Journal of King Saud University – Computer and Information Sciences (2015) 27, 13–24

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Concept relation extraction using Naïve Bayes classifier for ontology-based question answering systems

G. Suresh kumar [a,*], G. Zayaraz [b]

[a] Department of Computer Science, Pondicherry University, India
[b] Department of Computer Science and Engineering, Pondicherry Engineering College, India

Received 10 July 2013; revised 18 November 2013; accepted 13 March 2014
Available online 9 May 2014

KEYWORDS

Relation extraction;
Ontology development;
Dependency parsing;
Question answering system

Abstract Domain ontology is used as a reliable source of knowledge in information retrieval systems such as question answering systems. Automatic ontology construction is possible by extracting concept relations from unstructured large-scale text. In this paper, we propose a methodology to extract concept relations from unstructured text using a syntactic and semantic probability-based Naïve Bayes classifier. We propose an algorithm to iteratively extract a list of attributes and associations for the given seed concept from which the rough schema is conceptualized. A set of hand-coded dependency parsing pattern rules and a binary decision tree-based rule engine were developed for this purpose. This ontology construction process is initiated through a question answering process. For each new query submitted, the required concept is dynamically constructed, and ontology is updated. The proposed relation extraction method was evaluated using benchmark data sets. The performance of the constructed ontology was evaluated using gold standard evaluation and compared with similar well-performing methods. The experimental results reveal that the proposed approach can be used to effectively construct a generic domain ontology with higher accuracy. Furthermore, the ontology construction method was integrated into the question answering framework, which was evaluated using the entailment method.

1. Introduction

Question answering (QA) systems are considered more complex than information retrieval (IR) systems and require extensive natural language processing techniques to provide an accurate answer to the natural language questions. Question answering systems in general use external knowledge sources to extract answers. Domain-specific question answering systems require pre-constructed knowledge sources, such as a domain ontology. A major challenge in knowledge-based QA

* Corresponding author. Tel.: +91 9677528467.
E-mail addresses: mgsureshkumar@gmail.com (G. Suresh kumar), gzayaraz@pec.edu (G. Zayaraz).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

1319-1578
http://dx.doi.org/10.1016/j.jksuci.2014.03.001

system development is building a huge knowledge base with objective and correct factual knowledge in the preferred domain. The process of collecting useful knowledge from various sources and maintaining this information in a knowledge repository is a useful process when providing a required answer on demand with greater accuracy and efficiency. The domain ontology is considered a set of representational primitives used to model a knowledge domain. Ontology knowledge can be easily translated into first-order-logic representations for use with the semantic web (Horrocks, 2008). An ontology can provide extensive vocabularies of terms, each with a well-defined meaning and relationships with other terms; they are essential components in many knowledge-based applications (Miller, 1995). Ontologies have had a great impact on several fields, e.g., biology and medicine. Most domain ontology constructions are not performed automatically (Gacitua et al., 2008). Most of the work on ontology-driven QAs tend to focus on the use of ontology for query expansion (Mc Guinness, 2004). However, domain ontology is considered a rich source of knowledge (Ferrnandez et al., 2009) and is used to improve the efficiency of QAs.

Manually constructing an ontology with the help of tools is still practiced to acquire knowledge of many domains. However, this is a difficult and time-consuming task that involves domain experts and knowledge engineers (Navigli et al., 2003). The potential size, complexity and dynamicity of a specific domain increase the difficulty of the manual ontology construction process. The solution to this problem is the use of ontology learning techniques with knowledge-rich web resources. Many efforts have been undertaken in the last decade to automate the ontology acquisition process. However, there are many restrictions in terms of building ontologies that accurately express the domain knowledge and information required for a question answering system. Many supervised learning methods proposed for automatic ontology construction are deficient in the handling of large-scale data. Here, the "scale" represents a characterization of the algorithm (small, medium and large) with respect to its performance and adaptability as follows. Algorithms with high complexity are classified as small scale. The algorithms that deliver moderate performance were considered in the medium-scale category. The unsupervised algorithms that are scalable incrementally adapt and that work with voluminous data are considered to be in the large scale category.

The proposed methodology addresses the problem of how to construct the domain ontology from an empty ontology and keep updated for further question answering processes. The novelty of this approach relies on the combinations of mature NLP technologies, such as the semantic similarity-based attribute association identification for relational ontology conceptualization using a Naïve Bayes classifier with widely accepted large-scale web resources. In this proposed approach, the attributes and associations of the given seed concept are automatically extracted using a set of hand-coded rules devised from the dependency parsing pattern of relevant sentences. We introduced an extension to the Naïve Bayes classifier to learn concept relations from the extracted associations. Then, the predicted concept relations are used to model the domain concepts of the resulting ontology. Furthermore, we proposed an experimental framework for the concept-relational ontology-based question answering process.

The QA framework proposed in this paper includes two subsystems: (1) a dynamic concept relational (CR) ontology construction module capable of extracting new concepts from the web and incorporating the extracted knowledge into the CR Ontology knowledge base, and (2) an answer extraction module that formulates the query string from the natural language question according to the expected answer and retrieves the information from the ontology for answer formation. An experimental setup was established to test the performance of our proposed relation extraction approach using a benchmark data-set (Voorhees, 1999). The obtained result was compared with the performance of similar well-performing relation extraction and ontology construction methods. The proposed question answering approach was tested using a benchmark data set as well as using an entailment-based evaluation method. The QA performance improvement was proven by comparing the results with another similar QA system.

We established the following hypotheses to test the performance of the proposed methods for relation extraction and question answering:

H1. There will be an improvement in relation extraction accuracy by using our proposed hand-coded rules formulated from dependency-parsing sentence patterns and a binary decision tree-based rule engine.

H2. There will be a considerable improvement in the accuracy of the concept relation learning for automatic ontology construction using our proposed expectation maximization-based Naïve Bayes classifier with syntactic and semantic probabilities.

H3. There will be a considerable performance improvement in ontology-based open domain question answering using our proposed question answering framework.

The rest of this paper is organized as follows. The related work is summarized in Section 2. The proposed concept relation extraction method using hand-coded rules formulated from dependency-parsing sentence patterns and the binary decision tree-based rule engine are elaborated in Section 3. Section 4 depicts the design of an expectation maximization-based Naïve Bayes classifier using syntactic and semantic probabilities, followed by the proposed concept relational ontology-based question answering framework in Section 5. The evaluation method and experimental setup are elaborated in Section 6, after which, the results and discussion are presented in Section 7, followed by the conclusion and references.

2. Related work

2.1. Question answering systems

We investigated a number of novel techniques that perform open-domain question answering. The investigated techniques consist of document retrieval for question answering, domain ontology-based question answering systems, web-based semantic question answering systems, and answer extraction via automatically acquired surface matching text patterns for question answering.

Automatic QA systems, such as AnswerBus (Zhang et al., 2005) and MULDER (Kwok et al., 2001), extend their data resource from the local database to the web resources, which also extend the scope of the questions they can handle. In 1999, TREC set the first QA track (Voorhees, 1999). AquaLog (Lopez et al., 2007) is an ontology-based question answering system that processes input queries and classifies them into 23 categories. If the input question is classified into one of these categories, the system will process it correctly. There are a few question answering systems based on conditional knowledge structures, which were introduced by Areanu and Colhon (2009). In these systems, a conditional schema is used to generate XML-based conditional knowledge structure, which is used for question answering. Ferrnandez et al. (2009)proposed an ontology-based question answering system called QACID to answer natural language queries related to the cinema domain. This system extracts answers from a pre-constructed ontology by comparing question attributes with ontology attributes. QACID was evaluated using entailment queries composed for the cinema domain. The overall official *F*1-accuracy reported by QACID is 93.2% with an ABI threshold of 0.5.

