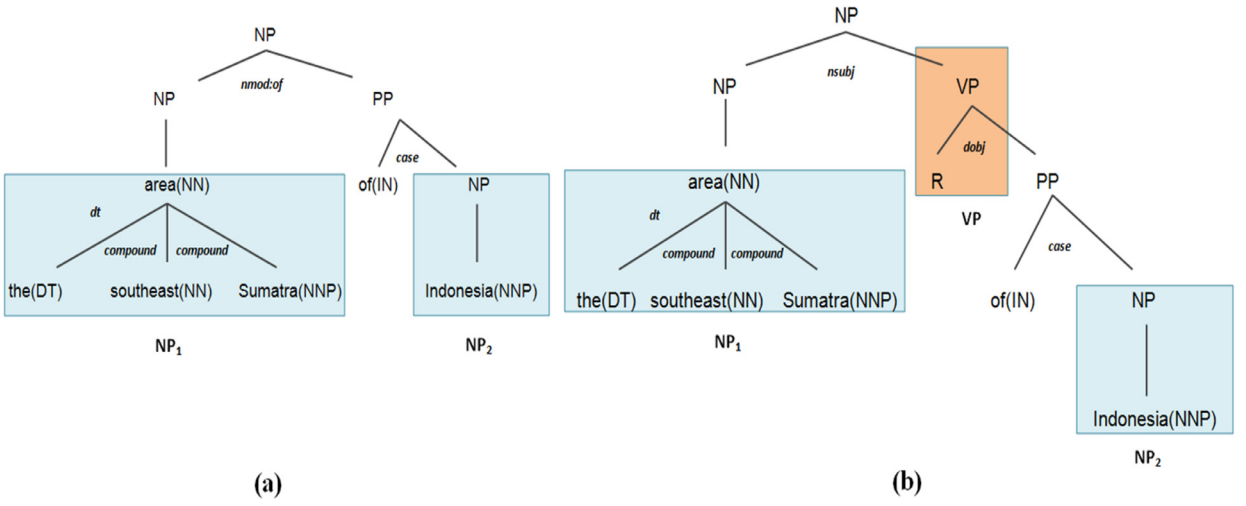

Fig. 3. (a) Shortest path between "*the southeast Sumatra*" and "*Indonesia*"; (b) the refined tree in sentence "*Maxus Energy Corp. discovered a new oil field in the southeast Sumatra area of Indonesia.*"

"*Indonesia*". This is of significance due to the fact that the relation between these structures can come to be explicitly observed and subsequently extracted. In our effort to refine the grammatical structure to improve clause-based relation extraction techniques, the following is the details of our proposed approach for the refinement of the grammatical tree structure.

We extract the shortest path (Bunescu & Mooney, 2005; Croce, Moschitti, & Basili, 2011) between two potential entity heads detected with SP and DP and use them for grammatical structure reformation. Potential entities such as Subject, Object and Complement are considered for arguments in the relation. We determine them by analyzing noun phrases in SP, and `nsubj` and `dobj` components in DP. To detect entities, all dependent components related with entities in NP phrases are also extracted. This process includes the extraction of modifier types (Marneffe & Manning, 2008) in DP, which are associated with entities involved in noun compounds, adverbs, and adjectives in the phrase. We stop when approaching the boundary of the target noun phrase. For instance, in the earlier example "*the southeast Sumatra area*" and "*Indonesia*" will be extracted as entities based on DP and SP analysis shown in Fig. 1. Words such as "*the*", "*southeast*" and "*Sumatra*" are connected to "*area*" via noun compound relationships defined as `det(area-13,the-10)`,`compound(area-13,southest-11)`and `compound(area-13, Sumatra-12)`. Also, words such as "*Indonesia*" is presented in NP and connected with "*area*" via modifier relationship as `nmod(area-13, Indonesia-15)`. Extraction of potential entities in the sentence could be presented in a new grammatical structure in which latent relations could be recognized. To this end, the grammatical structure is refined in the following manner:

1. Cases where the shortest path does not consist of either the "Subject-Verb-Object" or "Subject-Verb-Complement" structures as shown in Fig. 3(a): Our goal is to look for a structure "NP_1-PP-NP_2" where a latent relation could present the connection between "NP_1" and NP_2". When finding such a structure, an R (relation) node will be added as a central relation in the tree and R will be associated with the two phrases. NP_1 that holds the first entity ("*the southeast Sumatra area*") will represent the first argument and will be connected to R. The remaining part of the structure "PP-NP_2" will be connected to R on its right-hand side. We also order the nodes of the dependency tree in a way to place the dominant nodes on top and the dependent nodes at the bottom. Fig. 3(b) depicts the refined form when a new R node is added in the tree structure between the NP and PP phrases, hence creating a relation between "*the southeast Sumatra area*" and "*Indonesia*".
2. Cases when the shortest path "NP_1-VP-NP_2" does not have a structure in the form of "Subject-Verb-Object" or "Subject-Verb-Complement": These cases are when the structure "Subject-Verb" is observed. Besides detecting potential entities in NP discussed above, verb or verb phrases need to be detected in this case. We will start form Verb phrases (VP) to extract all dependent components related with the main verb in the phrase. We then stop when approaching the boundary of the target verb phrase. In these instances, we propose refining the structure by reversing NP_2 to NP_1. For instance, in Fig. 4(a) the two phrases "*four CEOs*" and "*were*" are indicative of the relationship "Subject-Verb" in DP. We add the remaining NP_2 to the structure by identifying the PP that has the closest connection with the VP. The refined structure of "*four CEOs*", "*were*" and "*in today's meeting*" is demonstrated in Fig. 4(b). In this context, the intended extraction of the wrong main subject, which in this case is "*There*" is replaced by "*four CEOs*".
3. Cases where the shortest path consists of a "Subject-Verb-Object" or "Subject-Verb-Complement": Such structures could be determinants of a relation; therefore, no changes to the structure of the tree are required. Our proposition is that the nodes of the dependency tree can be ordered in a way so as to place the dominant node at the top and the dependent

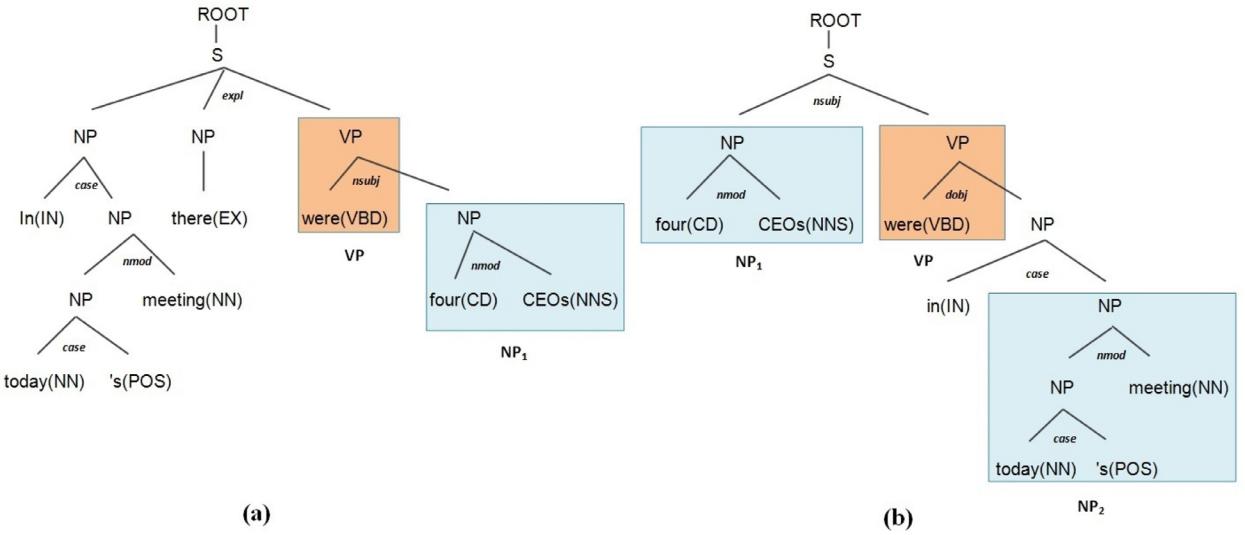

Fig. 4. (a) Shortest path tree between "*In today's meeting*" and "*four CEOs*" and (b) refined tree in sentence "*In today's meeting, there were four CEOs*."

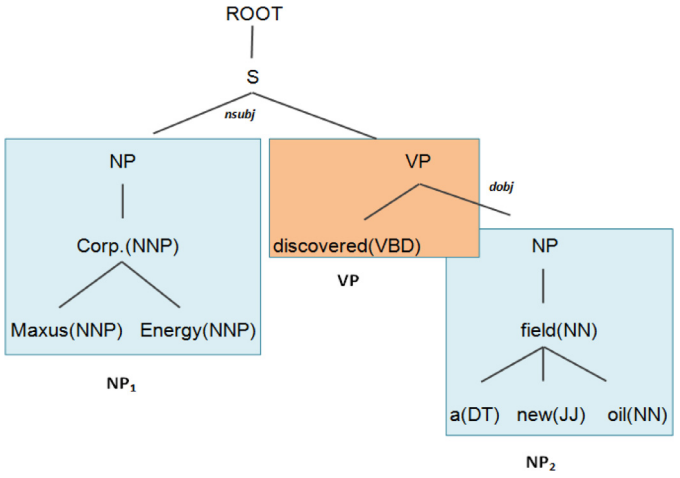

Fig. 5. Shortest path between "*Maxus Energy Corp.*" and "*a new oil field*" the verb "*discovered*" in "*Maxus Energy Corp. discovered a new old field in the southeast Sumatra area of Indonesia.*"

at the bottom. We recommend keeping the main phrases containing the associated links around the verbal phrase (VP) with two arguments in noun phrase (NP). For example, Fig. 5 displays a structure where the VP phrase that contains the verb "*discovered*" has associated links with the NPs containing "*Maxus Energy Corp.*" and "*a new oil field*".

