International Journal of Electrical and Computer Engineering (IJECE)
Vol. 5, No. 2, April 2015, pp. 318~332
ISSN: 2088-8708

Ontology-based Why-Question Analysis Using Lexico-Syntactic Patterns

A.A.I.N. Eka Karyawati, Edi Winarko, Azhari, Agus Harjoko
Department of Computer Science and Electronics, Gadjah Mada University, Yogyakarta, Indonesia

Article Info

Article history:

Received Jan 2, 2015
Revised Feb 13, 2015
Accepted Feb 27, 2015

Keyword:

Lexico-syntactic pattern
Nlp-based text mining
Question analysis
Question answering system
Typed dependency parse
Why-question

ABSTRACT

This research focuses on developing a method to analyze why-questions. Some previous researches on the why-question analysis usually used the morphological and the syntactical approach without considering the expected answer types. Moreover, they rarely involved domain ontology to capture thesemantic or conceptualization of the content. Consequently, some semantic mismatches occurred and then resulting not appropriate answers. The proposed method considers the expected answer types and involves domain ontology. It adapts the simple, the bag-of-words like model, by using semantic entities (i.e., concepts/entities and relations) instead of words to represent a query. The proposed method expands the question by adding the additional semantic entities got by executing the constructed SPARQL query of the why-question over the domain ontology. The major contribution of this research is in developing an ontology-based why-question analysis method by considering the expected answer types. Some experiments have been conducted to evaluate each phase of the proposed method. The results show good performance for all performance measures used (i.e., precision, recall, undergeneration, and overgeneration). Furthermore, comparison against twobaseline methods, the keyword-based ones (i.e., the term-based and the phrase-based method), showsthat the proposed method obtainedbetter performance results in terms of MRR and P@10 values.

Corresponding Author:

A.A.I.N. Eka Karyawati,
Lecturer Staff of Department of Computer Science, Faculty of Mathematics and Natural Sciences,
Udayana University, Bali, Indonesia
Doctoral Student of Department of Computer Science and Electronics,
Faculty of Mathematics and Natural Sciences, Gadjah Mada University, Yogyakarta, Indonesia.
Email: eka.karyawati@mail.ugm.ac.id

1. INTRODUCTION

A question analysis is a process to analyze a natural laguage question in order to convert the question into a formal query representation. The formal query representation is constructed so that the information contained in the question can be processed by a question answering system. The question analysis is a fundamental component of a question answering system because the query representation represents a user information need. Thus, the systemwill result accurate answers (i.e., satisfy the information need), if the user information need can be represented accurately.

This research focuses on developing a method to analyze a why-question (i.e., a why-question analysis method). According to the Aristotle philosophy, there is a close relation between understanding and why-question [1]. Human do not think understand something until they grasp the why of it. On the other words, it is necessary to know the answer of the why-question in order to understand something. It is the reason why developing a method to analyze a why-question is important. The good method of a why-question analysis will result accurate answers, hence users can understand something accurately.

Why-question is a question preceded by a why-question word and followed by a topic of the question. For example, for a question, "Why is a vector space model used in information retrieval?", the topic of the question is “A vector space model is used in information retrieval”. Actually, the why-question analysis needs to determine answer types. Verberne stated that it is necessary to split the answer type of a why-question into sub-types, for getting more specific answer type in order to select potential answer sentences or paragraphs [2]. However, there are still few researches studying on why-question analysis by considering the expected answer types.

To get more accurate answers, a question analysis method also should involve semantic approach to capture the conceptualizations associated with user information needs and contents. Nevertheless, most of the question analysis approaches only analyze syntactically and morphologically without considering the semantic of the question content [3]-[5].

Based on the above facts, a research problem is formulated, that is how to analyze a why-question by considering the expected answer types, and by utilizing domain ontology in order to capture the conceptualization of the question content. Thisresearch focuses on developing a method using the combination of part-of-speech (POS) tagging, typed-dependency parsing, verb classification, and domain ontology. Some researches have been used domain ontology to formulate a query, and to capture the conceptualization of the query content [6], [7], [9]-[13], but they did not focus on why-questions.

The proposed method is performed by utilizing lexico-syntactic patterns employed over typed dependency parse trees. The typed dependency parsing is used because the dependencies or relations between elements of a sentence are clearly defined. Therefore, typed dependency parsing together with POS tagging can be used more easily to construct the patterns used for extracting terms and relations of a why-question. Furthermore, by considering the verb classification, the lexico syntactic patterns can also be used to identify the expected answer types of the why-question. The proposed method adapts the simple, the bag-of-words like model, by using semantic entities (i.e., concepts/entities and relations) instead of words to represent a formal query representation. In addition, the proposed method expands the question by adding the additional semantic entities got by executing the constructed SPARQL query of the why-question over domain ontology.