2.2. Automatic ontology construction

Ontology learning is a knowledge acquisition activity that relies on automatic methods to transform unstructured data sources into conceptual structures. The first proposals for ontology learning (Maedche, 2002) built all resources from scratch, but the manner of the tackling ontology population has evolved due to the existence of complementary resources, such as top-level ontologies or semantic role repositories. Some ontology learning approaches, such as TERMINAE (Aussenac-Gilles et al., 2008), provide conceptualization guidance from natural language text integrating functions for linguistic analysis and conceptual modeling.

A number of methods have already been proposed for automatically constructing an ontology from text. Graph-based approaches are very popular for representing concept relations (Hou et al., 2011). There are some approaches using mixed methodologies, such as using relational databases and semantic graphs (Ra et al., 2012). Some ontology development tools have been proposed to extract deep semantic relation between concepts using mapping functions and to generate rough schema. OntoCmaps (Zouaq et al., 2011) is an ontology development tool that extracts deep semantic relations from text in a domain-independent manner. Mining the situation context from text and constructing a situation ontology is an interesting area in information retrieval. Jung et al. (2010) have performed notable work in this area. There were a few studies that utilized lexico-syntactic patterns and lexico-semantic probabilities for automatically extracting concept relationships (Hearst, 1992, 1998) from raw text.

2.3. Semantic relation extraction

The mining of concept relation semantics is a sophisticated technique for automatic conceptualization of ontology concepts and instances. Most machine-learning approaches used to automatically construct an ontology are deficient because of the need for annotated data. Even though this annotation is possible by using hand-coded rules, it requires a high level of processing time. Unsupervised methods, which can learn from un-annotated raw text, are considered superior alternative. Yangarber and Grishman, 2001proposed a method for relation extraction from large-scale text. They used pairs of co-occurring entities available in target documents for extracting relevant patterns of the given seed concept. However, the unsupervised methods are lacking in providing the required relation extraction accuracy. The importance of measuring semantic similarity between related concepts has been well-explored by many researchers (Said Hamani et al., 2014), and its effectiveness has been demonstrated in many natural language applications (Sarwar Bajwa et al., 2012).

Most methods of relation extraction start with some linguistic analysis steps, such as full parsing, to extract relations directly from the sentences. These approaches require a lexicalized grammar or link grammars. Information extraction tools such as GATE (developed from earlier TIPSTER architecture) NLP tools use a set of hand-coded rules to extract relations from text (Cowie and Wilks, 2000). There are few open IE (information extraction) systems proposed to extract relation axioms from large web documents (Banko et al., 2007; Wu and Weld, 2010; Zhu et al., 2009). The open IE systems have been used to learn user interests (Ritter et al., 2010), acquire common sense knowledge (Lin et al., 2010), and recognize entailment (Schoenmackers et al., 2010; Berant et al., 2011).

Open IE systems such as TEXTRUNNER (Banko et al., 2007), KNOWITALL (Etzioni et al., 2005), REVERB (Etzioni et al., 2005), WOEpos, and WOEparse (Wu and Weld, 2010), extract binary relations from text for automatic ontology construction. The Snowball (Agichtein and Gravano, 2000) system extracts binary relations from document collections that contain user specified tuples, which are used as sample patterns to extract more tuples. KNOWITALL automatically extracts relations by using a set of domain-independent extraction patterns to learn labeled data. REVERB uses the syntactic and lexical constraints to identify relation phrases and extracts pairs of arguments for each relation phrase. Then, a logistic regression classifier is used to assign a confidence score. Furthermore, we compared our method with an early relation extraction method originally proposed by Brin (1998)called DIPRE (Dual Iterative Pattern Relation Extraction). The overall *F*1-accuracy reported for the open-IE systems such as DIPRE, Snowball_VS, TextRunner, WOEparse, and REVERB is 67.94, 89.43, 50.0, 58.3, and 60.9, respectively.

Carlson et al. (2010) proposed a coupled semi-supervised learning method for information extraction. The goal of the method is to extract new instances of concept categories and relations using an initial ontology containing dozens of pre-constructed categories and relations. The method exploits the relationship among categories and relations through coupled semi-supervised learning. Fifteen seed concepts were used to extract relations from 200 million web pages. The average precision reported was 95%. Recently, Krishnamurthy and Mitchell (2013) proposed a component called ConceptResolver for the Never-Ending Language Learner (NELL) (Carlson et al., 2010) that learns relations from noun phrase pairs. ConceptResolver performs both word sense induction and synonym resolution on the extracted relations. The experimental evaluation was conducted using gold standard clustering data. When ConceptResolver was used to learn real-world concept

for use with NELL's knowledge base, it demonstrated an overall accuracy of 87%.

From the investigated related works, it is evident that the existing pattern-based relation extraction methods are deficient in handling large-scale data. On the other hand, the proposed supervised learning methods are deficient in providing the required level of accuracy. Most of the relation extraction system require pre-constructed labeled (Carlson et al., 2010) data for learning. The relation extraction method proposed in this paper addresses these issues.

3. Concept relation extraction for automatic ontology construction

The proposed method to automatically construct domain ontology concepts extracts the domain attributes and associations from a set of relevant documents. We used the Stanford dependency parser (Marie-Catherine de Marneffe, 2008) for generating a parse tree for each individual sentence in relevant documents concerning the seed concept. Then, the proposed binary decision tree-based rule engine applied the set of hand-coded rules to the dependency parsing pattern. The outcome of the rule engine is a set of triples consisting of three components: candidate key word, which represents the given seed concept; predicate and target object, which is considered the associated concept. The triple set is used to extract feature data for training the proposed expectation–maximization-based Naïve Bayes classifier, which predicts whether there exists a relation between the seed concept and associated concept through the predicate. Then, the ontology concept schema is generated for the relevant relations. In this paper, the concept relation extraction process is modeled as an unsupervised classification problem using an expectation–maximization-based Naïve Bayes classifier that makes use of lexico-syntactic and lexico-semantic probabilities calculated using the WordNet similarity between the seed concept and associated concept. The overall process sequence is depicted in Fig. 1.

3.1. Hand-coded rules for concept triple extraction

The concept triple extraction from the dependency parsing pattern is performed using hand-coded rules. The rules are formulated by empirical analysis.

Definition 1. Attribute(s) of a concept x is/are defined as the predicate(s) p *is a subset of* P {P: *set of all predicates*} used to define another concept y, where x is determined by y.

Definition 2. Association(s) between concepts x and y is/are defined as the relationship r *is a subset in* R {R: *relation set*} between them. The concepts x and y are said to be associated with each other if there exists a relation r between x and y such that r is a predicate and x and y belong to the superset of x and y.

Definition 3. When a concept x is succeeded by a verb phrase VP that is a subset in $VPList$, which is further succeeded by a $\{NP|S|PP|ADJP|ADVP\}$, then the object y in the NP is an attribute of x.