On account of these three refinements on the trees, new forms of logical relationships become visible that in turn expedite the extraction of increasingly more accurate relations.

5. Clause-based pattern extraction

As previously pointed out, a clause can consist of different components such as subject (S), verb (V), direct/indirect object (O), complement (C), and/or one or more adverbials (A). As illustrated in Table 1, a clause can be categorized into different types based on its constituent components. For instance, the clause type for "*Albert Einstein remained in Princeton*" is SVA with Subject: "*Albert Einstein*", Verb: "remained in" and Adverbial: "*Princeton*". For each clause, we determine the set of coherent derived-clauses based on the syntactic and dependency tree after refining the tree structure. Following Corro and Gemulla (2013), we first build a clause by starting to extract all subject dependencies in the DP. This includes the subject (S) and the governor of the verb (V) in the sentence. Second, all other constituents of the clause such as objects (O) and complements (C) extracted though `dobj`, `iobj`, `xcomp` or `ccomp`; and adverbials (A) extracted though dependency relations such as `advmod`, `advcl`, or `prep_in` that are dependent of the verb will be extracted for generating a clause. We obtain and exploit clauses for the purpose of relation extraction in the following manner:

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

Table 1
Sample clause types (Corro & Gemulla, 2013; Quirk et al., 1985); S: Subject, V: Verb, A: Adverbial, C: Complement, O: Object.

Clause types	Sentences	Patterns	Derived clauses
SV	Albert Einstein died in Princeton in 1955.	SV	(Albert Einstein, died)
		SVA	(Albert Einstein, died in, Princeton)
		SVA	(Albert Einstein, died in, 1955)
		SVAA	(Albert Einstein, died in, 1955, [in] Princeton)
SVA	Albert Einstein remained in Princeton until his death.	SVA	(Albert Einstein, remained in, Princeton)
		SVAA	(Albert Einstein, remained in, Princeton, until his death)
SVC	Albert Einstein is a scientist in the 20th century.	SVC	(Albert Einstein, is, a scientist)
		SVCA	(Albert Einstein, is, a scientist, in the 20 the century)
SVO	Albert Einstein has won the Nobel Prize in 1921.	SVO	(Albert Einstein, has won, the Nobel Prize)
		SVOA	(Albert Einstein, has won, the Nobel Prize, in 1921)
SVOO	RSAS gave Albert Einstein the Nobel Prize.	SVOO	(RSAS, gave, Albert Einstein, the Nobel Prize)
SVOA	The doorman showed Albert Einstein to his office.	SVOA	(The doorman, showed, Albert Einstein, to his office)
SVOC	Albert Einstein declared the meeting open.	SVOC	(Albert Einstein, declared, the meeting, open)

Step 1. Determining clauses and clause types

According to Corro and Gemulla (2013), we exploit an algorithm for generating clause types shown in Algorithm 1. The algorithm will start when finding the subjects (S) and the governor of the verb (V) from DP. This step seeks to identify the clauses in the input sentence by obtaining the head words of all the constituents of every clause. The mapping of syntactic and dependency parsings are utilized to identify various clause constituents. Subsequently, a clause is constructed for every subject dependency, dependent constitutes of the subject, and the governor of the verb. An example of this is the construction of clause relation (subject: "*Maxus Energy Corp.*", verb "*discovered*") in Fig. 5 via `nsubj` and `compound` for the subject, `nsubj` and `dojb` for the verb. Moreover, subjects like relative pronouns that have been obtained through the `rcmod` dependency (e.g., which or who) and reference a word in the DP or correspond to an artificially created verb are replaced by their antecedent. For instance, in the sentence "*Obama, who is the president of the U.S, came to Canada.*" '*who*' is replaced by '*Obama*' in the extracted patterns.

When Subjects (S) and Verbs (V) are obtained, they need to be associated with one of the main clause types as shown in Table 1. In this process, the algorithm will seek Objects (O), Complements (C) or Adverbs (A) as in Line 2, 10, and 12 of the algorithm for finding clause types as SVO, SVC or SVA, respectively. Clause types SVOO, SVOA, SVOC will be identified by the structure of the clause. Particularly, we will seek direct/indirect Objects for generating the SVOO clause type as in Line 4 or will seek C for generating the clause SVOC in Line 6. Otherwise, we will seek A for generating SVOA as in Line 8 of the algorithm. Moreover, clause types SVC, SVOO, and SVOC are identified solely based on the structure of the clause. All adverbials are optional for whole types. For examples shown in Table 1, the derived clauses from SVO are SVO and SVOA, or the derived clauses from SVC are SVC and SVCA.

Step 2. Extracting open relations

We extract relations from a clause based on the patterns of the clause type as illustrated in Table 1. Assuming that a pattern consists of a subject, a relation and one or more arguments, it is reasonable to presume that the most reasonable choice is to generate n-ary propositions that consist of all the constituents of the clause along with some arguments. To generate a proposition as a triple relation (arg1, rel, arg2), it is essential to determine which part of each constituent would be considered as the subject, the relation and the remaining arguments. We initially identify the subject of each clause and then use it to construct the proposition. To accomplish this, we map the subject of the clause to the subject of a proposition relation. This is followed by applying the patterns of the clause types in an effort to generate propositions. For instance, for the clause type SV in Table 1, the subject of the clause ("*Albert Einstein*") is used to construct the proposition with the following potential patterns: SV, SVA, and SVAA. We then recommend using DP to forge a connection between the different parts of the pattern. As a final step, n-ary facts are extracted by placing the subject first followed by the verb or the verb with its constituents. This is followed by the extraction of all the constituents following the verb in the order in which they appear. As a result, we link all arguments in the propositions in order to extract triple relations.

6. Self-training algorithm

Broadly speaking, bootstrapping methods begin with an un-annotated corpus and a small set of hand-tagged seed words. In contrast to bootstrapping approaches that require an input seed set, we propose a new self-training method based on bootstrapping that benefits from the patterns extracted from the previous step (Section 5) to identify and extract relations from the corpus. The emergent patterns for the determined clause types will be used to extract specified relations. The method proposed here eliminates the need for a manually prepared seed set at the onset and instead opts to automatically extract the required seeds from high confidence extracted patterns. We learn context clues from the learned seeds and use the clues to identify the category of a particular relation. The words in these relations are assigned to a seed set in order to incrementally complete a lexicon that can be used for further bootstrapping. Our method retrains using the new updated

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

Algorithm 1: Clause pattern generation.

```
Input:  DP analysis
Output: list of clause patterns
if found(S,V) do
    if seek(O)
        if seek(direct O) and seek(indirect O) do
            generate SVOO
        else if seek(C) do
            generate SVOC
        else if seek(A) and seek(direct O) do
            generate SVOA
        else generate SVO
    else if seek(C) do
        generate SVC
    else if seek(A) do
        generate SVA
    else generate SV
end if
```

seeds and the process is repeated iteratively. In light of the fact that the extracted patterns are organized with the (a_1, r, a_2) structure, we organize the seeds in the form of $S(E, R)$ where $E=\{a_1, a_2, \ldots, a_n\}$ and $R=\{r_1, r_2, \ldots, r_m\}$.

The basic idea behind our self-training algorithm is that the system takes a few initial selected seeds and a set of patterns T from Open IE as input and learns further patterns based on the initial selected seeds. The algorithm begins by scoring each pattern t of set of patterns T and selecting the top-k scored patterns, which will then be inserted into a *pattern-pool*. In order to score the patterns, we utilize the scoring function introduced in Patwardhan and Riloff (2007) and Thelen and Riloff (2002), known as the *RlogF* metric, which has already been used for learning lexicons in previous studies. The *RlogF* metric scores each extracted pattern by calculating the occurrences of the arguments and the relation of a given pattern within the seed set. Eq. (1) showcases the approach proposed in our work regarding the implementation of the *RlogF* metric.

$$RlogF(t_k) = \frac{\left(F_k^a \times log_2\left(F_k^a\right)\right) + \left(F_k^r \times log_2\left(F_k^r\right)\right)}{N_k} \tag{1}$$

where F_k^a is the number of argument seeds extracted by pattern t_k, F_k^r represents the number of relation seeds extracted by pattern t_k, and N_k stands for the total number of words extracted by pattern t_k. Immediately after all the extracted patterns are ranked using the *RlogF* metric, the *top-k* patterns with the highest score are selected and added to the *pattern-pool*. The ensuing step involves scoring the candidate seeds in the top-k selected patterns within the *pattern-pool*. Candidate seeds comprise of nouns, compound nouns, and verbs observed in the arguments and relations of the extracted patterns. For each candidate seed, the algorithm collects all the patterns used to produce the candidate seed in question. Our algorithm scores the candidate seed by computing the average number of patterns that are extracted by that seed. Eqs. (2) and (3) detail the

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

Table 2
Overview of the precision of the six systems.