Thus, the major contribution of this research is in developing an ontology-based why-question analysis method using the lexico-syntactic patternsby considering the expected answer types. The method uses OWL for building ontology, saves the data using RDF format, and uses SPARQL for query processing. In SPARQL construction, the proposed method considers two answer types of the why-question includingthe cause answer type, andthe motivation answer type. Some experiments have been conducted to evaluate each phase of the proposed method, by using some evaluation measures such as the precision, the recall, the undergeneration, and the overgeneration measure [8]. The results show good performance for all performance measures used. Furthermore, the comparisons against two baseline methods, show the proposed method obtained better performance results (i.e., in terms of MRR and P@10 values [35], [36], [37]) than both baseline methods, the keyword-based ones (i.e., the term-based and the phrase-based method).

The main assumption used in this research is the questions must be in correct English grammar. Other assumptionis the terms and the relations that are queried are restricted in the specific scope, because the implementation of the proposed method is in a specific domain (e.g., text retrieval domain).In addition, the questions asked should have the patterns already defined. As a result, the proposed method will show the best performance under those conditions.

The rest of this paper is organized as follows. Section 2 presents works related to this research. The theoreticalbasis and the proposed why-question analysis method are given in section 3 and section 4, respectively. Section 5 presents the research method. Discussions about the resultsare presented in Section 6. Finally, conclusions are given in Section 7.

2. RELATED WORK

The proposed why-question analysis method involves the domain ontology to grasp the conceptualization of the question contents through identifying the semantic annotations. Consequently, the question analysis adapts an ontology-based question answering model involving three main components, term/relation extraction, semantic entity mapping, and formal query construction. Most of the ontology-based question answering method used linguistic approach for extracting terms and relations [9]-[13]. Moreover, for semantic entity mapping, some researchers used string similarity matching and Wordnet [10], [11], [13]. Similar to the previous works, the proposed method uses linguistic approachfor extracting terms and relations and uses string similarity matching for semantic mapping. However, the proposed method does not employ a general domain lexicon (e.g., Wordnet), it employs a list of synonymies (i.e., a specific domain lexicon) instead. Furthermore, for constructing the formal queries that are compliant with the domain ontology, most

of researches used the triple-based data representations that are referred to as the query-triples and used OWL to build the ontology and saved the data using RDF format [9]-[12], [14]. Moreover, they employed SPARQL (i.e., SQL-like query language and suitable for accessing data in RDF format) to perform the query processing. The proposed method also builds the domain ontology using OWL format, saves the knowledge base in RDF format (i.e., triple-based representation), and uses SPARQL for query processing. It is for some practical reasons.

The proposed method uses NLP-based text mining for extracting terms and relations. The NLP-based text mining is performed through employing some patterns (i.e., lexico-syntactic patterns) on parse trees to generate structural representation of free text. In this research, the lexico-syntactic patterns are constructed over the typed dependency parse trees. The lexico-syntactic patterns have been widely used by researchers for extracting information (i.e., terms and relations) from sentences of a free text. Kim employed hand-crafted patterns on typed-dependency parses [14] to identify terms and relations. On the other hand, Zouaq combined POS tagging andtyped-dependency parsingto employ lexico-syntactic patterns in order to extract terms and relations contained in a sentence [15], [17]. Moreover, some other researches employed patterns on dependency parse trees to extract terms and relations from natural language text [16]-[18]. On the other hand, Mousavi applied NLP-based text mining through employing some patterns on phrase structure parse trees to generate structural representation of free text [19]. In contrast to the previous researches that focused on the free texts representations (i.e., not questions), the proposed method focuses on why-question representations instead.

3. THE THEORETICAL BASIS

3.1. Definitions

Definition 1 (Typed Dependency Parse) Typed dependency parse is a kind of dependency parse that represents dependencies between words and labels the dependencies by grammatical relations [20].

Definition 2 (Action Verb) The action verbs are verbs that express an action. Action means something happening or something changing. Most action verbs refer to physical actions, but some are verbs of reporting or verbs of thinking [21]. Examples of the action verbs are ‘use’, ‘utilize’, ‘employ’, ‘apply’, ‘perform’, and others.

Definition 3 (Process Verb and Causative/Inchoative Alternation) The process verbs are verbs that express a process. In this context, process means change of state or change of position. On the other hand, the causative/inchoative alternation is a transitivity alternation where the transitive use of a verb V can be paraphrased as roughly “cause to V-intransitive” [22]. Moreover, verbs under going the causative/inchoative alternation can be characterized as verbs of change of state or change of position. Thus, the process verbs are verbs from the causative/inchoative alternation, especiallyin an intransitive context. Example of the process verbs are ‘appear’, ‘arise’, ‘occur’, ‘happen’, ‘change’, ‘compress’, ‘collect’, ‘improve’, ‘increase’, and others.