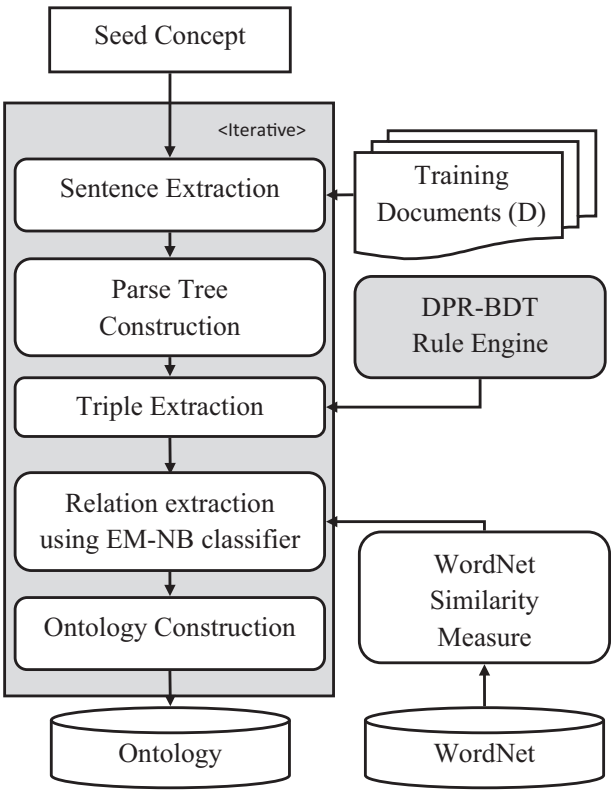

Figure 1 Ontology construction using concept relations.

The hand-coded rules framed using the Definitions 1–3 are shown in Table 1. The pattern-matching engine considers the presence of the "Attribute lexical-pattern" to identify the predicate used along with the connectors as one of the attributes of the concept C. The nearest VP node to target object is (right most NP) only considered for identifying attributes except the case "if" verb pattern is $< TO + VB > OR < VBS >$, "then" convert VB into VBZ and attach BY to it ($VBZ + BY$) to construct the attribute (e.g., "to create" as "created by").

Table 1 Hand coded dependency parsing pattern rules.

Rule No.	Rule components		Example attributes
	Attribute Lex-pattern	RHS of parent VP	
1	VBZ	NP, S	"is"
2	VBZ + DT	NP, S	"is a"
3	VBD	NP, S	"lived"
4	VBZ + IN	PP	"lives in"
5	VBD + IN	PP	"lived in"
6	VBG + IN	PP	"living in"
7	VBN + TO	PP	"used to"
8	VBN + IN	PP	"used for"
9	VB + RP	NP	"carry out"
10	VBP	ADJP	"are"
11	VBP	NP	"are"
12	VBP	ADVP	"drive west"

The following three basic pattern components are used to generalize the rule:

$A = [NP, \{PPER|NN|PRF\}]$
$B = [VP \text{ closest to } C] \ \& \ [\text{right child of } VP\{S|NP|PP|ADJP|ADVP\}]$
$C = [NP, \{PPER|NN\}]$
Precedence: $A < B < C$

The mapping $C(A, B)$ denotes that the B is one of the attributes that describes the concept A with the value C.

The general format of the rule is as follows:

$\{NP\ (concept)\} * \{VP\ (Rule: 1–12)\} * \{PP\ (Rule: 1–12)\}\ \{NP\ (object)\}$.

The rules are very specific to the dependency parsing patterns generated by Standford parser. Extracting concept triples

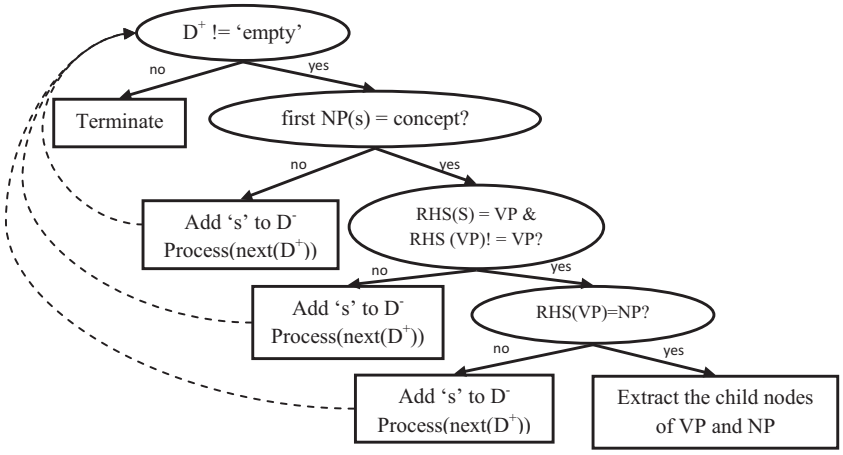

Figure 2 Recursive binary decision tree for concept triple extraction.

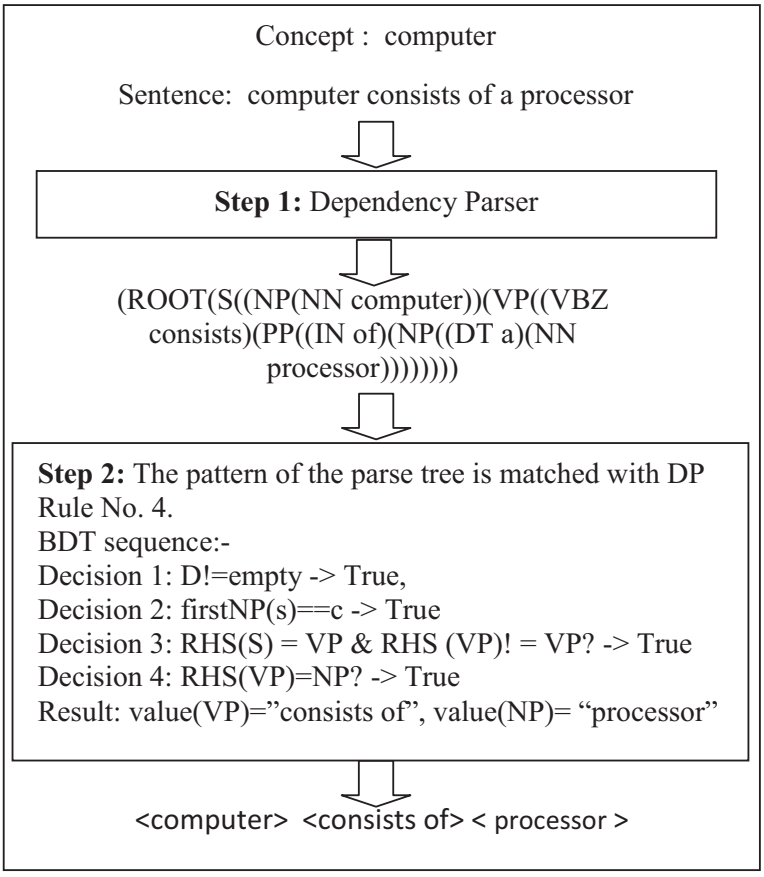

Figure 3 An example of concept triple extraction.

from the relevant sentences is a basic pattern matching process. Thus, the rules are treated in the rule engine as If-then Normal Form (INF) rules. A binary decision tree-based rule engine was designed for this purpose.