	ReVerb	Wikipedia	NYT
TextRunner	35.84% (286/798)	n/a	n/a
WOE	43.48% (447/1028)	n/a	n/a
ReVerb	53.37% (388/727)	66.26% (165/249)	54.98% (149/271)
OLLIE	44.04% (547/1242)	41.41% (234/565)	42.46% (211/497)
ClausIE	50.37% (1182/2348)	49.56% (397/797)	52.67% (493/936)
LS3RyIE	67.77% (1642/2425)	68% (614/903)	70.19% (690/983)

scoring method:

$$AvgLog(a_i) = \frac{\sum_{j=1}^{P_i} log_2\left(F_j^a + F_j^r + 1\right)}{P_i} \tag{2}$$

$$AvgLog(r_i) = \frac{\sum_{j=1}^{P_i} log_2\left(F_j^a + F_j^r + 1\ \right)}{P_i} \tag{3}$$

where P_i is the number of patterns that extract e_i/r_i. Also, F_j^a and F_j^r denote the number of entity seeds and relation seeds extracted by pattern j, respectively. Candidate a_i and r_i with the high score in the *pattern-pool* will be added to S. In each iteration, top-k patterns from the *pattern-pool* are selected and removed until the *pattern-pool* is deplete of any patterns. It is through this process that we obtain a list of updated patterns allowing us to categorize other patterns as they are extracted.

7. Experimentation

In this paper, we have presented a framework based on clause-based patterns aiming at two tasks: T1) Open information extraction and T2) Specified relation extraction. In this section, to carry out evaluations on our method for these tasks, we conduct experiments on several benchmark datasets and compare the performance of our proposed work with state of the art systems. Particularly, in the first task we use three different benchmark datasets, namely ReVerb, Wikipedia, and the New York Times datasets. Furthermore, MUC-6, and Nobel Prize corpora will be used for the second task. We will show how our proposed work can identify hidden relations and reduce extracting the number of erroneous relations compared to previous RE systems.

7.1. Open information extraction (T1)

7.1.1. Experimental dataset

In this task, we adopt the evaluation strategy proposed in Corro and Gemulla (2013) and use three different benchmark datasets, namely ReVerb, Wikipedia, and the New York Times. We used the Stanford parser to perform SP and DP on the sentences derived from the three standard benchmark datasets. In the first dataset, the ReVerb dataset that consists of 500 sentences has been extracted using Yahoo's random link service with manually labeled extractions from the Web. The sentences may have irrelevant phrases due to noise in the Web texts. The second dataset is comprised of 200 random sentences extracted from Wikipedia. These sentences have a tendency to be shorter and simpler than the ones extracted from the ReVerb dataset. Bearing in mind that a considerable majority of information in Wikipedia is generally edited by non-native speakers, at times the sentences display incorrect grammatical structures but they are less noisy than the information from the first dataset. The third dataset is comprised of 200 random sentences extracted from the New York Times collections (NYT). While these sentences are generally very clean, they are prone to be long and complex. Our experiments were carried out on these benchmark datasets in an effort to analyze the reliability and reproducibility of our work. Additionally, we compared our proposed approach with other OIE baselines including ClausIE, OLLIE, ReVerb, WOE and TextRunner. It should be noted that TextRunner and WOE have not publicly disclosed their code base confining us to use the results they reported in their publication for the ReVerb dataset and not for Wikipedia and the NYT datasets, marked as n/a in Table 2.

We manually labeled and verified all of the extractions by ReVerb, OLLIE and ClausIE systems from the aforementioned three datasets. To guarantee consistency among the labels, the ReVerb dataset is relabeled following the original labels from TextRunner, ReVerb, and ClausIE. As for the Wikipedia and the NYT datasets, each extraction is relabeled in accordance to the same output result from ClausIE. Each extraction was labeled by two independent experts, which was considered to be the gold standard. The experts were instructed to treat an extraction as correct if it was both informative and devoid of extraneous information. The correct extraction had to be approved and labeled as correct by both experts. In contrast, the extractions that lacked meaning were labeled as incorrect. The experts' ruling was measured using Cohen's Kappa with 0.57 on the ReVerb dataset, 0.68 on the Wikipedia dataset, and 0.63 on the NYT dataset.

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

Table 3
Extraction samples with Correct: 1 and Incorrect: 0.

OIE systems	Triple	Label
Sentence: "*Continuing to maintain his innocence of terrorism charges about a bombing in Iraq, Salim said the stabbing was unrelated to the escape plan and stemmed instead from his dissatisfaction with his lawyers.*"		
LS3RyIE	r_1: ("Salim", "be continuing to maintain", "his innocence of terrorism charges")	0
	r2: ("Salim", "said", "the stabbing was unrelated to the escape plan and stemmed instead from his dissatisfaction with his lawyers Continuing to maintain his innocence of terrorism charges")	1
	r_3: ("Salim", "said", "the stabbing was unrelated to the escape plan and stemmed instead from his dissatisfaction with his lawyers")	
	r_4: ("the stabbing", "was unrelated to", "the escape plan")	1
	r_5: ("the stabbing", "was", "unrelated")	0
	r_6: ("the stabbing", "stemmed", "instead")	0
	r_7: ("the stabbing", "stemmed instead from", "his dissatisfaction with his lawyers")	1
	r_8: **("he", "has", "a innocence of terrorism charges")**	1
	r_9: **("a bombing", "was in", "Iraq")**	1
	r_{10}: **("his dissatisfaction", "is with", "his lawyers")**	1
	r_{11}: **("he", "has", "a dissatisfaction")**	1
	r_{12}: **("he", "has", "lawyers")**	1
ClausIE	r_{13}: ("his", "has", "innocence of terrorism charges")	0
	r_{14}: ("Salim", "be Continuing", "to maintain his innocence of terrorism charges")	0
	r_{15}: ("Salim", "said", "the stabbing was unrelated to the escape plan and stemmed instead from his dissatisfaction with his lawyers Continuing to maintain his innocence of terrorism charges")	1
	r_{16}: ("Salim", "said", "the stabbing was unrelated to the escape plan and stemmed instead from his dissatisfaction with his lawyers")	1
	r_{17}: ("the stabbing", "was", "unrelated to the escape plan")	1
	r_{18}: ("the stabbing", "was", "unrelated")	0
	r_{19}: ("the stabbing", "stemmed", "instead from his dissatisfaction with his lawyers")	1
	r_{20}: ("the stabbing", "stemmed", "instead")	0
	r_{21}: ("his", "has", "dissatisfaction with his lawyers")	0
	r_{22}: ("his", "has", "lawyers")	0
OLLIE	r_{23}: ("the stabbing", "was unrelated to", "the escape plan")	1
	r_{24}: ("the stabbing", "was", "unrelated")	0
ReVerb	r_{25}: ("Salim", "said", "the stabbing")	0
	r_{26}: ("the stabbing", "stemmed instead from", "his dissatisfaction")	0
	r_{27}: ("the stabbing", "was unrelated to", "the escape plan")	1
WOE	r_{28}: ("Salim", "said stemmed from", "his dissatisfaction")	0
	r_{29}: ("Salim", "said unrelated to", "the escape plan")	0
	r_{30}: ("the stabbing", "stemmed from", "his dissatisfaction")	1
	r_{31}: ("the stabbing", "was unrelated to", "the escape plan")	1
TextRunner	r_{32}: ("Continuing", "to maintain", "Salim")	0
	r_{33}: ("Continuing", "to maintain", "his innocence of terrorism charges")	0
	r_{34}: ("his innocence of terrorism charges", "said", "the stabbing")	0
	r_{35}: ("the stabbing", "was to stemmed from", "his dissatisfaction")	0

7.1.2. Experimental results

The results of our proposed approach, which we call LS3RyIE, and the comparison to the other state-of-the-art OIE systems are presented in Table 3 and Fig. 6 on the three standard benchmark datasets. Fig. 6 plots the precision of each OIE system ordering them in decreasing confidence as a function of the number of extractions. It can be observed that LS3RyIE outperforms ClausIE, OLLIE, and ReVerb. The relative quality differences between our proposed approach and the state-of-the-art OIE systems employed in this study were essentially improved in all three datasets. The results reveal that we obtained 67.77% precision on the ReVerb dataset, 68% precision on the Wikipedia dataset, and 70.19% precision on the NYT dataset. The increase in precision is obtained through the discovery of hidden relations in addition to the removal of unrecognized relations due to the grammatical reformation proposed in our work. LS3RyIE identified 2425 extractions in the ReVerb dataset, 903 extractions in Wikipedia and 983 extractions in the NYT dataset. These extractions were higher in number compared to the other OIE systems. The precision of TextRunner was significantly lower than the other systems on the ReVerb dataset. The other systems obtain high precision on high-confidence extractions; the precision drops based on low confidence values in each extraction except for the ClausIE system on the NYT dataset (Fig. 6C). ClausIE identifies numerous incorrect extractions in possessive clauses, e.g., ("*his*", "*has*", "*a computer*"), with high confidence values, which is prevented in our work due to the proposed grammatical structure refinements.

