

Using Automatic Ontology Learning Methods in Human Plausible Reasoning Based Systems

A. R. Vazifedoost, M. Rahgozar, and F. Oroumchian

Abstract—Knowledge discovery from text and ontology learning are relatively new fields. However their usage is extended in many fields like Information Retrieval (IR) and its related domains. Human Plausible Reasoning based (HPR) IR systems for example need a knowledge base as their underlying system which is currently made by hand. In this paper we propose an architecture based on ontology learning methods to automatically generate the needed HPR knowledge base.

Keywords—Ontology Learning, Human Plausible Reasoning, knowledge extraction, knowledge representation.

I. INTRODUCTION

HUMAN Plausible Reasoning (HPR) theory [1],[2] is based on the way human thinks and reasons. It says having a knowledge base of information, what kind of inference patterns plausibly human uses to reach to an answer.

This theory mainly consists of two parts. In the first part it describes the kind of information representation we have in mind. The other part of theory says what kind of inference patterns we use for getting new information from our knowledge base. These patterns are results of investigating many peoples in the way they answer questions.

This idea has been used in many Information Retrieval (IR) related applications [13],[14],[15],[16],[17],[18]. However, one of main bottlenecks in these applications is providing the proper knowledge base. Human learns his knowledge, or in other words some concepts and the relationships between them, by means of experiences and his thinking skills during his life. Then he can make some inferences based on his knowledge. But a machine has no means to learn this knowledge unless what we give it.

The brief meaning of ontology is a shared kind of knowledge representation. It contains many concepts and represents the relations between them. These ontologies are mainly constructed by some experts who define what concepts exist and how they relate to each other. However In the last

decade there have been many efforts to automatically learning ontologies [4],[5],[7],[10],[14]. Their main idea is that having a great amount of knowledge written in free text format, we can extract concepts and knowledge from them.

Because of analogy between an ontology and HPR knowledge representation, we can potentially use ontology learning methods to achieve a HPR knowledge base. Here we suggest such a solution based on a proposed architecture for providing a proper knowledge base for HPR. This is through applying automatic ontology learning methods and then mapping their representation format to HPR format of knowledge representation.

II. BASIC ARCHITECTURE OF SYSTEM

Our goal is to introduce a system for automatically learning HPR kind of knowledge base, using adapted ontology learning methods. The source of this knowledge is relied on a text corpus instead of some expert's mind. A general schema for the architecture of the system is depicted in Fig. 1.

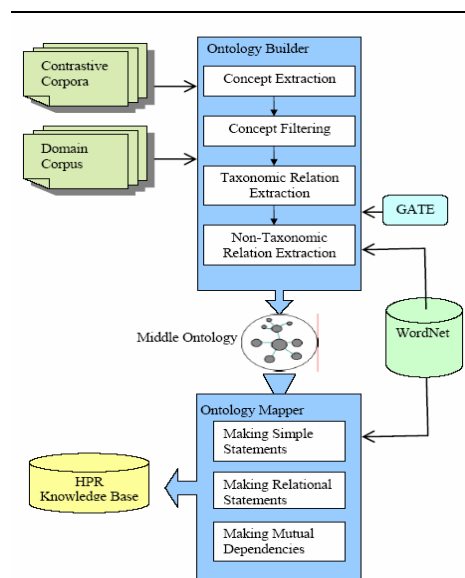


Fig. 1 General schema of our system's architecture

In Fig. 1, we have two kinds of corpora as the inputs of the system. The first one is a domain specific corpus which contains domain related content. We use this corpus as the source of the ontology's knowledge. The second one is a contrastive corpus. This corpus will help us to identify domain related concepts from the first corpus. We'll see more about

A. R. Vazifedoost is with the Database Research Group, faculty of ECE, School of Engineering, University of Tehran, Tehran, Iran (e-mail: a.vazifedoost@ece.ut.ac.ir).

M. Rahgozar is member of the Control and Intelligence Processing Center of Excellence, faculty of ECE, School of Engineering, University of Tehran, Tehran, Iran (e-mail: rahgozar@ut.ac.ir).

F. Oroumchian is with the College of IT, University of Wollongong in Dubai and member of Control and intelligent processing center of excellence of University of Tehran (e-mail: farhado@uow.edu.au).

this in next sections.

We include the General Architecture for Text Engineering (GATE) [3] from the Sheffield University in this architecture. That is an open source framework with reach Natural Language processing components. Also, GATE has a pattern extraction component, named JAPE which we have used it extensively in this project.

WordNet is other external source as a general ontology. We take the benefit of this ontology both in Ontology Builder subsystem and Ontology Mapper subsystem as a consultant source.