Definition 4 (Edit Distance) Edit distance is defined as the minimum number of edit operations to transform one string into the other. Two prevalent edit distance algorithms are the Levenstein distance [23], and the Damerau-Levenshtein distance [24]. The Levenshtein distance defines edit operations as insertions, deletions, and substitutions. The Damerau-Levenshtein distance is a variation of the Levenstein distance with the additional operationof transposition.

Definition 5 (Domain Ontology) Domain ontology is anexplicit specification of a conceptualization about domain knowledge [25]. It can be described as $O = (C,R)$, where C is the set of concepts, and R is the set of semantic relationships between concepts.

Definition 6 (SPARQL Query) A SPARQL query is based around graph pattern matching, where the graphs are RDF graphs [26]. More complex graph patterns can be formed by combining smaller patterns including basic graph patterns, group graph pattern, optional graph patterns, alternative graph pattern, and patterns on named graphs.

Definition 7 (RDF Graph, Basic Graph Pattern, and Alternative Graph Pattern) An RDF graph is a set of RDF triples $(s, p, o) \in (I \cup B) x I x (I \cup B \cup L)$, where I, B, and L are infinite sets IRIs, Blank nodes, and Literals, respectively [26]. In this triple, s is the subject, p the predicate, and o the object. A basic graph pattern is a set of triple patterns, where a triple pattern is a triple $(s, p, o) \in (I \cup V) x I x (I \cup V \cup L)$[26]. V is a set of variables disjoint from sets I, B, and L. A question mark? in a triple indicates that v is a variable. In an alternative graph pattern, two or more possible basic graph patterns are tried.

3.2. Characteristic of Natural Language Question

This research addresses some basic characteristics of a natural language question, including expected answer type, question topic, question terms, and relations [9]. Two expected answer type of why-questions that have been already observed arethe cause, and the motivation answer type. Question topic is a declarative sentence following the 'why'-question word, from which the question terms and relations will be extracted. Question terms can be a word or multiple words (i.e., noun phrases), that are focuses of the why-question, and identified as concepts/instances. Relations often are verbs of the topic question. The concepts/instances and relations will be used to construct a set of intermediate representations of the why-question. The representations employ triple-based representations referred to as query-triples.

3.3. Expected Answer Type of a Why-Question

A why-question is a question answered by a cause [1]. Furthermore, there are four types of causes (i.e., Aristotle four causes) including, the material, the formal, the efficient, and the final cause. The material cause is about what a thing is made of, the formal cause about its form or what it is, the efficient cause about who made it or how it came to be what it is, and the final cause about what a thing is made for or what its purpose is [1], [27]. Alvarez stated that the efficient cause is what most people think of as cause [27]. On the other words, the efficient cause is relating to the reason clause, and the final cause relating to the purpose clause. This research concerns in these two cause types.

According to Quirk, there are four types of the reason clause, including the cause-effect, the reason-consequence, the motivation-results, and the circumstances-consequence clause [28]. In addition, because the result relation in the result clause is the converse of that of motivation [28], the result clause is also considered in this research. However, the circumstances-consequence clause is not taken into account, because it seldom arises in a why-question collection. Thus, the proposed method observes two expected answer types of a why-question, which are the cause answer type and the motivation answer type. These types involve fiveclauses, where the cause answer type relates to the cause-effect clause, and the motivation answer type relates to the reason-consequence, the motivation-results, the result, and the purpose clause.

3.4. NLP-Based Text Mining

The proposed method uses NLP-based text mining to extract terms and relations. The NLP-based approach considers the morphological structure by parsing the sentences into parse tree [19]. Parse tree provides a morphological structure for text analyzing. Text mining through NLP-based technique is usually performed by employing lexico-syntactic patterns. The proposed method constructs the lexico-syntactic patterns using the combination of POS tagging, typed dependency parsing, verb classification, and ontology.

One of the popular typed dependency parses is Stanford typed dependency. The Stanford typed dependencies are generated from phrase structure parse trees through two-phase method, including the dependency extraction and the dependency typing phase [20]. The dependency extraction phase extracts dependencies by applying rules (i.e., Collin head rules [29]) on phrase structure trees. Furthermore, the dependency typing phase labels the dependencies with a grammatical relation which is as specific as possible. The identification of which grammatical relation used to label the dependencies is based on the patterns (i.e., over the phrase structure parse tree) defined using a tree-expression syntax, where the tree-expression syntax is defined by tregex [30].