3.2. Recursive BDT-based rule engine

Decision rules and decision trees are key techniques in data mining and knowledge discovery in databases (Takagi, 2006; Breiman et al., 1984). The proposed binary decision tree (BDT)-based rule engine is used to extract the three components of a relation from which the attribute and associations are predicted. The training sample set D is a collection of text documents that consists of the dependency parsing patterns for the corresponding sentences of the seed concepts. We used the subclass method proposed by Takagi (2006) to separate the negative only samples from the training set, so that the concept triple could be precisely extracted from the remaining positive sample. Fig. 2 depicts the BDT rule engine designed to extract the triple; subject, predicate and object. Each decision node generates only a negative sample set $D^- = \{y1, y2,\ldots yp\}$ when the decision result is 'false' and otherwise generates a subclass $D^+ = \{x1, x2,\ldots xp\}$ consisting of the remaining samples. For each false decision, a new BDT is constructed recursively until the sample set D becomes empty or reaches the goal decision. On reaching the goal decision, the subclass sample set D^+ will have only a single positive sample from which the resultant components are extracted. Then, the ontology concept schema is generated using a classifier designed to extract concept relations from the set of triples.

Fig. 3 shows an example of extracting a relation triple using our proposed hand-coded rules. When the seed concept (c) is "computer", and the sentence in the training sample instance is "computer consists of a processor", the parser generates an equivalent parse tree of the sentence (s). The parsing pattern is expected to match with any one of the 12 rules listed in Table 1. This rule matching process is automatically performed by the BDT rule engine by checking for the four decisions of the parse pattern. A concept triple is successfully extracted when all four decisions are TRUE. Otherwise, the sample instance is considered as a negative sample. In our example, the parse tree pattern structure matches with rule 4 in Table 1. Hence, the relation "consists of" and a related concept "processor" is successfully extracted for the given seed concept "computer."

4. Automatic relation classification using a Naïve Bayes classifier

Naïve Bayes (NB) classifiers have been proven to be very effective for solving large-scale text categorization problems with high accuracy. In this research, we used an expectation–maximization-based Naïve Bayes classifier for classifying the relation between the seed concept and predicate object through

Algorithm The ontology attributes classification
Input: A labeled training set D_l ; an unlabeled test set D_u ; A set T of triples
Output: converged $c^* = argmax_{c_l} P_{D_u}(c_l|t_i)$
1: Initialise $P_{D_u}(c_l), P_{D_u}(t_i|c_l), and\ P_{D_u}(t_i)$ using initial Naïve bayes classifier
2: **Repeat** {
3: t ← 1
4: **for each** $c \in c_l$ and $t_i \in D_u$ **do**
5: Calculate $NP_{rank} = \sum_{i=1}^{N} sim_{W\&P}, sim_{Li}, sim_{lch}\ (NP, NP_i)$
6: Calculate
$$P_{D_u}^{(t)}(LSemPt_{i,k}|c_l) = \begin{Bmatrix} 1 \text{ if } P_{D_u}^{(t)}(LSP_{t_i}|c_l)\ and\ NP_{rank} > h \\ 0, \quad \text{otherwise} \end{Bmatrix}$$
7: Calculate
$P_{D_u}^{(t)}(t_l|c_l)$ based on $P_{D_u}^{(t)}(LSP_{t_i}|c_l)$ and $P_{D_u}^{(t)}(LSemPt_{i,k}|c_l)$
8: Calculate $P_{D_u}^{(t)}(c|t_i)$ based on $P_{D_u}^{(t-1)}(c_l), P_{D_u}^{(t-1)}(t_i|c_l), and\ P_{D_u}^{(t-1)}(t_i)$
9: **end for**
7: $c^* = argmax_{c_l} P(c_l|t_i) = argmax_{c_l} \frac{P(c_l)P(t_i|c_l)}{P(t_i)}$
8: **Until** c^* **converged**

Figure 4 The EM-based Naïve Bayes classifier for attribute identification.

the predicate that exits in a sentence. Thus, the sentence classification problem is converted to a concept relation classification problem. The proposed classifier model is depicted in Fig. 4.

4.1. Naïve Bayes classifiers for concept relation classification

Several extensions to Naïve Bayes classifiers have been proposed (Nigam et al., 2000), including combining expectation–maximization (EM) (Dempster et al., 1977) and Naïve Bayes classifiers for learning from both labeled and unlabelled documents in a semi-supervised algorithm. The EM algorithm is used to maximize the likelihood with both labeled and unlabeled data. Liu et al. (2002) proposed a heuristic approach, Spy-EM, that can learn how to handle training and test data with non-overlapping class labels.

We extended the basic Naïve Bayes classifier model for concept relation classification in which the concept relation identification problem is posed as a self-supervised learning problem. The attribute a_i of the given concept c is described by the triple t, which consists of concept pair connected through a predicate. The attribute (predicate) a_i *is a subset in* A, where A is the attribute set of the concept c, and triple t_i *is a subset of* T, where T is the set of triples for all of the attributes. The proposed classifier is used to categorize the attribute candidate triples into two classes: relation class c_1 or non-relation class c_0. Thus, t_i is either classified into c_1 or c_0 depending on feature probabilities. A triple instance t_i will be considered for ontology construction only when it is classified as a relation class c_1. We used the lexico-syntactic probability of triples and lexico-semantic probability of the triples as features to compute the classification probability $P(t_i|c_l)$, where l is the label *0* or *1*. In the trained Naïve Bayes classifier model, the target class c^* of the triple t_i is computed as shown in Eq. (1).

$$c^* = \arg\max_{cl} P(c_l|t_i) = \arg\max_{cl} \frac{P(c_l) \cdot P(t_i|c_l)}{P(t_i)} \tag{1}$$

where $P(c_l)$ is the target label probability; $P(t_i)$ is the probability of the training sample initialized by the classifier, and $P(t_i|c_l)$ is computed probability of assigning the class label (*1* or *0*) to the triple t_i. We applied the lexico-syntactic probability $LSPt_i$ and lexico-semantic probability $LSemPt_i$; thus, $P(t_i|c_l)$ is rewritten as shown in Eq. (2).

$$P(t_l|c_i) = P(LSP_{t_i}|c_l) \cdot \sum_{k=1}^{|t|} P(LSemPt_{i,k}|c_l) \tag{2}$$

where $P(LSPt_i|c_l)$ and $P(LSemPt_{i,k}|c_l)$ can be learned from the annotated triple for the target attribute class. The initial training data D^+ and D^- are generated from the triples extracted by the BDT rule engine and annotated using WordNet similarity measures. We empirically fixed a threshold value for similarity score, using the class labels assigned to the initial training set. The expectation–maximization procedure is used with the Naïve Bayes classifier to optimize the classifier in the estimation of the probability for unlabeled new data-sets. The parameters proposed to train in the EM procedure are prior probability $P(c_l)$, lexico-syntactic probability $P(LSPt_i|c_l)$, and lexico-semantic probability $P(LSemPt_{i,k}|c_l)$. We used the Laplacian smoothing method to adjust the parameters of training data. The Naïve Bayes classifier is bootstrapped using EM procedure.