Two main subsystems are Ontology Builder and Ontology Mapper. The Ontology Builder is essentially the first producer in our architecture which will use ontology learning methods to create a middle ontology. It is named the *middle ontology* because it has to be transformed to our desired kind of knowledge representation. This transformation is the task of the next subsystem which is Ontology Mapper. This subsystem maps the middle ontology to a HPR knowledge base with its specific kind of knowledge representation.

We'll describe each of these two subsystems in next subsections and will explain what kind of methods will be used for performing the Task.

III. ONTOLOGY BUILDER

Ontology learning systems are different considering their approaches for extracting knowledge. However there are some common steps taken in most of them. Our proposed system works on a four steps process which is depicted in Fig. 1. These are the more common steps between different ontology learner systems.

A. Concept Extraction

This is the first step in most systems. Through this step, learning is focused on extracting concepts from a text corpus.

Usually noun phrases (NP) and nouns (N) are best candidates for being our concepts. That's because, these elements often have more semantic value than other parts of speeches.

For extracting these elements we use a kind of parsing which is called "Shallow parsing". This kind of parsing extracts less semantic information from text than "Deep parsing" but on the other hand it is faster.

There are three phases in shallow parsing. The first phase is part of speech tagging which determines grammatical class of each word. Second phase is specified for chunking in which words group together to make more meaningful noun phrases. The third one finds relations between words and the verb of sentences (e.g. subject, object,...). GATE framework is used in this step.

B. Concept Filtering

All of found concepts are not necessarily useful for us. This is determined by the amount of relevance of a concept to our intended domain. For example if you extract "SQL" as a concept in computer engineering field, this quite makes

senses. However "SQL" is not a related concept in telecommunication field by any means, even if we find it in some of our documents.

We explain three methods for concept filtering:

Domain Relevancy Parameter (DR) [6]: High term frequency in a corpus is a property for concepts which are both related and unrelated to our desired domain. The idea behind DR is that we can measure the relevance of a concept to a domain through a comparative analysis across different domain corpora. More precisely, given a set of n domains $\{D_1, \dots, D_n\}$ and related corpora, the domain relevance of a concept t in class D_k is computed as

$$DR_{t,k} = \frac{P(t | D_k)}{\max_{1 \leq j \leq k} p(t, D_j)} \quad (1)$$

Where conditional probabilities $P(t|D_k)$ are measured as

$$Estimate(P(t | D_k)) = \frac{f_{t,k}}{\sum_{t \in D_k} f_{t,k}} \quad (2)$$

Where $f_{t,k}$ is the frequency of term t in the domain D_k (i.e, in its related corpus).

Simply we can say that, more related concepts to a specified domain are appeared more frequently in that domain in contrast to other domains.

Domain Consensus Parameter (DC)[6]: It's important to have concepts on which more experts agree on their relevancy to our domain. However, during automatic construction of ontology we use some approaches for *simulating* this consensus because there is no real expert here.

We assume distribution of concepts can *simulate* a type of consensus over concepts. That's because the domain specific concepts are repeated more frequently in several documents of that domain while a not related concept can just occur in few documents (even if that concept has a high frequency in those documents).

We use DC parameter to show consensus over a concept which represents distribution of that concept over documents $d \in D_k$. More precisely, the domain consensus is expressed by entropy as follows:

$$DC_{t,k} = \sum_{d \in D_k} \left(P_t(d) \log \frac{1}{P(d)_t} \right) \quad (3)$$

Where

$$Estimate(P_t(d_j)) = \frac{f_{t,j}}{\sum_{d_j \in D_k} f_{t,k}} \quad (4)$$

The basic claim is that, a more distributed concept is more specific to a certain domain.

1. Combination of DC and DR

Practically both of the DC and DR parameters are important in weighting concepts. Therefore total weight will be:

$$TW_{t,k} = \alpha DR_{t,k} + \beta DC_{t,k}^{norm} \quad (5)$$

Where $DC_{t,k}^{norm}$ is a normalized entropy and $\alpha, \beta \in (0,1)$. This parameters should be tuned based on the corpus properties but according to [6] α will be close to 0.9 and if

the number of documents in corpus is great enough, β will be between 0.25 and 0.35.

C. Taxonomic Relations

These relations give a hierarchical structure to ontology. Two kinds of these relations are “IS-A” and “Has-A”¹.

There are some suggested methods for extracting this relation from text [7],[8],[9]. Although we’ve adopted just one method for “IS-A” relations and one method for “Has-A” relations through Lexico-Syntactic Patterns.

These patterns are basically some regular expressions. These regular expressions are found by experiments [7],[8] for finding both “IS-A” relations and “Has-A” relations. First we’ll explore their usage in finding “IS-A” relations.