4. THE PROPOSED ONTOLOGY-BASED WHY-QUESTION ANALYSIS METHOD

As can be seen in Figure 1, the proposed method includes three main components, which are the term/relation extraction that has main goal to extract terms and relations contained in a why-question in order to construct an intermediate representation (i.e., query-triples), the semantic mapping that has main goal to map between the extracted terms and relations into semantic entities (i.e., ontological elements of the domain ontology) in order to identify semantic annotations of the original query and to construct ontology-compliant query-triples, and the SPARQL query construction and processing that has goal to construct a SPARQL query of the why-question, and then to process the query over the knowledge base in order to identify the additional semantic annotations. The query expansion expands the original semantic annotations using the additional ones, where semantic annotations of a question is defined as a set of semantic entities (i.e., ontological elements including concepts/instances and relations) used to represent a question.

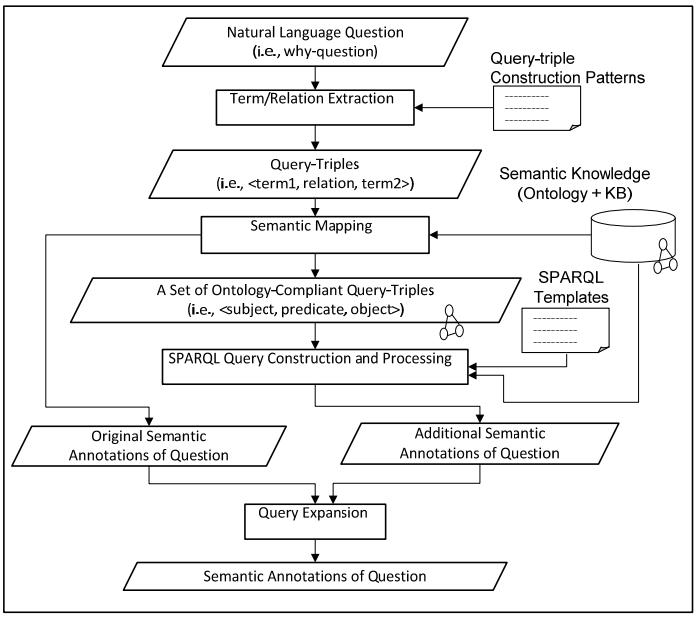

Figure1. Graphical representation of the proposed ontology-based why-question analysis method

4.1 Terms and Relations Extraction

The proposed question analysis method employs patterns (i.e., named as query-triple construction patterns) represented using the convention,

'*Gramatical Relation* (*Head-Index*/*POS*, *Dependent-Index*/*POS*) → *Transformation*',

Where *Grammatical Relation* represents a dependency relation, *Head* and *Dependent* are variable names, *POS* represents a part-of-speech, *Index* represents the position of the word in the sentence, and *Transformation* describes the resulting expression [15]. The proposed method employs Stanford POS tagging for tagging a word, and Stanford parser for constructing typed dependency parse trees. Table 1 shows examples of the lexico-syntactic patterns for identifying noun phrases (i.e., terms). Moreover, Table 2 shows examples of the lexico-syntactic patterns for extracting relations. As a note, Agent is agentive noun (e.g., we, researcher, user, and others), NN is POS for all noun phrases (i.e., NN, NNS, NNP, NNPS), and VB is POS for all verbs (i.e., VB, VBZ, VBD, VBG, VBN, VBP). All POS labels can be seen in [31]. Furthermore, all dependency relation labels can be seen in [32].

Table 1. The lexico-syntactic patterns for identifying concepts

Lexico-Syntactic Pattern	Example
nn(X/NN, Y/NN) → Y_X	nn(retrieval-2/NN, information-1/NN) →information_retrieval
nn(X/NN, Y/NN), nn(X/NN, Z/NN) → Y_Z_X	nn(model-4/NN, vector-2/NN), nn(model-4/NN, space-3/NN) →vector_space_model
amod(X/NN, Y/JJ) → Y_X	amod(system-5/NN, smart-4/JJ) →smart_system
amod(X/NN, Y/JJ), nn(X/NN, Z/NN) → Y_Z_X	amod(engine-4/NN, semantic-2/JJ), nn(engine-4/NN, search-3/NN) →semantic_search_engine