4.1.1. Lexico-syntactic probability

The structural similarities of a sentence can be used as features for extracting useful knowledge from the sentence (Kang and Myaeng, 2005). Our sentence pattern shown in Eq. (3) is expressed as a triple consisting of concept noun (N), an attribute describing the concept, composed using the functional verb combined with any connectives ($VP|DT$), and a text segment with one or more nearest nouns (NN). The missing elements in the source sentence are indicated using NULL values.

$$SP_f(x, y, z) = \{N, Attr(VP|DT), NN\} \tag{3}$$

The following features are computed using the above sentence pattern: presence of concept noun, presence of functional verb, structural similarity between the nouns, and semantic similarities between nouns. The presence of a concept noun and functional verb is indicated by the value *1*. Otherwise, the value is *0*. In addition, another feature, sentence weight score, is calculated from the original source sentence s. The three components such as concept (a), predicate (b), and target object (c) presence in the triple extracted for each sentence are considered as feature parameters. The sentence weight is the sum of weights calculated from the list of weight values assigned to various sentence features given in the Table 2. A particular feature value is selected based on the arrangement of triple components in the original sentence. The sentence weight (SW) score is calculated from the following Eq. (4) using dependency parsing pattern generated by the Stanford parser:

$$\mathrm{SW}_{\mathrm{score}} = \sum_{i=1}^{N} w_i * f_i \tag{4}$$

where, f_i is the feature, and w_i is the corresponding weight. The value of f_i as *1* or *0* depends on the presence or absence of the particular feature in the sentence. Some of the features are mutually exclusive. The lexico-syntactic probability, LSP, is considered as '1' when the $\mathrm{SW}_{\mathrm{score}}$ is greater than or equal to '1'.

The feature weight values are calculated using a confidence scoring function that adjusts the weight based on the keyword based information gain. We used 1000 manually annotated sentences extracted from TREC 2008 documents. The weight values were empirically tuned to achieve optimum precision and recall values. For example, the lexico-syntactic probability for the concept triple < computer, consists of, processor > extracted from the sentence "computer consists of a processor" is calculated by adding the weight values of the sentence pattern features 1, 2, 3, and 4. Hence, the calculated

Table 2 Sentence pattern features.

Feature No.	Feature	Weight
1	a, b and c cover all the words in source sentence	1.32
2	Sentence starts with 'a'	0.58
3	'b' is the immediate successor of 'a'	0.42
4	'a' is a proper noun	0.16
5	'b' is a proper noun	0.35
6	There is a verb between 'a' and 'b'	−0.50
7	There is a preposition before 'a'	−0.43
8	There is an NP after 'c'	−0.93

SW_{score} is 1.32 + 0.58 + 0.42 + 0.16 = 2.48, which is greater than or equal to '1', and hence the LSP = 1.

4.1.2. Lexico-semantic probability

The noun-phrase pairs are formed using the candidate concept noun paired with each noun in the attribute target value in the triple. For each noun pair, the rank calculated using the WordNet similarity based semantic similarity measure is shown in Eq. (6). Overall, *NPrank* is the sum of all similarity scores of all noun phrase pairs. The lexico-semantic probability is calculated as shown in Eq. (5). If a predicate exists between the noun pairs, the lexico-semantic probability is assigned as *1*, and the rank of the noun phrase pair is greater than the mean threshold *h*. Otherwise, it is *0*.

$$P(LSemPt_{i,k}|c_l = \begin{cases} 1 & \text{if } P(LSP_{ti}|c_l), NP_{rank} > h \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

We used weighted links, a WordNet semantic similarity-based measure, to calculate the *NPrank* of two noun phrases in the noun phrase pairs. Weighted links (Richardson et al., 1994) are proposed for computing the similarity between two concepts using the WordNet taxonomy of two concepts. The weight of a link is calculated by (1) the density of the taxonomy at that point, (2) the depth in the hierarchy, and (3) the strength of connotation between parent and child nodes. The similarity between two concepts is computed by calculating the sum of weights of the links. We calculated the following three similarity scores, based on which the overall rank was calculated. Wu and Palmer (1994) similarity measure considers the position of concepts of c_1 and c_2 in the taxonomy relative to the position of the most specific common concept c. Li et al. (2003), which was intuitively and empirically derived, combine the shortest path length between two concepts. The measurement of Leacock and Chodorow (1998) is a relatedness measure between two concepts.

$$NP_{rank} = \sum_{i=1}^{N} sim_{w\&p}, sim_{Li}, sim_{lch}(NP_1, NP_2) \quad (6)$$

For example, the *w&p*, *Li*, and *lch* similarity values for the concept noun phrases "computer" and "processor" are 3.445068277596125, 2.0794415416798357, and 0.04632716059120144, respectively. Thus, the NP_{rank} is equal to 5.57083698. We initialized the threshold value '*h*' as the mean threshold value of 2.0. As the calculated NP_{rank} is greater than '*h*' and also the LSP is 1, the lexico-semantic probability LSemP is calculated as '1'. Thus, the training sample is assigned to the positive label '1', which indicates that the concept "computer" has a relation with the concept "processor" through the relationship "consists of".

After the whole training corpus is classified with an initial classifier, highly ranked triples are selected as the initial attribute class annotated set. From this, the parameters of the Naïve Bayes classifier are initialized. The second training stage is called the Expectation step. The whole training corpus, including the annotated part, is classified with the current classifier. The final training stage is called the Maximization step. Based on the newly classified data, parameters are re-estimated. The expectation and maximization steps are repeated while the classifier parameters converge.

4.2. Ontology concept schema modeling

The rough schema of the ontology concept is dynamically modeled using the set of concept relations extracted for the given seed concept. The ontology schema is generated with a bottom-up approach in which the attributes are identified using instances. An attribute is considered for inclusion into the target schema when there is an existing relationship between the candidate concept and associated concept keyword in the instance. A sample ontology schema constructed using this approach is depicted in Fig. 5.

5. CR-Ontology portable question answering framework

The proposed framework is similar to Watson's three component architecture (Hirschman and Gaizauskas, 2001), which describes the approach taken to build QA systems. Our proposed framework consists of a (1) question analysis component, (2) answer extraction component and (3) automatic ontology construction component. Fig. 6 depicts our proposed three component framework for our Concept Relational Ontology-based Question Answering system (CRO-QAs).

The role of the question analysis component is to identify the question type (QT) from which the expected answer target (AT) is selected. We used an AT database consisting of 52 QTs and corresponding ATs.

To extract an answer from the ontology for the natural language query, we utilized Attribute-Based Inference (ABI), which was introduced by Ferrnandez et al. (2009). The ontology attribute that is available in the submitted query is identified using ABI. An ontology attribute considered for generating an answer to a query depends on the ABI score. The score value is obtained by using positive weights assigned to the patterns matched between the query attribute and ontology attribute. The final weight obtained by this inference is defined as shown in Eq. (7).

$$ABI_{score} = \frac{\sum_{a_i \in Q, a_j \in O} Eql(a_i, a_j)}{|Q|} \quad (7)$$

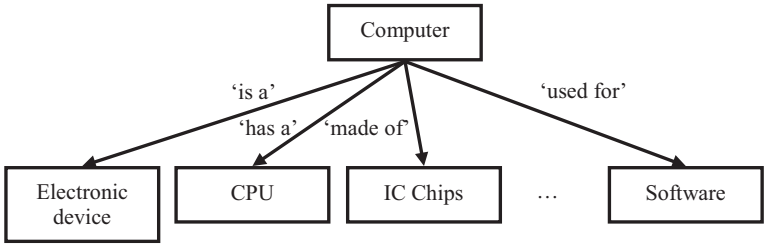

Figure 5 A sample concept schema of the constructed ontology.