1. IS-A Patterns

Some Lexico-Syntactic patterns which we’ll use for extracting “IS-A” relations are depicted in Fig. 2. When these patterns match with some portion of text, we can then infer:

$$\forall NP_i, 1 \leq i \leq n, \quad IS-A(\text{head}(NP_i), \text{head}(NP_0))$$

(1) NP_0 such as $NP_1, NP_2, \dots, NP_{n-1}$ (and/or) NP_n^2
 (2) such NP_0 as $NP_1, NP_2, \dots, NP_{n-1}$ (and/or) NP_n
 (3) NP_1, NP_2, \dots, NP_n (and/or) other NP_0
 (4) NP_0 , (including) especially $NP_1, NP_2, \dots, NP_{n-1}$ (and/or) NP_n
 (5) NP_1 is NP_0
 (6) NP_1 , another NP_0
 (7) NP_0 like NP_1

Fig. 2 Lexico-Syntactic patterns for extracting "IS-A" relations

We can measure a confidence factor for detected matches. This confidence is computed through the following formula:

$$\text{Conf}(GC, Ci) = \frac{\text{freq}(GC, Ci)}{\text{freq}(GC)} \quad (6)$$

Where the $\text{Conf}(GC, Ci)$ is the confidence value. Also Ci and GC are two concepts for which we have “Ci IS-A GC” and $\text{freq}(GC, C)$ is number of times two concepts GC and Ci matched through the patterns. Finally, $\text{freq}(GC)$ is the frequency number of GC .

2. Has-A Patterns

Just like previous patterns for “IS-A” relations, we can expect other patterns which detect “Has-A” relations. Current pattern collection which we use is based on [8] and [9]. These patterns are depicted in figure 3. “Whole” in this figure refers to owner of an attribute and “Part” refers to that attribute.

Like previous relationship, for each “Has-A” relation we need to assign a confidence factor which will be used further in Ontology Mapper subsystem. This is a simple formula as following:

$$\text{Conf}(\text{Whole}, \text{Part}) = \frac{\text{freq}(\text{Whole}, \text{Part})}{\text{freq}(\text{Whole})} \quad (7)$$

Where $\text{Conf}(\text{Whole}, \text{Part})$ is the value of confidence, $\text{freq}(\text{Whole}, \text{Part})$ is the number of times Whole and Part are co-occurred together in above patterns, and also $\text{freq}(\text{Whole})$ is the total number of times which Whole concept is occurred in the corpus.

¹ “Part-of” in reverse direction

A. whole {NN | NP}[-S]'s part {NN | NP}[-S]
 ... building's basement...

B. part {NN|NP}[-S] of {the|a}
 [JJ|NN]* whole NN[-S]
 ... basement of a building...

C. part {NN|NP} in {the|a} [JJ|NN]* whole NN
 ... basement in a building...

D. parts NNS of wholes NNS
 ... basements of buildings...

E. parts NNS in wholes NNS
 ... basements in buildings...

F. part JJ whole NP
 ... blue eye...

G. whole {NP|NN} NPS|NNS; Subj
 to have part {NP|NN} NPS|NNS; Obj
 ... country has a capital...

NP: Noun Phrase
 NN: Noun
 NNS: Plural Noun
 NPS: Plural Noun Phrase

Fig. 3 Patterns for "Has-A" Relation

D. Non-Taxonomic Relations

Any other relations between two concepts other than taxonomic relations are considered as non-taxonomic relations. These relations may convey possession, causation, synonymy, Antonymy or any other kind of relations. We use four kinds of methods which are Association Rules, sub-Categorization Frames, Causal relations and Similarity relations.

1. Association Rules

Let’s consider a set of concepts $A = \{w_1, w_2, \dots, w_n\}$ and also a collection of documents like $T = \{t_1, t_2, \dots, t_n\}$, i.e. each t_i is associated with a subset of A, denoted by $t_i(A)$.

Let $W \subseteq A$ be a set of concepts, the set of all documents t in T such that $W \subseteq t(A)$ will be called *covering set* for W and denoted $[W]$.

Any pair (W, w) , where $W \subseteq A$ is a set of concepts and $w \in A \setminus W$, will be called association rule, and denoted by $R: (W \Rightarrow w)$. Although our relations in current system are binary and just two concepts are involved in each relation.

Given an association rule $R: (W \Rightarrow w)$:

$$C(R, T) = \frac{|[W \cup \{w\}]|}{|\{w\}|} \quad (8)$$

is called the **confidence** of R, with respect to collection T. By $C(R, T)$ we mean the probability of existing a concept w in a document if there is already concept set W in the same document.

We assign a “Related-To” label for such relationships because the type of relationship is unknown. An example of this relationship is shown in Fig. 4 in which the association between ‘gold’ and two countries is determined.