Table 2. The lexico-syntactic patterns for extracting relations

Lexico-Syntactic Pattern	Example	
	Question Topic	Relation Extraction
nsubj(X1/VB, X2/Agent), dobj(X1/VB, X3/NN), prep(X3, X4) → Be_V3(X1)_Prep(X3, X4)	We use a vector_space_model for text_retrieval.	nsubj(use-2/VB, We-1/PRP), dobj(use-2/VB, vector_space_model-4/NN), prep_for(vector_space_model-4/NN, text-retrieval-6/NN) →**is_used_for**(vector_space_model, text_retrieval)
nsubj(X1/JJ, X2/NN), cop(X1/JJ, X3/VB) → hasQuality(X2, X1)	A vector_space_model is useful.	nsubj(useful-4/JJ, vector_space_model-2/NN), cop(useful-4/JJ, is-3/VB)→**hasQuality**(vector_space_model, useful)
expl(X1/VB, there/EX), nsubj(X1/VB, X2/NN), prep_in(X2/NN, X3/NN) → occur_in(X2/NN, X3/NN)	There are word_mismatches in search_engine.	expl(are-2/VB, there-1/EX), nsubj(are-2, word_mismatches-3/NN), prep_in(word_mismatches-3/NN, search_engine-5/NN) →**occur_in**(word_mismatches, search_engine)

4.2 Expected Answer Type Identification and Query-Triple Construction

There are two answer types of a why-question concerned in this research including the cause, and the motivation answer type, where their identification is based on verb classification. A question has the cause answer type, if the main verb of the question is classified as the process verb. In addition, a question also has a cause answer type if the main verb is an affect verbs such as 'affect', or 'influence', a verb with feature 'intens', an existential "there" question, a question that has subject complement 'adjective phrase', a question that has modal auxiliary 'can/could' or 'have/has to'.On the other hand, a question has the motivation answer type, if the main verb of the question is classified as the action verbs. Moreover, a question also has the motivation answer typeif the main verb is need verbs such as 'need', or 'require', the main verb is consider verbs such as 'consider', or 'take-into-account', and the question that has modal auxiliary 'shall/should'. Some ideas of the identification are got from [2].

Moreover, the lexico-syntactic patterns are used as a basis for constructing SPARQL templates. Thus, the patterns involveelements of the domain ontology. In SPARQL template construction, the proposed method considers thetwo answer types involving causalities. Consequently, the domain ontology is designed so that the causalities can be easily detected. Even though, there are five clauses concerned (see sub-chapter 3.3), but all of them are represented in two relation representations, the *cause* relation (i.e., for the cause-effect, the reason-consequence, the motivation-result, and the result clause), and the *purpose* relation (i.e., for the purpose clause). The causality representations are designed by involving *has Component* relation, where the *has Component* relation separates a sentence into some sentence components (i.e., terms that are noun phrases, and relations that are usually verbs) that refer to as semantic entities of a sentence. It is suitable to thegoal of the why-question analysis method, that is to identify semantic annotations (i.e., refer to semantic entities) of a question.

Table 3 shows representation of the causalities in the knowledge base of the domain ontology. In this case, X is referred to as a question topic, and Y as the answer. For example,a question, "why X?",may has answer "because Y" (i.e., the cause relation, *Y cause X*) or "in order to Y" (i.e., the purpose relation, X *has Purpose* Y), depending on the answer type of the question.If a question has the cause answer type, the question only has answer containing the cause relations. In the other hand, if the question has the motivation answer type, the question may have answers containing both the cause and the purpose relations.

Table 3. The representation of causality in domain ontology

Cause Relation	Purpose Relation
Y cause X; X hasComponent A1; XhasComponent A2; ... Y hasComponent B1; Y hasComponent B2; ...	X hasPurpose Y; X hasComponent A1; X hasComponent A2; ... Y hasComponent B1; Y hasComponent B2; ...

Together with the term/relation extraction patterns, the causality representations are used for constructing the query-triples of a why-question. The query-triples construction is performed in two steps. First, the terms of the why-question are extracted using the term extraction patterns (see Table 1), and then, after getting the extracted terms, the relation extraction patterns (see Table 2), the verb classification (i.e., identifying expected answer type), and the causality representations (see Table 3) are employed together for constructing query-triples. Examples of the patterns for constructing query-triples of a why-question are shown in Table 4 and Table 5.