7.1.3. Discussions

7.1.3.1. Output samples analysis. Several sample relations extracted from a similar sentence using each of the OIE baseline systems are demonstrated in Table 3. In light of the fact that our proposed approach (LS3RyIE) and ClausIE explore the clause structure of the sentence, the two systems have demonstrated an ability to extract the highest number of relations,

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

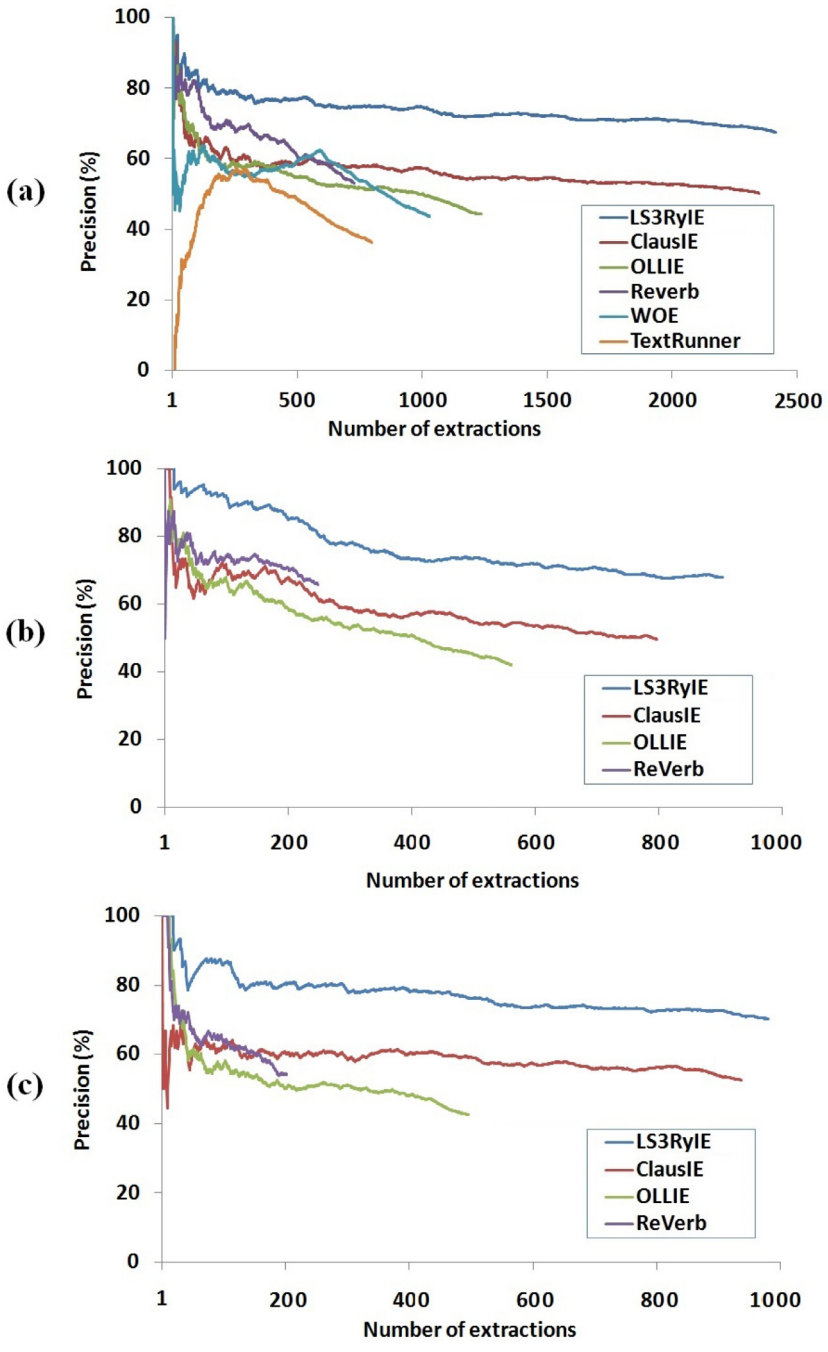

Fig. 6. Comparative results on (a) ReVerb; (b) Wikipedia; and(c) NYT datasets.

which are 12 and 10, respectively. In the process of exploring the clause structure of a sentence, the adverbials in a clause are considered in addition to the verb or verbal phrases adverbials. However, the refinement of the tree structure in LS3RyIE has led to improved performance. This helps us discover hidden relations and reduced noise in the identified relations. As seen in the table, relations r_7, r_8, r_9, r_{10}, r_{11}, and r_{12} (bold lines) are correct relations based on the results obtained from our system. These relations, which would have otherwise not been identified, have specifically been detected because of the structure refinements proposed in our approach. The limitations imposed by DP also lead to extractions by ClausIE that are not correct. As a consequence, ClausIE extracts incorrect relations in r_{13}, r_{21}, and r_{22}. The relations r_9 and r_{10} have only been identified and extracted using our approach due to the fact that the other systems rely on DP to identity the subject in the sentence. Consequently, the aforementioned systems were unable to recognize "*bombing*" as a subject. As illustrated in Table

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

Table 4
The execution time breakdown for LS3RyIE.

	Parsing time (s)	Generating clause time (s)	Generating clause with structure reformation time (s)
Short sentence	0.016	0.018	0.060
Medium sentence	0.216	0.271	0.319
Long sentence	1.003	1.054	1.857
Mean	0.411	0.447	0.745

Table 5
The employed corpora.

Corpus	Number of Documents	References
Nobel Prize:		
Nobel Prize A (1999–2005)	2296	Xu et al. (2007)
Nobel Prize B (1981–1998)	1032	Xu et al. (2010)
MUC-6:		Xu et al. (2007)
MUC-6a (training)	256	Stevenson (2007)
MUC-6b (testing)	227	Swampillai and Stevenson (2010)

3, together with ClausIE, our proposed approach was able to correctly identify the subject and henceforth extract a number of correct relations such as r_2, r_3, r_4, r_{15}, r_{16}, and r_{17}.

ReVerb returned r_{25} and r_{26} which are both incorrect; these relations have been obtained because ReVerb restricts subjects to noun phrases without prepositions and as a result incorrectly omits "... *the stabbing was unrelated to the escape plan and stemmed instead from* ..." for r_{25} and "*with his lawyers*" for r_{26}. OLLIE makes use of DP and obtains r_{23} and r_{24}, but r_{24} is incorrect because OLLIE fails to correctly identify the subject and object in the structure. WOE, meanwhile, fails to identify verbal phrases because in using DP, a non-informative connection is made between "*said*" and "*stemmed*". Hence, WOE obtains the incorrect r_{28} and r_{29} relations. TextRunner obtains incorrect relations r_{32}, r_{33}, r_{34}, and r_{35}, because it uses POS tagging and chunking for data training. Problems arise for TextRunner when faced with identifying connection words for a relation in a long sentence.

The utilization of three different datasets in our experiments is indicative of the fact that LS3RyIE is not overfitted for a specific dataset. Some of the incorrect extractions in our proposed approach are due to the incorrect tree obtained from SP and DP. There have been instances where the incorrect DP resulted from noise in the input sentences, including incorrect grammatical structures or the presence of spurious words. For instance, for the incorrect relation r_1, DP incorrectly determines "*Continuing*" to be an adverbial, which in turn leads to a connection being made with the subject "*Salim*".

7.1.3.2. System scalability. We have measured several factors in our system that can impact execution time such as the parsing process, clause generation and grammatical structure reformation when the system deals with a large number of sentences. Given the fact that the execution time of the system can depend on sentence type, we have performed our experiments on 3 different sentence types based on their structure, namely short sentences (simple), medium sentence (borderline complex) and long sentence (complex). In short sentences, the numbers of extracted patterns are in the range of 1–2 patterns. Medium sentence can produce 3–5 patterns while more than 5 patterns are extracted from long sentence. We ran our system on a desktop computer with 4 cores, 8GB RAM and 1 TB hard disk. Table 4 shows the detailed execution time of our system. It takes on average 0.447 s for the system to process sentences when clause generation is only used, while it takes 0.745 s on average when the system includes clause generation as well as grammatical structure reformation. This could indicate that LS3RyIE has no limitation to deal with large numbers of sentences. For instance, LS3RyIE is able to process one million sentences in under 7 days. However, it should be noted that in our work, the results of the extracted patterns are considered at the sentence-level and are therefore independent from the results of the other sentences. Therefore, LS3RyIE can run in parallel on different segments of an input dataset. Therefore, if our work is executed on a powerful server, which supports for many more cores than the desktop that we had access to, the execution time will be significantly reduced in LS3RyIE. For instance, execution time of LS3RyIE can be reduced by 10 folds when the system is run in parallel by ten concurrent threads. On the same note and for the same reason, due to the fact that our approach performs at a sentence level, it will have limitations in performing co-reference resolution on sentence elements such as pronouns.

7.2. Relation extraction with self-training (T2)

7.2.1. Experimental setting

For benchmarking our approach in this task, we conducted experiments on two widely used datasets, namely the Nobel Prize and MUC-6 corpora shown in Table 5. The content of the Nobel Prize corpus is comprised of reports from the New

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

York Times, BBC Online, and CNN News. The data, proposed by Xu et al. (2007), is available[1] for evaluation purposes which was extracted from the Nobel Prize website.[2] The corpus comes in two parts consisting of Nobel Prize A (1999–2005) and Nobel Prize B (1981–1998), based on the timestamp of the content as proposed by Xu et al. (2007). Nobel Prize A are records of newspapers extracted from 1981 to 1998 and Noble Prize B are records of online news extracted from 1999 to 2005. The targeted relations for the experiments are binary to quaternary relation such as ⟨Recipient, Prize, Area, Year⟩, e.g., ("Albert Einstein", "was awarded", "Nobel Prize for Physics", "1921").