Query: 'find all associations between a set of topics including gold and any country.'
 Result: [gold] => Canada 5/0556
 [gold] =>USA 18/0.692

Fig. 4 Sample association rules

2. Sub-Categorization Frames

Sub-categorization frames as our second types of relation are based on an idea from ASIUM [11]. However the algorithm which we use is quite different because we don't have clustering mechanism which is introduced there.

These frames in fact determine the kind of words which are more usual to come with specified verbs as a subject or object. For a better imagination see Fig. 5.a in which 'jack' and 'bike' are more frequent to come with verb, 'travel-by' in a hypothesized corpus. This relation can be generalized to convey a relation between 'human' and 'vehicle' as depicted in Fig. 5.b. The concepts which play subject or object roles together with the verb can convey a very valuable relationship.

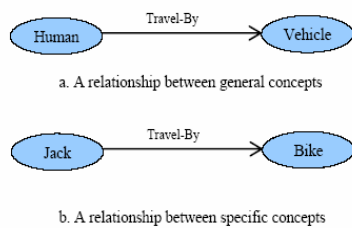


Fig. 5 Relations based on sub-categorization frames

Our proposed algorithm for finding such relations and generalizing them is as follows:

- 1- Use the shallow parsed corpus to extract the patterns like "Subject verb Object".
- 2- For each verb consider the set of (subject, object) tuples as the lowest level in a hierarchy. Each of these tuples delegate a relation.
- 3- As a loop do the following in each steps
- 4- Divide each node to three nodes as you see in figure 6.
- 5- Combine every nodes that are just like together.
- 6- If just remained one node e.g.(object,object), finish, else go to step 3.

As the algorithm progresses we find relations with more general concepts. For example it can start from relation b in Fig. 6 and end in relation a in the same figure.

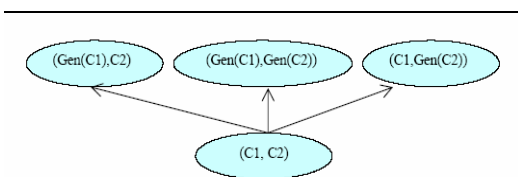


Fig. 6 Each tuple will be divided into tree tuples. "Gen" stands for General

For each node which represents a relation a confidence value should be assigned. This value represents our certainty about the relationship. Measurement of this value is different for the first level of nodes which are extracted directly from the text.

The confidence for the first level nodes can be achieved through the following formula:

$$conf = \frac{Nvd}{N} \times \frac{Nvn}{Vn} \quad (9)$$

Where in the first part of this notation, Nvd stands for the number of documents which our desired verb is appeared in. Also N stands for the total number of documents in our corpus. This part shows the distribution of our verb.

In the second part, Nvn is the number of times which our specified verb is appeared with concepts inside a node. Also Vn is the total frequency of this verb.

The confidence factor for inner nodes can be measured in two cases. In first case consider the nodes which are not from some merged nodes and are originated directly from a below node. In this case we assign the confidence of source node for derived nodes. But in the second case, if a node is produced from some merged nodes, then the confidence will be average of the confidence of merged nodes.

3. Causal Relations

Method which we use for causal relation extractions is base on a work in [12]. This work is relying on extracting lexico-syntactic causal patterns in a corpus. However these patterns create ambiguous extracted relations. Though for mitigating this problem, some semantic constraint is predicted. We will explain this method in more detail in the following.

The algorithm has two main procedures. At first we should discover the causal Lexico-Syntactic patterns and then we should apply the semantic constraints over items which matched with patterns.

a) First Step

1-Select two concepts C1 and C2 which we know some causal relationship exists between them.

2-Find all patterns like "C1 verb C2". This will get the causal relations which may have ambiguity.

3-Search the corpus to find some matches for found patterns.

b) Second Step

This technique doesn't use the common word sense disambiguation algorithms. Instead it applies some constraints over detected matches and tries to rank them based on their validation. These constraints are based on the place of concepts in WordNet hierarchy which consider the parents of a concept in that hierarchy.

Two types of these constraints are, first on C1 and C2 and second on the verb. We will not explore semantic constraints on cause and the effect more because of limitation in page number, more information may be found in [12].

The result is some relations with a ranking between 1 to

5. This ranking will be used in Ontology Mapper subsystem.

4. Finding Similarity Relationship

“Has-A” relations are one kind of taxonomic relations which we extract. We believe that concepts which are connected to a concept through “Has-A” relation can be considered as the features of that concept.

The similarity measurement between two concepts is based on how many features two concepts have in common. Having this idea we used a simple measurement as following:

$$SIM(C1, C2) = \frac{|Features(C1) \cap Features(C2)|}{|Features(C1) \cup Features(C2)|} \quad (10)$$

Where $Features(C_i)$ denotes the set of concepts which are linked to C_i through “Has-A” relations as you can see in Fig. 7.