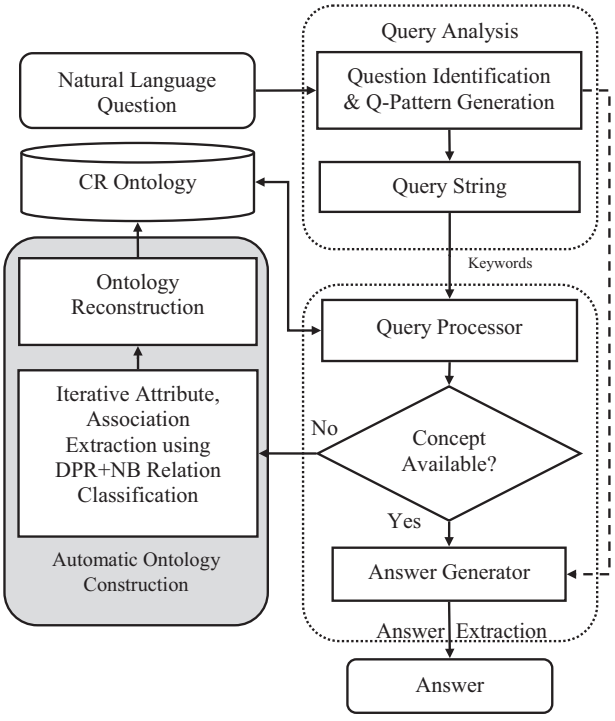

Figure 6 Proposed ontology based question answering system architecture.

where O is the list of ontology attributes and |Q| is a set of query attributes. Then, $Eql(a_i,a_j)$ is calculated using the Eq. (8).

$$Eql(a_i,a_j)=\{\begin{matrix}1 & if & a_i=a_j & \text{or} a_i\in a_j\\ 0 & & & \text{otherwise}\end{matrix}\} \tag{8}$$

For each positive inference, a similarity weight between zero and one is assigned, and then, the final entailment coefficient is calculated using the sum of all weights divided by the number of inferences. We empirically established a threshold using the number of user query patterns, based on which the entailment decision was made. The answer is constructed using the ontology attribute with entailment coefficients higher than the threshold. When the relevant ontology attribute to the input query is not present in the ontology, the procedure for automatic ontology construction for the new concept is initiated. Once again, the answer construction process is restarted after updating the existing ontology with the newly constructed ontology concept.

6. Experimental setup

6.1. Relation extraction for automatic ontology construction

The proposed relation extraction algorithm was implemented using Java, and we ran the implementation to extract 1000 concepts from ten different domains. Each concept extraction was experimented using the benchmark TREC-QA 2008 dataset. The data set was validated using the10-fold cross validation technique. The data-set was clustered into 10 equal partitions with 100 instances each. Then, the experiment was repeated by changing the validation data set '*k*' from 1 to 10. For each experiment, we calculated the mean accuracy and mean error (ME). A similar experiment was conducted for all data samples that cover the concepts belonging to ten different domains. Finally, the confidence interval (CI) value was calculated for each mean accuracy value by using a *t*-test. The ontology concepts were constructed using the maximum of eight attributes and twelve associations. We used the standard measure of precision, recall and *F*1-measure in the field of information extraction for calculating the relation extraction accuracy.

The precision *P* is defined as shown in Eq. (9).

$$P=\frac{|\{\text{Relevant}\}\cap\{\text{Found}\}|}{|\text{Found}|} \tag{9}$$

The Recall *R* is defined as shown in Eq. (10).

$$R=\frac{|\{\text{Relevant}\}\cap\{\text{Found}\}|}{|\text{Relevant}|} \tag{10}$$

where Relevant is the set of relevant relations (attributes or associations) and Found is the set of found relations. There is a trade-off between precision and recall, and thus, the *F*1 measure is computed ($\beta = 1$). The *F*1 measure is applied to compute the harmonic mean of precision and recall as shown in Eq. (11).

$$F_1=\frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{11}$$

6.2. Question answering using CR-Ontology

Our experimental model was designed to evaluate our proposed CR-Ontology based question answering system. We used the benchmark TREC-QA 2008 data set with definition questions and factoid questions that cover all 10 domains. In our entailment evaluation, 10 new users were asked to formulate five queries for each domain stored in CR-Ontology. In total, 500 new input queries were generated. These new queries were used to adjust the entailment decision threshold and to evaluate the final system performance. The accuracy was calculated using the ratio between the number of questions correctly answered by the system and the total number of questions submitted to the system. To evaluate the effectiveness of our proposed system, we compared the overall accuracy with the accuracy obtained by the well-performing benchmark TREC QA systems and another ontology-based QA system, QACID, which uses entailment and an on-field evaluation framework.

7. Results and discussion

7.1. Relation extraction for automatic ontology construction

We hypothesize that the creation of well-performing dependency parsing-based hand coded rules with a self-supervised learning approach will deliver a greater accuracy when compared with the existing semi-supervised and unsupervised methods. The rule set exploits the relationship between the candidate concept and target concepts by using the predicate that connects both. Because of the very few and limited number of rules that are executed in predetermined sequence, the design of the recursive BDT-based rule engines naturally reduces the complexity of eliminating the negative samples and allowing remaining subclass to the next iteration. For each

successful relation extraction, it takes only three decision computations and the number of iterations is directly proportional to the number sentences in the training set. Thus, the DPR engine is comparatively less expensive than the other compared methods.

The 10-fold cross validation results obtained with 10 different domains of data are presented in Table 3. The *t*-test was performed on the overall accuracy obtained for each domain data for a 95% confidence value (level of significant $\alpha = 0.05$). All of the mean accuracy values fall within the calculated CI with '*p*' value less than 'α'.

The proposed method achieved the highest accuracy of 95.63% in the electronic domain, and the overall mean accuracy is 90.55%, which is 10–15% higher than the best performing relation extraction method Snowball_VS. The 10-fold cross validation results of the electronic domain data sample are presented in Fig. 7. The standard error value is minimum for the value of $k = 7$. The comparison between the relation extraction performance (*F*1-accuracy with $\beta = 1$) obtained by our proposed method and Snowball_VS method for the same data set is visualized in Fig. 8. Except for the finance and automobile domains, our proposed method achieved better accuracy. Thus, the objective of achieving better performance by using a DPR-based self-supervised method was achieved.

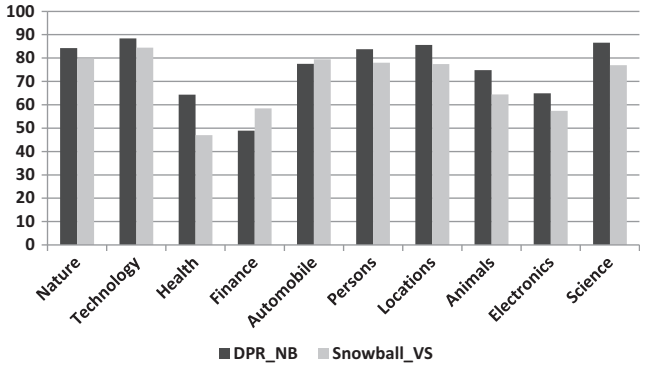

Figure 8 Relation extraction performance comparisons between proposed DPR_NB method and a well-performing Snowball_VS method.