The MUC-6 corpus is smaller than the Nobel Prize corpus and describes events related to '*the person who obtained a position*' and '*the person who left a position*'. The targeted relations are defined with several factors such as: (1) PersonIn: The person who is currently in a position or the person who obtains a new position; (2) PersonOut: The person who left a position; (3) Position: The position which a person has worked or the position which a person has left; (4) Organization: the company where the person has worked or has left. The gold standard of the relations in MUC-6 is available for evaluation purposes. The gold standard contains 200 documents separated into test and training sets. The training dataset (MUC-6a) consists of 256 events in an additional 100 documents; the test dataset (MUC-6b), meanwhile, presents 227 events in 100 documents. We adopt the evaluation strategy proposed in Xu et al. (2007) and use the abovementioned datasets for evaluation. It should be noted that when using the MUC-6 corpus, Xu et al. only evaluated their work based on the training dataset, which consisted of 256 events.

We separately extract patterns in two datasets with all OIE systems mentioned in Section 7.1, i.e., ReVerb, OLLIE, ClausIE and LS3RyIE. The output of these systems will be used for the bootstrapping process (BT) for identifying relations. Given the fact the approach is a self-training-based bootstrapping model, which does not require the manual specification of the initial seed set, it is worth mentioning what initial seed sets were determined in the first step of our approach for each of the two corpora. For setting seeds of the bootstrapping process, we automatically extract the patterns with the highest confidence value in each OIE systems to build the seed set and use the pronouns and compound nouns observed in these patterns for the argument seed set, together with the verbs in the patterns for the relation seed set. As a result, we obtained the following argument seed set for the Nobel Prize corpus: {*Peace, Nobel, Medicine, Literature, Laureate*} as well as the relation seed set:{*won, awarded*}, and the argument seed set of {*President, Chief, Officer*} along with the relation seed set of {*appointed, named, succeeded, retired*} for MUC-6.

In the bootstrapping process, the outcome of each iteration is updated and used in the training for subsequent iterations. The number of candidate relations and seeds should be determined for selecting in each iteration. Normally, the candidate relations will not be selected if they have a low score. A higher score for a candidate relation will show that the candidate has a higher significance. To proceed between iterations, the algorithm needs to define how many suitable candidate relations need to be added in each iteration. If the number of selections is not enough, the algorithm will stop when no more candidate relations are found. To this end, we have selected two different values for each of the two configurable parameters, namely the number of extracted patterns (#p) and the number of added seeds (#s) in our experiments as shown later in this section.

7.2.2. Experimental results

We first extract patterns from Nobel Prize and MUC corpuses with all OIE systems such as ReVerb, OLLIE, ClausIE and LS3RyIE. Table 6 summarizes the output of these systems. In the bootstrapping process, the number of iterations is set based on the number of relevant output patterns. We ran the algorithm with a number of iterations according to the number of relevant relations in each OIE systems. For instance, in case of (#p = 10, #s = 5) for Nobel Prize A, the algorithm has been ran with 130 iterations in the ReVerb+BT, 300 iterations in OLLIE+BT, 300 iterations in ClausIE+BT, and 420 iterations in LS3RyIE+BT. Experiments are applied in two cases with (#p = 5, #s = 3) and (#p = 10, #s = 5).

Tables 7–10 show the bootstrapping results on each OIE extractor on the Nobel Prize and MUC corpuses. In every case, recall increased and precision decreased until the F-measures produced significant results in a reasonable number of iterations. Regarding Nobel Prize corpus, OLLIE+BT and ClausIE+BT produced lower scores than ReVerb+BT while the best results were obtained by LS3RyIE+BT with F-measure of 67.01% (#p = 5, #s = 3) for Noble Prize A, and F-measure of 73.19% (#p = 5, #s = 3) for Noble Prize B. In terms of MUC corpus, Tables 9 and 10 indicate that LS3RyIE+BT and OLLIE+BT perform better than ReVerb+BT and ClausIE+BT. More specifically, OLLIE+BT obtained its best value with the F-measure of 67.4.3% (#p = 5, #s = 3) for MUC-6a, and its best value with the F-measure of 68.94% (#p = 5, #s = 3) for MUC-6b while LS3RyIE+BT obtained the best results with 68.23% of F-measure (#p = 5, #s = 3) for MUC-6a, and 70.38% of F-measure (#p = 5, #s = 3) for MUC-6b.

7.2.3. Comparison

We compared the performance of all OIE systems with bootstrapping (best cases) with DARE (Xu et al., 2007) as a baseline on both Nobel Prize and MUC-6 corpora. Note that we did not compare our method with (Xu et al., 2010) because the authors used a supervised learning method to build their seeds. Results on ReVerb+BT are only used to compare its bootstrapping performance with other OIE systems due to its limitation on extracting patterns. Table 11 demonstrates the

[1] http://dare.dfki.de/.
[2] http://nobelprize.org/.

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

Table 6
Extractions by OIE systems.

Corpus	# Relevant _relations	# patterns
Nobel Prize A		
ReVerb	1052	3925
OLLIE	2832	8179
ClausIE	2924	12,309
LS3RyIE	4238	14,606
Nobel Prize B		
ReVerb	478	1718
OLLIE	1365	3749
ClausIE	1345	5665
LS3RyIE	1857	6545
MUC-6a		
ReVerb	103	252
OLLIE	327	511
ClausIE	402	781
LS3RyIE	438	863
MUC-6b		
ReVerb	131	269
OLLIE	294	497
ClausIE	364	700
LS3RyIE	384	788

Table 7
Performance on Nobel Prize A.

	p = 10 and s = 5				p = 5 and s = 3			
	#Iteration	Precision(%)	Recall(%)	F-measure(%)	#Iteration	Precision(%)	Recall(%)	F-measure(%)
ReVerb	110	61.81	64.67	63.19	180	66.33	56.74	61.16
	120	60.00	68.44	63.94	200	65.60	62.35	63.93
	130	58.00	71.67	64.11	220	64.09	67.01	65.52
OLLIE	260	66.65	61.19	63.80	520	64.46	59.18	61.71
	280	65.14	64.41	64.72	560	62.85	62.15	62.50
	300	63.13	66.87	64.98	600	61.00	64.61	62.76
ClausIE	240	70.00	57.45	63.11	520	65.57	58.31	61.73
	260	68.61	61.01	64.59	560	64.53	61.79	63.13
	280	66.85	64.01	65.40	590	63.59	64.16	63.87
LS3RyIE	340	74.99	60.09	66.69	720	69.91	59.39	64.23
	360	73.11	62.10	67.19	800	67.00	63.23	65.06
	380	71.68	64.28	67.01	880	64.59	67.06	65.80

performance of the best cases from our proposed method and the best results reported in Xu et al. (2007). The DARE system obtained 59.36% and 46.31% of F-measure in Nobel Prize A and Nobel Prize B and 54.1% of F-measure in MUC-6a. The authors used two different sets of seeds for the two corpora of Nobel Prize. The results from bootstrapping with OIE systems confirm that we succeeded in improving upon Xu et al.'s work. In Nobel Prize, OLLIE+BT, ClausIE+BT and LS3RyIE+BT obtain 64.95%, 65.41% and 67.78% in F-measures for Nobel Prize A, and 67.92%, 64.58%, 73.19% in F-measures for Nobel Prize B, respectively. LS3RyIE+BT obtained better results compared with other baselines where the system improved 8.42% and 26.88%, respectively, over the Nobel Prize A and Nobel Prize B corpora in F-measures compared to the baseline.

As for the MUC-6 corpus shown in Table 12, the baseline (DARE system) succeeded in producing 54.1% of F-measure in MUC-6a. OLLIE+BT, ClausIE+BT and LS3RyIE+BT obtain 67.4%, 66.66%, 68.23% in F-measure for MUC-6a, and 68.94%, 61.92%, 70.38% in F-measure for MUC-6b, respectively. Performance wise, LS3RyIE+BT has also established its superiority to DARE with a margin of 14.13% in F-measure on the MUC-6a corpus. It should be pointed out that Xu et al. only conducted experiments on the MUC-6a with 256 events with a large number of seeds (55 seeds) and did not perform experiments or report results on MUC-6b. It is also worth noting that in DARE high precision is obtained at the cost of recall. Our system has succeeded in addressing such limitation and overcoming it.

7.2.4. Errors analysis and discussion

One of the important considerations that demand an in-depth analysis is the required number of iterations for extracting patterns. As discussed in the literature (Thelen & Riloff, 2002; Patwardhan & Riloff, 2007; Xu et al., 2010), there are no standard techniques for determining the right or exact number of iterations; therefore, in our work, we terminate the process after so many iterations until the best F-measure is obtained. Given the fact that LS3RyIE extracts a higher number of patterns compared to the other systems, on average, it requires a higher number of iterations to reach its best F-measure.