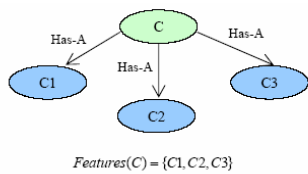


Fig. 7 Feature set of Concept "C"

IV. ONTOLOGY MAPPER

The second important subsystem is the “Ontology Mapper”. The duty of this subsystem is producing appropriate knowledge representation for an HPR based system. The source of this knowledge would be the middle ontology which is produced during the previous phase of ontology learning by “Ontology Builder” subsystem.

A. Knowledge representation in HPR

By knowledge representation we mean the kind of relating concepts. Brief overview of HPR’s knowledge representation in Fig. 8. There are four major elements in this kind of knowledge representation:

- 1) **Statements:** This statements consist of a descriptor (d) applied to an argument (a) and realized by a referent (r). A simple statement like “flying is the means of locomotion of birds” can be represented by “ means-of-locomotion (birds) = {Flying...}”. The brackets and dots around the referent indicated that there may be other means of locomotion for birds, such as walking.
- 2) **Relational Statement:** The second kind of expression involves one of four relations: generalization (GEN) (e.g. Bird GEN robin), specialization (SPEC) (e.g. Chicken SPEC fowl), similarity (SIM) (e.g. duck SIM goose), and dissimilarity (DIS) (Women DIS Man).
- 3) **Mutual Implication:** a mutual implication specifies how statements (or compound statements) are related together. The example in Fig. 8 states that warm temperature and heavy rainfall imply rice growing and vice versa.
- 4) **Mutual Dependency:** mutual dependency relates two terms for example, latitude(Place) and temperature(place).

The example in Fig. 8 represents the belief that the average temperature of a place is inversely related to its latitude.

<p>Statements (S) $d(a) = r$ means-of-locomotion(birds) = {flying . . .}</p> <p>Relational statements (R) $a_1 \text{ REL } a_2 \text{ in CX (A,d) where REL = GEN, SPEC, SIM, or DIS}$ bird GEN robin in CX (birds, all characteristics) chicken SPEC fowl in CS (birds, biological characteristics) duck SIM goose in CX (birds, habitat) duck DIS goose in CX (birds, neck length)</p> <p>Mutual implications (I) $d_1(a) = r_1 \iff d_2(a) = r_2$ temperature(place) = warm & rainfall(place) = heavy \iff grain(place) = rice</p> <p>Mutual dependencies (D) $d_1(a) \xrightarrow{+} d_2(a)$ average temperature (place) $\xrightarrow{-}$ latitude (place)</p>
--

Fig. 8 Knowledge representation in HPR [2]

There are also uncertainty parameters which are related to each element in this representation. A brief summary of these parameters are provided in Fig. 9.

<p>α Conditional likelihood that the right-hand side of a dependency or implication has a particular value (referent) given that the left-hand side has a particular value. Applies to dependencies and implications.</p> <p>β Conditional likelihood that the left-hand side of a dependency or implication has a particular value given that the right-hand side has a particular value. Applies to dependencies and implications.</p> <p>γ Degree of certainty or belief that an expression is true. Applies to any expression.</p> <p>τ Degree of typicality of a subset within a set (e.g., robin is a typical bird and ostrich is an atypical bird). Applies to GEN and SPEC statements.</p> <p>σ Degree of similarity of one set to another set. Applies to SIM and DIS statements.</p> <p>ϕ Frequency of the referent in the domain of the descriptor (e.g., a large percentage of birds fly). Applies to any nonrelational statement.</p> <p>δ Dominance of a subset in a set (e.g., chickens are not a large percentage of birds, but are a large percentage of barnyard fowl). Applies to GEN and SPEC statements.</p> <p>μ_r Multiplicity of the referent (e.g., many minerals are produced by a country like Venezuela). Applies to any nonrelational statement.</p> <p>μ_a Multiplicity of the argument (e.g., many countries produce a mineral like oil). Applies to any nonrelational statement.</p>

Fig. 9 Parameters related to elements in HPR’s knowledge representation [1]

There are two problems to keep in mind when we talk about transferring ontological relations to HPR relations. First which patterns among Ontological relations can be mapped to HPR relations and second how can we find the parameters which are needed for a relation in HPR. We explore each relation in HPR in more detail and mention the ways for covering these problems.

B. Simple Statements

This is the first kind of knowledge in HPR which represents some relations mentioned by a “descriptor” between an “argument” and some “referent”.