7.2. Question answering using CRO-QAs

The user queries formulated by the entailment evaluation process were used to experiment with our proposed CR-Ontology-based question answering system. In addition, the answers were extracted for the definition questions in the TREC 2008

Table 3 Relation extraction performance results, 10-fold cross validation performed on ten different domain data with 95% confidence interval (CI).

Data set	*t*	df	Sig. (2-tailed)	Mean difference	95% Confidence interval of the difference	
					Lower	Upper
Nature	22.740	9	.000	86.94600	78.2967	95.5953
Technology	28.519	9	.000	91.13000	83.9015	98.3585
Health	49.016	9	.000	91.29000	87.0768	95.5032
Finance	37.513	9	.000	83.96000	78.8969	89.0231
Automobile	52.818	9	.000	92.55000	88.5861	96.5139
Persons	49.655	9	.000	90.80000	86.6634	94.9366
Locations	52.487	9	.000	92.59000	88.5994	96.5806
Animals	27.366	9	.000	86.85000	79.6706	94.0294
Electronics	103.481	9	.000	95.63000	93.5395	97.7205
Science	67.849	9	.000	93.36500	90.2521	96.4779

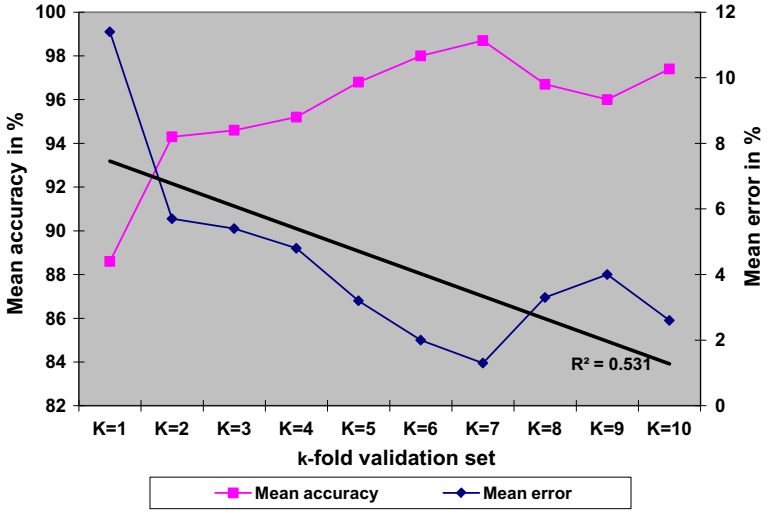

Figure 7 Results of 10-fold cross validation performance on electronics domain data set, *F*1 accuracy mean (%) and error mean (%).

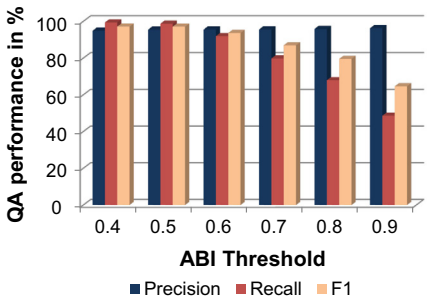

Figure 9 Question answering performance of proposed CRO-QA system, precision, recall and *F*1-measures obtained with entailment query data set with different ABI threshold values.

Table 4 QA accuracy of proposed and compared QA systems.

Systems	TREC 2007 BEST	TREC 2009 (QA@CLEF)	QACID	CRO-QAs
Accuracy ($\beta = 1$)	0.706	0.61	0.932	0.97

data set. We used the precision, recall and *F*1 measure for evaluating our proposed CRO-QAs and to compare with the well-performing benchmark QA systems and a similar ontology-based QA system, QACID. The experiment was conducted in an iterative manner by varying the entailment threshold from 0.4 to 1 with a 0.1 scale. Fig. 9 depicts the resulting precision, recall and accuracy ($\beta = 1$) obtained for the six different ABI threshold values. Our proposed method achieved the maximum recall of 99% without compromising the precision (96%) for the ABI threshold value of 0.5. The maximum recall value was achieved for the same ABI threshold value as that of the compared ontology-based question answering system, QACID. However, there is a great improvement in the accuracy percentage achieved by our proposed method. The overall QA accuracy obtained by our proposed method and other well-performing QA systems is given in Table 4. It is evident that the performance of our system is much better than the best-performing QA systems in terms of the TREC 2008 and TREC 2009 benchmark.

8. Conclusion

A system for automatically extracting attributes and associations from a large volume of unstructured text for automatic domain ontology modeling was successfully developed, and the experimental results were presented in this paper. The proposed dependency parsing pattern-based iterative concept relation extraction algorithm was implemented to extract attributes and associations using lexico-syntactic and lexico-semantic probabilities. The empirical results were encouraging, and it has been proven that our proposed method outperforms similar well-performing relation extraction methods. The suitability of the constructed concept relational ontology for use with ontology portable question answering systems was experimentally evaluated using our concept relational ontology-based question answering framework. The system performance was above average for all three question types: factoid, list, and definition. The main objectives of this research of automatically constructing a domain ontology using concept relations and creating QA systems capable of precisely answering natural language questions without compromising the efficiency and accuracy were achieved. It is encouraging that not only are the techniques introduced in this paper capable of answering questions relatively quickly, but their answer performance is better than the available web-based and ontology-based QA systems when independently evaluated using a benchmark data-set. The proposed QA framework can be extended to generate answers for more complex types of questions by introducing additional natural language techniques.

References

Agichtein, E., Gravano, L., 2000. Snowball: extracting relations from large plain-text collections. In: Fifth ACM Conference on Digital libraries, 2–7 June 2000, San Antonio, TX, USA, pp. 85–94.

Areanu, N.T., Colhon, M., 2009. Conditional graphs generated by conditional schemas, Annals of the University of Craiova. Math. Comput. Sci. Ser. 36, 1–11.

Aussenac-Gilles, N., Despres, S., Szulman, S., 2008. The TERMINAE method and platform for ontology engineering from texts. In: Proceeding of the 2008 Conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge, pp. 199–223.

Banko, Michele, Michael, J., Cafarella, Stephen Soderland, Matt Broadhead, Oren Etzioni, 2007. Open information extraction from the web. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence, pp. 2670–2676.

Berant, Jonathan, Ido Dagan, Jacob Goldberger, 2011. Global learning of typed entailment rules. In: Forty Ninth Annual Meeting of the Association of Computational Linguistics: Human Language Technologies, 19–24 June 2011, Portland, Oregon, USA, pp. 610–619.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.H., 1984. Classification and Regression Trees. Chapman & Hall, Wadsworth, Belmont, CA.

Brin, S., 1998. Extracting patterns and relations from the world wide web. In: WebDB Workshop at EDBT, 27–28 March 1998, Valencia, Spain, UK, pp. 172–183.

Carlson, Andrew, et al., 2010. Toward an architecture for never ending language learning. In: Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, pp. 1306–1313.

Cowie, Jim, Wilks, Yorick, 2000. Information extraction. In: Dale, R., Moisl, H., Somers, H. (Eds.), Handbook of Natural Language Processing. Marcel Dekker, New York, pp. 249–269.

Dempster, A.P., Maird, L.N., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. 39, 1–38.