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

Table 8
Performance on Nobel Prize B.

	p = 10 and s = 5				p = 5 and s = 3			
	#Iteration	Precision(%)	Recall(%)	F-measure(%)	#Iteration	Precision(%)	Recall(%)	F-measure(%)
ReVerb	40	74.75	60.77	67.04	80	75.00	60.97	67.26
	45	70.00	64.02	66.87	90	73.55	67.27	70.27
	50	66.20	67.27	66.73	100	70.80	71.95	71.37
OLLIE	120	71.33	62.71	66.74	240	68.25	60.00	63.86
	130	69.76	66.45	68.06	260	66.54	63.37	64.91
	140	67.07	68.79	67.92	270	66.07	65.34	65.71
ClausIE	100	72.50	53.90	61.83	220	70.18	57.39	63.15
	110	70.09	57.32	63.06	240	67.58	60.29	63.73
	120	67.58	60.29	63.73	260	65.69	63.49	64.57
LS3RyIE	120	83.00	53.63	65.16	340	74.64	68.36	71.35
	140	80.85	60.69	69.51	360	73.39	71.13	72.24
	190	72.37	74.04	73.19	380	72.26	73.93	73.09

Table 9
Performance on MUC-6a.

	p = 10 and s = 5				p = 5 and s = 3			
	#Iteration	Precision(%)	Recall(%)	F-measure(%)	#Iteration	Precision(%)	Recall(%)	F-measure(%)
ReVerb	9	59.34	52.42	55.67	20	56.00	54.36	55.17
	12	53.71	63.10	58.04	25	49.60	60.19	54.39
	15	52.08	72.81	60.72	30	49.32	69.90	57.83
OLLIE	25	69.20	52.91	59.97	60	60.00	55.05	57.41
	30	66.67	61.16	63.79	70	61.43	65.74	63.51
	35	64.29	68.81	66.49	80	61.25	74.92	67.40
ClausIE	35	66.57	57.96	61.96	70	66.57	57.96	61.96
	40	64.00	63.68	63.84	80	65.75	65.42	65.58
	45	62.00	69.40	65.49	90	63.11	70.64	66.67
LS3RyIE	40	68.50	62.55	65.39	80	69.25	63.24	66.10
	45	65.77	67.57	66.67	90	66.00	67.80	66.89
	50	63.60	72.06	67.80	100	54.00	73.05	68.23

Table 10
Performance on MUC-6b.

	p = 10 and s = 5				p = 5 and s = 3			
	#Iteration	Precision(%)	Recall(%)	F-measure(%)	#Iteration	Precision(%)	Recall(%)	F-measure(%)
ReVerb	9	65.68	51.14	57.51	20	58.00	44.27	50.21
	12	56.25	48.09	51.85	25	59.20	56.49	57.81
	15	58.59	57.25	57.91	30	58.67	67.17	62.63
OLLIE	25	64.40	54.76	59.19	50	66.00	56.12	60.66
	30	61.67	62.92	62.28	60	64.67	65.98	65.31
	35	62.29	74.15	67.70	70	63.43	75.51	68.94
ClausIE	35	50.85	48.90	49.86	70	49.71	47.80	48.74
	40	53.75	59.06	56.28	80	52.00	57.14	54.45
	45	56.00	62.23	61.92	90	53.11	65.66	58.72
LS3RyIE	40	65.50	68.23	66.83	80	70.40	68.75	69.56
	45	62.22	72.92	67.15	85	68.82	73.95	70.21
	50	60.60	78.90	68.52	88	65.91	75.52	70.38

Therefore, our observation is that the higher the number of extracted patterns is, the more iterations it would take on average to reach the best F-measure.

Now from a performance perspective, despite the better performance of our method compared to DARE, we are aware of potential drawbacks that our work suffers from. While we explored the results produced by our method on a case-by-case basis, we have become aware of errors in the pattern extraction step. Specifically, some errors stemmed from incorrect parsing of the input sentences in the clause-based pattern extraction method. In certain cases, the incorrect parsing resulted from the noise in the input sentences, including erroneous grammatical forms or spurious words. Table 13 shows the total output errors (t_errors) encountered including the grammatical errors (g_errors).We report both the number of errors and the percentage of errors in the table. These errors include patterns that contain incoherent information or have

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

Table 11
Overall comparison on Noble Prize domain.

Methods	Precision (%)	Recall (%)	F-measure (%)
Nobel A			
DARE (baseline)	71.60	50.70	59.36
ReVerb+BT (#p = 5, #s = 3)	64.09	67.02	65.52
OLLIE+BT (#p = 10, #s = 5)	63.13	66.87	64.95
ClausIE+BT (#p = 10, #s = 5)	66.85	64.02	65.41
LS3RylE+BT (#p = 10, #s = 5)	71.68	64.28	67.78
Nobel B			
DARE (baseline)	83.80	32.00	46.31
ReVerb+BT (#p = 5, #s = 3)	70.80	71.95	71.37
OLLIE+BT (#p = 10, #s = 5)	67.07	68.79	67.92
ClausIE+BT (#p = 5, #s = 3)	65.69	63.49	64.58
LS3RylE+BT (#p = 10, #s = 5)	72.36	74.04	73.19

Table 12
Overall comparison on MUC-6 domain.

Methods	Precision (%)	Recall (%)	F-measure (%)
MUC-6a			
DARE (baseline)	62.00	48.00	54.10
ReVerb+BT (#p = 10, #s = 5)	52.08	72.81	60.73
OLLIE+BT (#p = 5, #s = 3)	61.25	74.92	67.40
ClausIE+BT (#p = 5, #s = 3)	63.11	70.65	66.66
LS3RylE+BT (#p = 5, #s = 3)	64.00	73.06	68.23
MUC-6b			
DARE (baseline)	n/a	n/a	n/a
ReVerb+BT (#p = 5, #s = 3)	58.67	67.18	62.63
OLLIE+BT (#p = 5, #s = 3)	63.43	75.51	68.94
ClausIE+BT (#p = 10, #s = 5)	56.00	69.23	61.92
LS3RylE+BT (#p = 5, #s = 3)	65.91	75.52	70.38

Table 13
Errors analysis.

Cases	ReVerb+BT		OLLIE+BT		ClausIE+BT		LS3RylE+BT	
	#g_errors/ #t_errors	g_rate/ t_rate(%)	#g_errors/ #t_errors	g_rate/ t_rate (%)	#g_errors/ #t_errors	g_rate/ t_rate (%)	#g_errors/ #t_errors	g_rate/ t_rate(%)
Nobel Prize A								
(#p = 10,#s = 5)	24/48	1.85/3.70	56/103	1.86/3.43	115/259	4.11/9.25	95/202	2.50/5.32
(#p = 5,#s = 3)	27/46	1.64/2.80	50/101	1.65/3.35	124/273	4.18/9.22	102/225	2.32/5.11
Nobel Prize B								
(#p = 10,#s = 5)	15/28	3.00/5.60	33/61	2.35/4.35	53/105	4.41/8.75	61/150	3.21/7.89
(#p = 5,#s = 3)	13/23	2.47/4.38	27/55	1.99/4.07	61/110	4.69/8.46	58/135	3.05/7.11
MUC-6a								
(#p = 10,#s = 5)	14/24	9.68/16.60	19/46	5.44/13.16	26/52	5.77/11.55	26/46	5.2/9.20
(#p = 5,#s = 3)	16/25	10.94/17.10	31/63	7.75/15.75	19/49	4.29/10.88	22/47	4.4/9.40
MUC-6b								
(#p = 10,#s = 5)	23/27	14.56/21.09	22/38	6.28/10.85	39/52	8.66/11.55	13/19	2.60/3.80
(#p = 5,#s = 3)	29/35	19.30/23.30	26/39	7.22/10.83	38/51	8.44/11.33	10/12	2.67/2.72

wrong grammatical structure. Most of the g_errors occur when the parser fails to correctly disambiguate a noun, verb or adjective in a sentence. For example, the verb "award" is often detected to be a noun. Errors in ClausIE and LS3RylE are higher than OLLIE due to a higher number of extracted patterns. As discussed in Section 7.1.3, the number of errors in LS3RylE is less than the errors in ClausIE primarily because of the structure reformation process. Even though the number of errors in our proposed method is not significant, we are aware of these deficiencies and their impact on the F-measure. We anticipate that the use of increasingly more robust SP and DP methods in the future will translate into improved performance for pattern extraction.

Regarding the overall performance, ReVerb+BT obtained better results than OLLIE+BT and ClausIE+BT for the Nobel Prize domain. The structures of sentences in Nobel Prize domain have a propensity to be long and complex. ReVerb extracted patterns mediated by verbs and nouns around the verbs therefore potentially missing to extract arguments with complex or long content. For the instances shown in Table 14, ReVerb+BT could recognize the relevant pattern e_1 in sentence 1 that would produce a new candidate argument seed of "*John Hume*" instead of "*John Hume*" and "*David Trible*". In OLLIE+BT, the extracted relevant pattern e_2 would produce new candidate argument seeds of "*John Hume*" and "*David Trimble*". Conse-

Table 14
Sample extracted relations based on bootstrapping on OIE systems.