Table 4. The lexico-syntactic patterns for constructing query-triples

Lexico-Syntactic Pattern (left-hand-side of pattern, LHS)	Answer Type	Query-Triples (right-hand-side of pattern, RHS)
nsubj(X1/VB, X2/Agent), dobj(X3/VB, X3/NN), prep(X3/NN, X4/NN), X3 is an action verb	Motivation	Be_V3(X1)_Prep(X3, X4); cause(A1, Q); hasPurpose(Q,A2); hasComponent(Q, X3); hasComponent(Q, X4)
nsubj(X1/JJ, X2/NN), cop(X1/JJ, X3/VB), X1 is an adjective	Cause	hasQuality(X2, X1), cause(A, Q), hasComponent(Q, X1); hasComponent(Q,X2)
expl(X1/VB, there/EX), nsubj(X1/VB, X2/NN), prep-in(X2/NN, X3/NN), an existential "there" question	Cause	occur_in(X2, X3), cause(A, Q), hasComponent(Q, X2); hasComponent(Q,X3)

Table 5. Examples of query-triples construction

Question Topic	Lexico-Syntactic Pattern (LHS)	Answer Type	Query-Triples(RHS)
We employ a vector_space_model for text_retrieval.	nsubj(employ-2/VB, We-1/PRP), dobj(use-2/VB, vector_space_model-4/NN), prep_for(vector_space_model-4/NN, text_retrieval-6/NN), 'employ' is an action verb	Motivation	is_employed_for(vector_space_model, text_retrieval); cause(A1, Q); hasPurpose(Q,A2); hasComponent(Q, vector_space_model); hasComponent(Q, text_retrieval)
A vector_space_model is useful.	nsubj(useful-4/JJ, vector_space_model-2/NN), cop(useful-4/JJ, is-3/VB) 'useful' is an adjective	Cause	hasQuality(vector_space_model, useful), cause(A, Q), hasComponent(Q, useful); hasComponent(Q,vector_space_model)

4.3 Semantic Mapping, SPARQL Construction, and Semantic Annotation

In this research, semantic mapping is performed in two main phases, first the extracted terms and relations are matched with all labels defined in domain ontology by using edit distance, and then they are mapped into semantic entities (i.e., object properties, and instances) of the domain ontology. Some researchers used Wordnet as a lexical resource. However, the use of some general domain lexical resources, such as WordNet, would not be practicable because they will discard several terms belonging to the specific domain. Thus, the proposed method employs manually lists of synonymies of terms and relations instead of Wordnet as a specific domain lexicon. In implementation, synonymies are saved as knowledge base in RDF format, where each instance and relation (i.e., object property) has list of synonymy saved as *label* elements. Moreover, the proposed method uses Damerau-Levenstein edit distance because transposition of characters often occurs when users inputs a question. Semantic entities of a why-question will be used to identify the semantic annotations of the original query, and to construct ontology-compliant query-triples that are basis of SPARQL construction. Table 6 presents example of ontology-compliant query-triples, where *OP(x)* is object property of label *x*, and *I(y)* is instance of label *y*.

Table 6. Example of ontology-compliant query-triples construction

Expected Answer Type	Query-Triples	Ontology-Compliant Query-Triples
Motivation	Be_V3(X1)_Prep(X3, X4); cause(A1, Q); hasPurpose(Q,A2); hasComponent(Q, X3); hasComponent(Q, X4); hasComponent(Q, Gerund(X1));	OP(Be_V3(X1)_Prep)(I(X3), I(X4)); cause (A1, Q); hasPurpose(Q,A2); hasComponent(Q, I(X3)); hasComponent(Q, I(X4))
Cause	hasQuality(X2, X1), cause(A, E), hasComponent(E, X1); hasComponent(E,X2)	hasQuality(I(X2), I(X1)), cause(A, Q), hasComponent(Q, I(X1)); hasComponent(Q,I(X2))

Table 7 presents examples of a SPARQL template for a why-question that has the motivation answer type. SPARQL queries are constructed by using SPARQL templates. The templates are manually constructed based on the the query-triple construction patterns. A SELECT query form is employed, because it is most suited for representing why-question. To retrieve more potential answers, the proposed method

considers taxonomical relations between concepts. Though the knowledge base does not contain causality of concepts asked in a question, the system still can identify the additional semantic annotations of the question.

For instance, for question, "Why does some word mismatches arise in IR?", even though the knowledge base does not contain causality between concepts WordMismatch and IR (e.g., {<X, cause, WordMismacth>, <WordMismatch, OccurIn, IR>}), the semantic annotations still can be identified if the knowledge base contains causality between concept WordMismatch and sub-concepts of IR, such as TextRetrieval, SearchEngine, and others. Furthermore, the query body of the SPARQL query is obtained by transforming the query-triples into alternative graph pattern for representing the alternative of sub-classes (i.e., sub-concepts) and the alternative of relations included in the motivation answer type, where the motivation answer type includes two relations, the cause and the purpose relation (see sub-chapter 3.3 dan 4.2).