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D., Yates, A., 2005. Unsupervised named-entity extraction from the web: an experimental study. Artif. Intell. 165, 91–134.

Ferrnandez, Oscar, Izquierdo, Rubén, Ferrández, Sergio, Vicedo, José Luis, 2009. Addressing ontology-based question answering with collections of user queries. Inf. Process. Manage. 45, 175–188.

Gacitua, R., Sawyer, P., Rayson, P., 2008. A flexible framework to experiment with ontology learning techniques. Knowl. Based Syst. 21, 192–199.

Hearst, M.A., 1992. Automatic acquisition of hyponyms from large text corpora. In: Fourteenth Conference on Computational Linguistics (COLING 1992), pp. 539–545.

Hearst, M., 1998. Automated discovery of WordNet relations. In: Fellbaum, Christiane. (Eds.), WordNet: An Electronic Lexical

Database and Some of its Applications, MIT Press, USA, pp. 131–151.

Hirschman, Lynette, Gaizauskas, Robert, 2001. Natural language question answering: the view from here. Nat. Lang. Eng. 7, 275–300.

Horrocks, I., 2008. Ontologies and the semantic web. Commun. ACM 51, 58–67.

Hou, Xin, Ong, S.K., Nee, A.Y.C., Zhang, X.T., Liu, W.J., 2011. GRAONTO: a graph-based approach for automatic construction of domain ontology. Expert Syst. Appl. 38, 11958–11975.

Jung, Yuchul, Ryu, Jihee, Kim, Kyung-Min, Myaeng, Sung-Hyon, 2010. Automatic construction of a large-scale situation ontology by mining how-to instructions from the web. Web Semant.: Sci. Serv. Agents World Wide Web 8, 110–124.

Kang, B.Y., Myaeng, S.H., 2005. Theme assignment for sentences based on head-driven patterns. In: Proceedings of Eighth Conference on Text Speech and Dialogue (TSD), pp. 187–194.

Krishnamurthy, Jayant, Mitchell, Tom M., 2013. Jointly learning to parse and perceive: connecting natural language to the physical world. Trans. Assoc. Comput. Linguist. 1, 193–206.

Kwok, C., Etzioni, O., Weld, D., 2001. Scaling question answering to the web. Proceedings of the Tenth World Wide Web Conference, pp.150–161.

Leacock, Claudia, Chodorow, Martin, 1998. Combining Local Context and WordNet Similarity for Word Sense Identification. The MIT Press, USA.

Li, Yuhua, Zuhair, A., Bandar, David McLean, 2003. An approach for measuring semantic similarity between words using multiple information sources. IEEE Trans. Knowl. Data Eng. 15, 871–882.

Thomas Lin, Mausam, Oren Etzioni, 2010. Identifying functional relations in web text. In: 2010 Conference on Empirical Methods in Natural Language Processing, 9–11 October 2010, MIT, Massachusetts, USA, pp. 1266–1276.

Liu, B., Lee, W.S., Yu, P.S., Li, X., 2002. Partially supervised classification of text documents. In: Nineteenth International Conference on Machine Learning, 8–12 July 2002. Australia, Sydney, pp. 387–394.

Lopez, V. et al, 2007. AquaLog: an ontology-driven question answering system for organizational semantic intranets. J. Web Semant. 5, 72–105.

Maedche, A., 2002. Ontology Learning for the Semantic Web. The Kluwer Academic Publishers in Engineering and Computer, Science, p. 665.

Marie-Catherine de Marneffe, Christopher, D., Manning, 2008. The Stanford typed dependencies representation. In: COLING Workshop on Cross-framework and Cross-domain Parser Evaluation, 23 August 2008, Manchester, UK, pp. 1–8.

Mc Guinness, D., 2004. Question answering on the semantic web. IEEE Intell. Syst. 19, 82–85.

Miller, G.A., 1995. WordNet – a lexical database for English. Commun. ACM 38, 39–41.

Navigli, R., Velardi, P., Gangemi, A., 2003. Ontology learning and its application to automated terminology translation. IEEE Intell. Syst. 18, 22–31.

Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T., 2000. Text classification from labeled and unlabeled documents using EM. Mach. Learn. 39, 103–134.

Minyoung Ra, Donghee Yoo, Sungchun No, Jinhee Shin, Changhee Han, 2012. The mixed ontology building methodology using database information. In: International Multi-conference of Engineers and Computer Scientists, 14–16 March 2012, Hong Kong, China, pp. 650–655.

Richardson, R., Smeaton, A., Murphy, J., 1994. Using WordNet as a knowledge base for measuring semantic similarity between words. Technical Report Working paper CA-1294, School of Computer Applications, Dublin City University, Dublin, Ireland.

Ritter, Alan, Mausam, Oren Etzioni, 2010. A latent dirichlet allocation method for selectional preferences. In: Forty Eighth Annual Meeting of the Association for Computational Linguistics; 11–16 July 2010, Uppsala, Sweden, pp. 424–434.

Mohamed Said Hamani, Ramdane Maamri, Yacine Kissoum, Maamar Sedrati, 2014. Unexpected rules using a conceptual distance based on fuzzy ontology. J. King Saud Univ. – Comput. Inf. Sci. 26, 99–109.

Sarwar Bajwa, Imran, Mark Lee, Behzad Bordbar, 2012. Translating natural language constraints to OCL. J. King Saud Univ. – Comput. Inf. Sci. 24, 117–128.

Stefan Schoenmackers, Oren Etzioni, Daniel, S., Weld, Jesse Davis, 2010. Learning first-order horn clauses from web text. In: 2010 Conference on Empirical Methods in Natural Language Processing, 9–11 October 2010, MIT, Massachusetts, USA, pp. 1088–1098.

Takagi, Noboru, 2006. An application of binary decision trees to pattern recognition. J. Adv. Comput. Int. Intell. Inform. 10, 682–687.

Voorhees, E.M., 1999. The TREC-8 question answering track report. In: Proceedings of the Eighth Text REtrieval Conference, NIST Special Publication, pp. 500–246.

Wu, Z., Palmer, M., 1994. Verb semantics and lexical selection. In: Proceedings of the Thirty second Annual Meeting of the Associations for Computational Linguistics (ACL'94), Las Cruces, New Mexico, pp. 133–138.

Wu, Fei, Daniel S. Weld, 2010. Open information extraction using Wikipedia. In: Proceedings of the Forty eighth Annual Meeting of the Association for Computational Linguistics (ACL '10'), 11–16 July 2010, Morristown, NJ, USA, pp. 118–127.

Yangarber, Roman, Ralph Grishman, 2001. Machine learning of extraction patterns from unannotated corpora: Position statement. In: Proceedings of Workshop on Machine Learning for Information Extraction, pp. 76–83.

Zhang, P., Li, M., Wu, J., et al, 2005. The community structure of science of scientific collaboration network. Complex Syst. Complexity Sci. 2, 30–34.

Zhu, Jun, Zaiqing Nie, Xiaojiang Liu, Bo Zhang, Ji-Rong Wen, 2009. StatSnowball: a statistical approach to extracting entity relationships. In: Eighteenth international conference on World Wide Web, 20–24 April 2009, Madrid, Spain, pp. 101–110.

Zouaq, Amal, Gasevic, Dragan, Hatala, Marek, 2011. Towards open ontology learning and filtering. Inf. Syst. 36, 1064–1081.