1: "*The 1998 Nobel Peace Prize is awarded to John Hume and David Trimble, leaders of ...*"	
ReVerb	e_1: ("The 1998 Nobel Peace Prize", "is awarded to", "John Hume")
OLLIE	e_2: ("The 1998 Nobel Peace Prize", "is awarded to", "John Hume and David Trimble")
ClausIE	e_3: ("The 1998 Nobel Peace Prize", "is awarded", "to John Hume")
	e_4: ("The 1998 Nobel Peace Prize", "is awarded", "to John Hume and David Trimble")
LS3RyIE	e_5: ("The 1998 Nobel Peace Prize", "is awarded to", "John Hume")
	e_6: ("The 1998 Nobel Peace Prize", "is awarded to", "John Hume and David Trimble")
2: "*SDLP leader John Hume and Ulster Unionist leader David Trimble receive their joint 1998 Nobel Peace Prize in Oslo.*"	
ReVerb	e_7: ("SDLP leader John Hume and Ulster Unionist leader David Trimble", "receive", "their joint 1998 Nobel Peace Prize")
OLLIE	e_8: ("SDLP leader John Hume and Ulster Unionist leader David Trimble", "receive", "their joint 1998 Nobel Peace Prize")
	e_9: ("John Hume and Ulster Unionist leader David Trimble", "be leader of", "SDLP")
ClausIE	e_{10}: ("SDLP leader John Hume and Ulster Unionist leader David Trimble", "receive", "their joint 1998 Nobel Peace Prize")
	e_{11}: ("SDLP leader John Hume and Ulster Unionist leader David Trimble", "receive", "their joint 1998 Nobel Peace Prize in Oslo")
LS3RyIE	e_{10}: ("SDLP leader John Hume and Ulster Unionist leader David Trimble", "receive", "their joint 1998 Nobel Peace Prize")
	e_{11}: ("SDLP leader John Hume and Ulster Unionist leader David Trimble", "receive", "their joint 1998 Nobel Peace Prize in Oslo")

quently, OLLIE will use "*David Trimble*" to train a new non-relevant pattern like e_9 with a high score in the next iteration. But ReVerb will not train this non-relevant pattern. Although ReVerb extracted the smallest number of relevant patterns compared to other OIE systems, but the structure of their patterns are shorter and simpler. Extraction of e_3 and e_4 are incompletely correct extractions by ClausIE+BT but they are resolved in LS3RyIE+BT with e_5 and e_6. ClausIE and LS3RyIE do not extract pattern e_9 that helps it avoid extracting non-relevant relations. In the MUC domain, the structures of sentences have a propensity to be shorter and simpler with arguments presented as noun phrases. However, many incorrect patterns were extracted based on noun phrases by ReVerb. For instance, ReVerb extracts an incorrect pattern ("*56*", "*was named to*", "*AT&T 's board*") from the sentence "*Mr. Pelson, 56, was named to AT&T's board.*". Due to the incorrect patterns in ReVerb, bootstrapping could not identify relations very well. Similar to the results in Section 7.1, LS3RyIE provides better extractions compared toReVerb, OLLIE and ClausIE. LS3RyIE reduced the disadvantages in ClausIE and produced more relevant extractions than OLLIE, the best-performing alternative method. The results indicate that the quality of the initial seed set has a significant impact on the obtained results. However, the number of extracted patterns and the number of extracted seeds in each iteration does not seem to have a significant impact on the results.

8. Concluding remarks

We have presented a general framework for information extraction by taking advantage of clause-based patterns for information extraction. The framework mainly focuses on several major problems such as identifying hidden relations and reducing the extraction of the number of erroneous relations. Our method provides a grammatical refined structure when using English grammar clauses to identify the set of clauses. In each clause, the corresponding clause type is determined as an extractable relation according to the grammatical function of its coherent constituent. We also presented a self-training algorithm for extracting specified relations based on clause pattern extraction. Initially, we advanced an approach for extracting patterns that might contain relations. The identified relations are then used to construct a seed set. Based on the identified seeds, we proposed a self-training algorithm that extracts more relations based on the initial seed set. Based on the identified seeds, we proposed a self-training algorithm that extracts more relations based on the initial seed set. We have carried out extensive experiments for two important tasks on extracting open relations (T1) and specific relations (T2). The results reveal that our method obtained not only higher precision extractions but also had more flexible generation of relations over other state-of-the-art IE systems.

Acknowledgments

The authors acknowledge the gracious support of Natural Sciences and Engineering Research Council of Canada.

References

Abacha, A. B., & Zweigenbaum, P. (2016). MEANS: A medical question-answering systems combining NLP techniques and semantic Web technologies. In *Information processing & management: Vol. 51* (pp. 570–594).
Agichtein, E., & Gravano, L. (2000). Snowball: Extracting relation from large plain-text collections. In *Proceedings of the fifth ACM conference on digital libraries 2000 (DL 2000), San Antonio, Texas, USA, June 02-07, 2000*.
Akbik, A., Visengeriyeva, L., Herger, P., Hemsen, H., & Loser, A. (2012). Unsupervised discovery of relations and discriminative extraction patterns. In *Proceedings of the 24th international conference on computational linguistics (COLING 2012), Mumbai, India, December 2012* (pp. 17–32).
Angeli, G., Tibshirani, J., Wu, J., & Manning, C. D. (2014). Combining Distant and Partial Supervision for Relation Extraction. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP 2014), Doha, Qatar, October 25-29, 2014*.
Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007). Open information extraction from the web. In Proceedings of the 20th international joint conference on artificial intelligence (IJCAI 2007), *Hyderabad, India, 06-12 January 2007* (pp. 2670–2676).
Batista, D. S., Martins, B., & Silva, M. J. (2015). Semi-supervised bootstrapping of relationship extractors with distributional semantics. In *Proceedings of the 2016 conference on empirical methods in natural language processing (EMNLP2015), Lisbon, Portugal, 17-21 September 2015* (pp. 499–504).

Brin, S. (1998). Extracting patterns and relations from the world wide web. In *Proceedings of the international workshop on the World Wide Web and databases (WebDB98), 27-28 March 1998* (pp. 172–183).
Bunescu, R., & Mooney, R. J. (2005). A shortest path dependency Kernel for relation extraction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing (HLT/EMNLP 2005), Vancouver, British Columbia, Canada, 6-8 October 2005* (pp. 724–731).
Choi, M., & Kim, H. (2013). Social relation extraction from texts using a support vector machine-based dependency. In *Information processing & management: Vol. 49* (pp. 303–311).
Corro, L. D., & Gemulla, R. (2013). ClausIE: Clause-Based Open Information Extraction. In *Proceedings of the 22nd international conference on World Wide Web (WWW 2013), Rio de Janeiro, Brazil, 13-17 May 2013* (pp. 355–366).
Croce, D., Moschitti, A., & Basili, R. (2011). Structured lexical similarity via convolution kernels on dependency trees. In *Proceedings of the 2011 conference on empirical methods in natural language processing(EMNLP 2011), Edinburgh, UK, 27-31 July 2011* (pp. 1034–1046).
Etzioni, O., Cafarella, M., Downey, D., Popescu, A., Shaked, T., Soderland, S., et al. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence, 165*(1), 91–134.
Etzioni, O., Fader, A., Christensen, J., Soderland, S., & Mausam (2011). Open information extraction: The second generation. In *Proceedings of the 22nd international joint conference on artificial intelligence(IJCAI 2012), Barcelona, Catalonia, Spain, 16-22 July 2011* (pp. 3–10).
Fader, A., Soderland, S., & Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the 2011 conference on empirical methods in natural language processing(EMNLP 2011), Edinburgh, UK, 27-31 July 2011* (pp. 1035–1545).
Garcia, M., & Gamallo, P. (2011). Dependency-based text compression for semantic relation extraction. In *Proceedings of the workshop information extraction and knowledge acquisition (IEKA 2011), Hissar, Bulgaria, 16 September 2011*.
Greenwood, M. A., & Stevenson, M. (2006). Improving semi-supervised acquisition of relation extraction patterns. In *Proceedings of the workshop on information extraction beyond the document (IEBD 2006), Sydney, Australia, 22 July 2006* (pp. 19–35).
Gupta, S., & Manning, C. D. (2014). Spied: Stanford pattern-based information extraction and diagnostics. In *Proceedings of the ACL workshop on interactive language learning, visualization, and interfaces (ACL-ILLV 2014), Baltimore, Maryland, USA, 27 June 2014* (pp. 38–44).
Kambhatla, N. (2004). Combining lexical, syntactic and semantic features with maximum entropy models for extracting relations. In *Proceedings of the association for computational linguistics (ACL2004), Barcelona, Spain, 21-26 July 2004* (pp. 178–181).
Kok, S., & Domingos, P. (2008). Extracting semantic networks from text via relational clustering. In *Proceedings of the 2008 European conference on machine learning and knowledge discovery in databases (ECML-PKDD 200), Antwerp, Belgium, 15-19 September 2008* (pp. 624–639).
Mausam, M. S., Bart, R., & Soderland, S. (2012). Open language learning for information extraction. In *Proceedings of the 2012 conference on empirical methods in natural language processing(EMNLP 2012), Jeju Island, Korea, 12-14 July 2012* (pp. 523–534).
Marneffe, M. C., & Manning, C. D. (2008). The stanford typed dependencies representation. In *Proceedings of COLING workshop on cross-framework and cross-domain parser evaluation (CrossParser 2008), Manchester, UK, 23 August 2008* (pp. 1–8).
Mintz, M., Bills, S., Snow, R., & Jurafsky, D. (2009). Distant supervision for relation extraction without labeled data. In *Proceedings of the joint conference of the 47th annual meeting of the association on computational linguistics and the 4th international joint conference on natural language processing of the AFNLP (ACL-IJCNLP 2009), Suntec, Singapore, 2-7 August 2009* (pp. 1003–1011).
Min, B., Grishman, R., Wan, Li., Wang, C., & Gondek, D. (2013). Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies (NAACL 2013), Atlanta, Georgia, 9–14 June 2013*.
Nebot, V., & Berlanga, R. (2014). Exploiting semantic annotations for open information extraction: An experience in the biomedical domain. *Knowledge and Information Systems, 38*(2), 365–389.
Oramasa, S., Espinosa-Ankeb, L., Sordoc, M., Saggionb, H., & Serraa, H. (2016). Information extraction for knowledge base construction in the music domain. *Data & Knowledge Engineering*. http://dx.doi.org/10.1016/j.datak.2016.06.001.
Pantel, P., & Pennacchiotti, M. (2006). Espresso: Leveraging Generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st international conference on computational linguistics (COLING) and 44th annual meeting of the association for computational linguistics (ACL), Sydney, Australia, 17-18 July 2006* (pp. 113–120).
Patwardhan, S., & Riloff, E. (2007). Effective information extraction with semantic affinity patterns and relevant regions. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL 2007), Prague, June 2007* (pp. 717–727).
Quirk, R., Greenbaum, S., Leech, G., & Svartvik, J. (1985). *A comprehensive grammar of the English language*. Longman.
Ravichandran, D., & Hovy, E. H. (2002). Learning surface text patterns for a question answering system. In *Proceedings of the 40th annual meeting of the association for computational linguistics (ACL 2002), Philadelphia, July 2002* (pp. 41–47).
Riedel, S., Yao, L., & McCallum, A. (2010). Modeling relations and their mentions without labeled text. In *Proceedings of European conference on machine learning and principles and practice of knowledge discovery in databases (ECML-PKDD 2010), Barcelona, Spain, 20–24 September 2010*.
Riedel, S., Yao, L., McCallum, A., & Marlin, M. (2013). Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies (NAACL 2013), Atlanta, Georgia, 9-14 June 2013* (pp. 74–84).
Rosenfeld, B., & Feldman, R. (2007). Clustering for unsupervised relation identification. In *Proceedings of the sixteenth ACM conference on conference on information and knowledge management (CIKM2007), Lisbon, Portugal, 6-10 November 2007* (pp. 411–418).
Ryu, P. M., Jang, M. G., & Kim, H. K. (2015). Open domain question answering using Wikipedia-based knowledge model. *Information Processing & Management, 50*, 683–692.
Santos, C. D., Xiang, B., & Zhou, B. (2015). Classifying relations by ranking with convolutional neural networks. In *Proceedings of 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (ACL-IJCNLP 2015), Beijing, China, 26-31 July 2015* (pp. 626–634).
Singhal, A., Simmons, M., & Lu, Z. (2016). Text mining for precision medicine: Automating disease-mutation relationship extraction from biomedical literature. *Journal of the American Medical Informatics Association*. http://dx.doi.org/10.1093/jamia/ocw041.
Socher, R., Huval, B., Manning, C. D., & Ng, A. Y. (2012). Semantic compositionality through recursive matrix vector spaces. In *Proceedings of the 2012 conference on empirical methods in natural language processing(EMNLP 2012), Jeju Island, Korea, 12-14 July 2012* (pp. 1201–1211).
Stevenson, M. (2007). Fact distribution in information extraction. *Language Resources and Evaluation, 40*, 183–201.
Suchanek, F., Kasneci, G., & Weikum, G. (2007). YAGO: A core of semantic knowledge unifying WordNet and Wikipedia. In *Proceedings of the 16th international conference on World Wide Web (WWW 2007), Banff, Alberta, Canada, 8-12 May 2007* (pp. 697–706).
Sudo, K., Sekine, S., & Grishman, R. (2003). An Improved extraction pattern representation model for automatic IE pattern acquisition. In *Proceedings of 41st annual meeting of the association for computational linguistics (ACL 2003), Sapporo, Japan, 7-12 July 2003* (pp. 224–236).
Surdeanu, M., Tibshirani, J., Nallapati, R., & Manning, C. D. (2012). Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL 2012), Jeju Island, Korea, 12-14 July 2012* (pp. 455–465).
Swampillai, K., & Stevenson, M. (2010). Inter-sentential relations in information extraction corpora. In *Proceedings of the seventh international conference on language resources and evaluation (LREC 2010), Malta, 17-23 May 2010*.
Takamatsu, S., Sato, I., & Nakagawa, H. (2011). Probabilistic matrix factorization leveraging contexts for unsupervised relation discovery. In *Proceedings of the 15th Pacific-Asia conference on advances in knowledge discovery and data mining (PAKDD 2011), Schezhen, China, 24–27 May 2011*.
Takamatsu, S., Sato, I., & Nakagawa, H. (2012). Reducing wrong labels in distant supervision for relation extraction. In *Proceedings of the 50the annual meeting of the association for computational linguistics (ACL 2012), Jeju, Korea, 8–14 July 2012*.