Statement is mainly a relation between three concepts playing the roles “descriptor”, “argument” and “referent”, e.g. “means-of-locomotion”, “flying” and “birds”. Therefore we need to use more than one ontological relation between

concepts to make a statement. We've proposed three patterns between concepts that can be mapped to statements.

a) Pattern 1

This pattern is depicted in Fig. 10 and its example in Fig. 11. This figure shows an "Argument" which has "Descriptor" as one of its features. The "Descriptor" also has some "Referents" as instances. In fact there is "IS-A" relation between this instances and the "Descriptor". Now what we can infer from this pattern can be expressed by a simple statement as we see below the pattern.

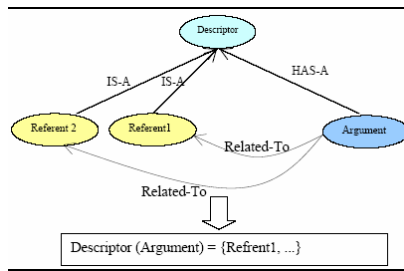


Fig. 10 First pattern for finding statements

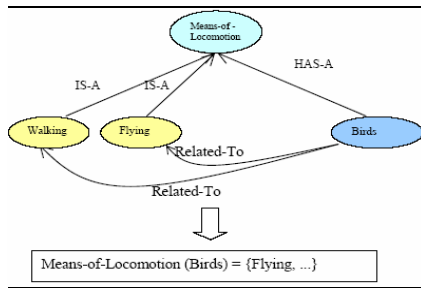


Fig. 11 Example for Pattern 1

b) Pattern 2

This pattern is shown in Fig. 12. The pattern is quite similar to previous pattern, while some differences exist. There may be situations which we can't find the "Has-A" relationship between the descriptor itself and our argument. Instead what we find is a "Has-A" relation between the argument and referent. Fig. 2 gives an example.

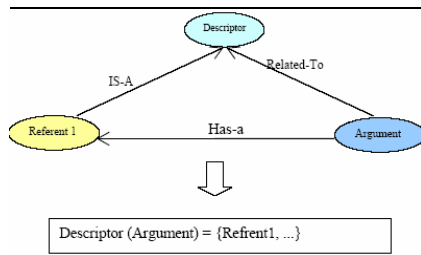


Fig. 12 Pattern 2 for extracting statements

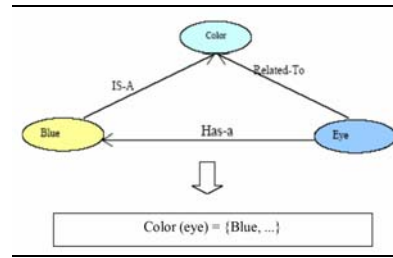


Fig. 13 Example for pattern 2

c) Pattern 3

We can use non-taxonomic relations for constructing simple statements by some patterns. You can see our proposed pattern in Fig. 14. This pattern considers a "Relation" between an "Argument" and a "Referent". Also we consider an IS-A relation founded between the "Referent" and another concept which is shown by "Parent of Referent".

For making the statement we place the "Relation" label beside the "Parent of Referent" to produce a new concept of "Relation-Parent of Referent". Then we add this concept to our ontology and also use this new concept as a descriptor. The resulted statement is shown beneath the pattern. An example is given in Fig. 15.

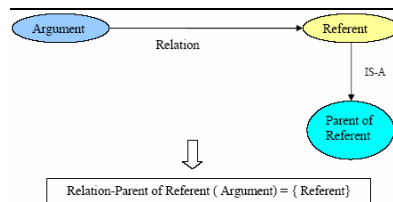


Fig. 14 Pattern 3 for extracting statements

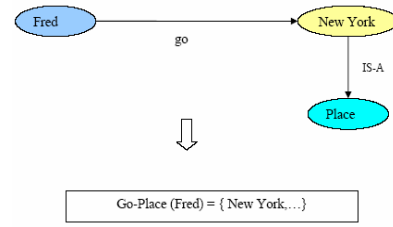


Fig. 15 Example for Pattern 3

d) Parameters

The remained discussion about statement mapping is about the parameters which should be calculated for each statement in HPR.

γ : This parameter shows the confidence degree of a statement. Currently, we compute this parameter based on the simple average of confidence of each relationship from which statement is made. In section III we have shown how confidence factor of each relationship in the pattern is computed.

μ_a : This one shows the multiplicity of argument in a statement which refers to the percent of arguments which can be placed instead of current argument while statement remains valid. We should compute the percent of valid arguments between their sibling arguments. The sibling arguments are those having the *descriptor* as a common feature (there exists

“Has-a” relation with *descriptor*).