As can be seen in Table 7, the SPARQL template represents the alternative of sub-classes, and the alternative of relations included in the motivation answer type, where Instance1, Instance2, and Gerund are slots for instances of term1 (i.e., concept1), term2 (i.e., concept2), and the present participle of a verb (i.e., relation), respectively. Term1, term2, and verb are extracted from a why-question. Moreover, TR represents the Text Retrieval ontology. After constructing the SPARQL query, the additional semantic annotations are identified by executing the query against the knowledge base of the domain ontology. The semantic annotations are all semantic entities (i.e., instances and object properties) that satisfy the SPARQL query.

Table 7. Example of SPARQL template

Ontology-Compliant Query-Triples	SPARQL Template

relation(Instance1, Instance2);
cause(A1, Q);
hasPurpose(Q, A2)
hasComponent(Q, Instance1);
hasComponent(Q, Instance2)
	SELECT ?instance
	WHERE {
	{ TR:Instance1 TR:relation TR:Instance2. ?A1 TR:cause ?Q. ?QTR:hasComponent TR:Instance1. ?Q TR:hasComponent TR:Instance2. ?A1 TR:hasComponent ?instance }
	UNION{ ?x TR:relation TR:Instance2. ?A1 TR:cause ?Q. ?Q TR:hasComponent ?x. ?QTR:hasComponent TR:Instance2.?x rdf:type ?c1.?c1 rdfs:subClassOf ?c2. TR:Instance1 rdf:type ?c2. ?A1 TR:hasComponent ?instance }
	UNION{ TR:Instance1 TR:relation ?x. ?A1 TR:cause ?Q. ?Q TR:hasComponent TR:Instance1. ?QTR:hasComponent ?x.?xrdf:type ?c1.?c1 rdfs:subClassOf ?c2. TR:Instance2rdf:type ?c2. ?A1 TR:hasComponent ?instance }
	UNION{ ?x TR:relation ?y. ?A1 TR:cause ?Q. ?Q TR:hasComponent ?x. ?Q TR:hasComponent ?y. ?x rdf:type ?c1. ?c1 rdfs:subClassOf ?c2.TR:Instance1rdf:type ?c2.?y rdf:type ?c3. ?c3 rdfs:subClassOf ?c4.TR:Instance2rdf:type ?c4.?A1 TR:hasComponent ?instance }
	UNION{ TR:Instance1 TR:relation TR:Instance2. ?Q TR:hasPurpose ?A2. ?QTR:hasComponent TR:Instance1.?QTR:hasComponent TR:Instance2. ?A2 TR:hasComponent ?instance}
	UNION{ ?x TR:relation TR:Instance2. ?Q TR:hasPurpose ?A2. ?Q TR:hasComponent ?x ?QTR:hasComponent TR:Instance2.?xrdf:type ?c1.?c1 rdfs:subClassOf ?c2. TR:Instance1rdf:type ?c2. ?A2 TR:hasComponent ?instance }
	UNION{ TR:Instance1 TR:relation ?x. ?Q TR:hasPurpose ?A2. ?QTR:hasComponent TR:Instance1. ?Q TR:hasComponent ?x. ?xrdf:type ?c1. ?c1 rdfs:subClassOf ?c2. TR:Instance2rdf:type ?c2. ?A2 TR:hasComponent ?instance }
	UNION{ ?x TR:relation ?y. ?Q TR:hasPurpose ?A2. ?Q TR:hasComponent ?x. ?QTR:hasComponent ?y.?xrdf:type ?c1. ?c1 rdfs:subClassOf ?c2. TR:Instance1rdf:type ?c2. ?y rdf:type ?c3. ?c3 rdfs:subClassOf ?c4. TR:Instance2rdf:type ?c4. ?A2 TR:hasComponent ?instance }
	}

5. RESEARCH METHOD

Developing the proposed method needs some supported data including a question collection, and domain ontology (i.e., ontology schema and knowledge base). The question collection is constructed through three steps, first collecting why-questions (i.e., general domain questions) from web and from Verberne's why-question collection [33], second analyzing the questions to identify general patterns of the why-questions, and third generating why-question in a specific domain (i.e., *Text Retrieval*) using the patterns. As default, the questions are set in well-ordered forms (i.e., the questions have correct English grammar, the patterns have been already defined, and the terms and relations have been already covered).