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009

Thelen, M., & Riloff, E. (2002). A bootstrapping method for learning semantic lexicons using extraction pattern contexts. In *Proceedings of the 2002 conference on empirical methods in natural language processing(EMNLP 2002), Stroudsburg, PA, USA, 2002* (pp. 214–221).
Thenmozhi, D., & Aravindan, C. (2015). An automatic and clause-based approach to learn relations for ontologies. *The Computer Journal*. doi:10.1093/comjnl/bxv071.
Turney, P. D. (2008). The latent relation mapping engine: Algorithm and experiments. *Journal of Artificial Intelligence Research, 33*, 615–655.
Vlachidis, A., & Tudhope, D. (2016). A knowledge-based approach to Information Extraction for semantic interoperability in the archaeology domain. *Journal of the Association for Information Science and Technology, 67*, 1138–1152.
Vo, D. T., & Bagheri, E. (2015). Syntactic and Semantic Structures for Relation Extraction. In *Proceedings of the 6th Symposium on Future Directions in Information Access (FDIA 2015), Thessaloniki, Greece, September 2015* (pp. 28–33).
Vo, D.T., & Bagheri, E. (2016). Open Information Extraction. arXiv:1607.02784.
Wu, F., & Weld, D. S. (2010). Open information extraction using Wikipedia. In *Proceedings of the 48th annual meeting of the association for computational linguistics (ACL 2010), Uppsala, Sweden, 11-16 July 2010* (pp. 118–127).
Xu, F., Uszkoreit, H., & Li, H. (2007). A seed-driven bottom-up machine learning framework for extracting relations of various complexity. In *Proceedings of the 45th annual meeting of the association for computational linguistics (ACL 2007), Prague, Czech, June 2007* (pp. 584–591).
Xu, F., Uszkoreit, H., Krause, S., & Hong Li, H. (2010). Boosting relation extraction with limited closed-world knowledge. In *Proceedings of the 23rd international conference on computational linguistics (COLING 2010), Beijing, China, 23–27 August 2010* (pp. 1354–1362).
Xu, Y., Kim, M. Y., Quinn, K., Goebel, R., & Barbosa, D. (2013). Open information extraction with tree Kernels. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies (NAACL-HLT 2013), Atlanta, Georgia, 9-14 June 2013* (pp. 868–877).
Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., & Jin, Z. (2015). Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP 2015), Lisbon, Portugal, 17-21 September 2015* (pp. 1785–1794).
Xu, W., & Zhang, C. (2014). Trigger word mining for relation extraction based on activation fore. *International Journal of Communication Systems, 27*, 2134–2146.
Yahya, M., Whang, S. E., Gupta, R., & Halevy (2014). ReNoun: Fact extraction for nominal attributes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP 2014), Doha, Qatar, 25-29 October 2014* (pp. 325–335).
Yangarber, R., Grishman, R., Tapanainen, P., & Huttunen, S. (2000). Automatic acquisition of domain knowledge for information extraction. In *Proceedings of the 18th conference on computational linguistics (COLING 2000), Saarbrucken, Germany, 31 July - 04 August 2000* (pp. 940–946).
Yao, L., Riedel, S., & McCallum, A. (2012). Unsupervised relation discovery with sense disambiguation. In Proceedings of the 50th annual meeting of the association for computational linguistics (ACL 2012), *Jeju Island, Korea, 8-14 July 2012* (pp. 712–720).
Yao, L., Haghighi, A., Riedel, S., & McCallum, A. (2011). Structured relation discovery using generative models. In *Proceedings of the 2011 conference on empirical methods in natural language processing (EMNLP 2011), Edinburgh, Scotland, UK, 27-31 July 2011* (pp. 1456–1466).
Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014). Relation classification via convolutional deep neural network. In *Proceedings of the 25th international conference on computational linguistics (COLING 2014), Dublin, Ireland, 23-29 August 2014* (pp. 23–29).
Zeng, D., Liu, K., Chen, Y., & Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 conference on empirical methods in natural language processing (EMNLP 2015), Lisbon, Portugal, 17-21 September 2015* (pp. 1753–1762).
Zhang, C., Xu, X., Ma, Z., Gao, S., Li, Q., & Guo, J. (2015). Construction of semantic bootstrapping models for relation extraction. *Knowledge-Based Systems, 83*. doi:10.1016/j.knosys.2015.03.017.
Zhang, C., Zhang, Y., Xu, W., Ma, Z., Leng, Y., & Guo, J. (2015). Mining activation force defined dependency patterns for relation extraction. *Knowledge-Based System, 83*, 128–137.
Zhou, G., & Zhang, M. (2007). Extracting Relation information from text documents by exploring various types of knowledge. *Information Processing & Management, 43*, 969–982.
Zhou, G., Qian, L., & Fan, J. (2010). Tree Kernel based semantic relation extraction with rich syntactic and semantic information. *Information Sciences, 180*, 1313–1325.

Please cite this article as: D.-T. Vo, E. Bagheri, Self-training on refined clause patterns for relation extraction, Information Processing and Management (2017), http://dx.doi.org/10.1016/j.ipm.2017.02.009