Therefore we can measure this parameter by considering the number of all statements which have the descriptor and referents in common with different arguments and divide it by the number of all sibling arguments of current arguments as we see in following formula:

$$\mu_a = \frac{|Current Arguments|}{|Sibling Arguments|} \quad (11)$$

μ_r : This is very similar to previous one but applies over referents. We measure this parameter by considering the number of referents for a specified descriptor and argument and dividing that by the number of sibling referents for the current referent. By sibling arguments we mean concepts which have the descriptor as their common parent in “IS-A” relationship. It is shown in following formula:

$$\mu_b = \frac{|Current referents|}{|Sibling referents|} \quad (12)$$

ϕ : As the definition suggests this is the frequency of referents in the domain of arguments. Unfortunately we couldn't find clue for measuring this parameter from the text mainly because it is statistical information.

C. Relational Statements

The second kind of expression involves one of four relations: generalization (GEN), specialization (SPEC), similarity (SIM), and dissimilarity (DIS) which you can see in figure 8. Each relational statement specifies a context (CX) which we can compute related parameters based on that. Although most HPR based system doesn't use it so we will ignore it for simplicity

Following this section we will investigate each of these four types of *Relational statement* and how we can extract them from middle ontology.

1. GEN and SPEC Relations

We consider both of GEN and SPEC relation as the same and delegate them by just SPEC relationship. The reason is that these relations are essentially reverse of each other.

a) Pattern 4

We suppose that “IS-A” relations can be transformed directly to “SPEC” relations.

b) Parameters

Now we focus on the parameters which should be measured for SPEC relations. Those parameters are measured as follows:

γ : This parameter shows the confidence value for a relational statement. This parameter of “SPEC” relations is the same confidence value which we measured by the formula (6).

δ : this parameter indicates the dominance of a child between other children of a more general concept. We measure the dominance with respect to the covering percentage of that child on the set of children. This can be computed through following formula:

$$\delta(GC, Ci) = \frac{T(GC, Ci)}{\sum_{\forall Cj, Cj \text{ SPEC } GC} T(GC, Cj)} \quad \forall Ci, Ci \text{ SPEC } GC \quad (13)$$

Where $T(GC, Ci)$ is the total number of documents in which Ci is matched with GC within one of pattern of Fig. 2. Through this formula we can measure how frequent a concept is compared to its siblings.

τ : Typicality is the last parameter to measure. That refers to how much a subset is typical comparing with its siblings. For example speaking about birds, you may imagine robins and certainly not a penguin.

We assume that the greater distribution of a concept indicates that the concept is more common. Also, we should consider how much certainty exists about the “SPEC” relation because it wouldn't be a good situation if our certainty of “SPEC” relation is low and its typicality just based on distribution is high! Therefore the following formula is a good candidate for typicality parameter:

$$\tau(GC, C) = A(\gamma(GC, C)) + B(\delta(GC, C)) \quad (14)$$

Where the values for A and B would be selected based on the corpus. Here the $\gamma(GC, C)$ is the confidence and the $\delta(GC, C)$ shows the distribution.

2. SIM Relations

a) Pattern 5

Next kind of rational statements are SIM relations. As we've seen before in section III, we measure the similarity between two concepts based on their common features. We use that directly here.

b) Parameters

γ : Unfortunately we can't yet determine this parameter.

σ : This parameter indicates how much two concepts are similar. We use the formula (10) from previous section to measure this parameter.

D. Mutual Dependency

This kind of relation relates two terms of “Descriptor(Argument)” to each other as shown for example in Fig. 8. Therefore this relationship can effect greatly on HPR inferences.

We use two kinds of ontological relationship extracted in previous phase to achieve mutual dependency relationship. The first relation is the causal relations and the second one is the association rules. We will explore them in the two last following patterns.

a) Pattern 6

This pattern considers using causal relations as the source of mutual dependency relations. In fact, one option is to use this relation directly. In this case we just consider two concepts $C1$ and $C2$ which are in fact two descriptors with no arguments. Although this is not a complete form of a mutual dependency but it can yet serves as a source of knowledge in

HPR systems like TelQAS [18] because this systems add argument information through the context of a user query.

Our proposed method for adding arguments into this relation is based on an assumption. We assume probable argument set is a subset of concepts which have C1 and C2 as their two features (exists Has-a relationship). For example in a relation like “Temperature (place) \leftrightarrow altitude (place)” both “Temperature” and “altitude” are features of “Place”.

We mine associations like “(C1, C2) \Rightarrow H)” in which “C1” and “C2” are both the attributes of “H”. We can choose associations with confidence over a threshold.

Of course this is not a complete method because we miss relations with different arguments in each side.

b) Pattern 7

Other ontological source of mutual dependency beside causal relations is “Related-To” relationships extracted through association rule mining. These relations just determine an unknown relationship between two concepts. However we can use them to represent a weak mutual dependency relationship.

c) Parameters

The parameters to measure are as following:

γ : This parameter displays the certainty of a relation as before. We consider two cases based on the source of a mutual dependency i.e. causal relations or association rules.

Causal relation detecting algorithm as shown before has a ranking mechanism which determines how much ambiguity a detected relation has. We use this mechanism to assign a confidence factor to each mutual dependency relation.

The second case is using “Related-To” relation as the source of mutual dependency relation. In this case we use the confidence factor borrowed from association rules as γ parameter of a mapped mutual dependency relation.

α , β : This two parameters are conditional likelihood of a mutual dependency which show how much one side of relation affects the other side. Each of these parameters belongs to one direction in a mutual dependency relationship.

Unfortunately, we can not yet measure these two parameters. This is mainly because no clue in ontological relations shows the degree of effect.

E. Non-Detected Relations

This architecture can't yet extract two kinds of relationship which are *Dissimilarity Relationship (DIS)* and *Mutual implication*. This is because we have found no appropriate algorithm yet.

V. FUTURE WORKS

The proposed architecture suffers from some limitations that can be mitigated in future refinements. First is about the limit number of algorithms used in Ontology Builder subsystem. We can add more algorithms to this set and expect more exact results. The Second one is about some missed algorithms for detecting relations like dissimilarity and axioms

which could be added.

The third and most important point is about evaluation. We haven't yet considered methods for evaluating the extracted relationships. This is a challenging task because there is not an agreed evaluating method like precision and recall in Information Retrieval community.

REFERENCES

- [1] A.Collins, R. Michalski, The Logic Of Plausible Reasoning A Core Theory, Cognitive Science, Vol. 13, pp. 1-49. 1989.
- [2] M. Burstein, A. Collins and M. Baker. Plausible Generalization: Extending a Model of Human Plausible Reasoning Journal of the Learning Sciences, Vol. 1, No. 2., 1992.
- [3] H.Cunningham, Y. Wilks and R. Gaizauskas. GATE -- a General Architecture for Text Engineering. In Proceedings of the 16th Conference on Computational Linguistics (COLING96), 1996
- [4] P. Cimiano, J. Volker, Text2Onto, a Framework for Ontology Learning and Data-driven Change Discovery, In Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB'05). 2005
- [5] P. Velardi, R. Navigli, A. Cuchiaroli, F. Neri, Evaluation of OntoLearn, a Methodology for Automatic Learning of Domain Ontologies, In Ontology Learning from Texts: Methods, Evaluation and application, IOS Press.2005
- [6] R. Navigli and P. Velardi. Learning Domain Ontologies from Document Warehouses and Dedicated Websites. Computational Linguistics, 30(2), 2004
- [7] M.A. Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. In proceedings of the 14th international conference on computational linguistics, 1992.
- [8] E. Charniak and M. Berland. Finding parts in very large corpora. In Proceedings of the 37th Annual Meeting of the ACL, 1999.
- [9] P. Cimiano, A. Pivk, L. Schmidt-Thieme, and S. Staab. Learning taxonomic relations from heterogeneous evidence. In Ontology Learning from Text: Methods, Applications and Evaluation. IOS Press, 2005.
- [10] Hearst. Untangling text data mining. In Proceedings of the ACL'99: the 37th Annual Meeting of the Association for Computational Linguistics. University of Maryland, 1999
- [11] D. Faure and C. N'edellec. ASIUM: Learning subcategorization frames and restrictions of selection. In 10th Conference on Machine Learning -- Workshop on Text Mining, Chemnitz, Germany, 1998.
- [12] R. Girju and D. I. Moldovan. Text mining for causal relations. In FLAIRS 2002
- [13] F. Oroumchian, R.N. Oddy, An Application of Plausible Reasoning to Information Retrieval, ACM's SIGIR 1996
- [14] M. Karimzadegan, F. Oroumchian, J. Habibi, XML Information Retrieval by Means of Plausible Inferences. In Proceedings of the 5th International Conference on Recent Advances in Soft Computing, 2004.
- [15] F. Oroumchian, B.N. Araabi, E. Ashoori, An Application of Plausible Reasoning and Dempster-Shafer Theory of Evidence in Information Retrieval. In proceeding of International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2002.
- [16] F. Oroumchian, B. Khandzah, Modeling an Intelligent Tutoring System by Plausible Inferences. In proceeding of International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), 2002.
- [17] A. Jalali, F. Oroumchian, An Evaluation of Document Clustering by means of Plausible Inferences. International Journal of Computational Intelligence, 2004.
- [18] E. Darrudi, M. Rahgozar, F. Oroumchian, Human Plausible Reasoning for Question Answering Systems. In proceeding of Advances in Intelligent Systems - Theory and Applications. Luxembourg, 2004.