For domain ontology building, *Text Retrieval* (TR) ontology is defined in order to represent concepts and relations used to construct SPARQL translation of the why-questions. The *Text Retrieval* ontology is also used to identify the additional semantic annotations of the why-questions by executing the

SPARQL query against the knowledge base of the domain ontology. In the *Text Retrieval* ontology, each concept generally has one instance, where each instance has a set of labels as synonymies of the instances. These labels also represent synonymies of the concept. The reason of this is in science domain such as *Information Science* or *Computer Science*, it is difficult to identify instances of a concept. It is different from other domain, for instance *Academic* domain, there is *student* concept that has some instances representing by name of the students. An instance of a concept in the *Text Retrieval* ontology is defined as a termappearing in the information sources of the knowledge base (i.e.,papers). Because the terms representing concepts can be in various forms, an instance is labeled in some synonymies defined manually. In addition, relations also are labeled in some synonymies defined manually based on the English thesaurus. Taxonomical relations of the domain ontology use the taxonomy of *Information Retrieval Model* [34] as a starting point. Expansion of the taxonomical relations, identification of non-taxonomical relations, and identification of termsof the domain ontology are performed by learning the terms and relations of *Text Retrieval* domain from IR (i.e., Information Retrieval) textbook [35], [36], and some IR journals.

The proposed method is implemented by using Java programming (i.e., NetBeans IDE). Some API libraries are embedded in the system, such as Stanford parser and Apache Jena.The Stanford parser API (i.e., *stanford-parser.jar*) is used for constructing POS tagging, and typed dependency parsing. For implementing SPARQL, the ARQ, a query engine for Jena is employed.The ARQ API is bundled in the Jena packages (i.e., *jena-arq.jar*). In addition, Protégé is used for supporting ontology schema construction, butthe knowledge base is developedthrough Netbeans IDE.

There are two kinds of evaluation that have been conducted, including first, evaluation for each phase of the method (see Figure 1), including phase 1 that is the term/relation extraction phase (i.e., the output is a set of query-triples), phase 2 that is thesemantic entity phase (i.e., the output is a set of ontology-compliant query-triples), and phase 3 that is the SPARQL construction and processing phase (i.e., the output is a set of semantic annotations), and second, evaluation by comparing the proposed method (i.e., retrieving document based on the ontology-based why-question analysis) against two baseline methods, the term-based and the phrase-method.

The first evaluation is performed by comparing the output of system agaisntthe manual identification of a set of query-triples, a set of ontology-compliant query-triples, and a set of semantic-entities of a why-question (i.e., as gold standard). Thus, there are three evaluation datasets, first dataset composing pairs of why-question and a set of query triples, second dataset composing pairs of why-question and a set of ontology-compliant query-triples, and third a dataset composing pairs of why-question and a set of semantic annotations.In this research, the evaluation measures of Barker [8] are used for phase 1 and phase 2. It includesfour measures, the precision, the recall, the under-generation, and the over-generation measure. The evaluation measure formulas are,

$$\text{Precision}\,(P) = \frac{\text{correct} + 0.5\,\text{x partial}}{\text{actual}} \quad (1)$$

$$\text{Recall}\,(R) = \frac{\text{correct} + 0.5\,\text{x partial}}{\text{possible}} \quad (2)$$

$$\text{Undergeneration}\,(U) = \frac{\text{missing}}{\text{possible}} \quad (3)$$

$$\text{Overgeneration}\,(O) = \frac{\text{spurious}}{\text{actual}} \quad (4)$$

where,

- *Correct* is the number of triples of a question from outputs of the proposed methodthat match a triple from the gold standard;
- *Partial* is the number of triples of a question from the outputs that almost match the gold standard (i.e., reasonable triple that differ by at most one element);
- *Actual* is total triples of a question from the output;
- *Possible* is total triples of a question in the gold standard;
- *Missing* is the number of triples of a question in the gold standard that have no counterpart in the outputs;
- *Spurious* is the number of triples of a question from the outputsthat have no counterpart in the gold standard.

However, evaluation of phase 3 uses the four measures, the precision, the recall, the under generation, and the over generation without *partial* measure, because the outputs are not in triple-based form.

This evaluation is performed by conducting experiments that generate randomly 100, 200, and 300 questions in 20 iterations. The evaluation performances are the average values of each measure for each phase. The formula of the average measure, $\overline{M}$ is,

$$\overline{M} = \frac{\sum_{i=1}^{20} \frac{\sum_{j=1}^{n} M(Q_{ji})}{n}}{20} \quad (5)$$

where, M is the measure P, R, O, or$U$, n is 100, 200, or 300, and Q_{ji} is the j^{th} question of the i^{th} iteration.

Furthermore, the second evaluation is performed by comparing the result of searching documents (i.e., documents that contain answer of why-questions) based on the proposed ontology-based why-question analysis method against the results of searching documents based on the keyword-based methods (i.e., term (i.e., one-word)-based, and phrase (i.e., multi-word)-based method). This evaluation uses dataset composing pairs of why-question and a set of relevant document that contain answers. The evaluation measures that used in this research are the two standard evaluation measures, MRR (Mean Reciprocal Rank) and P@10 (precision at 10) [35], [36], [37].

$$MRR = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank(first\_relevant\_passage) for question-}i} \quad (6)$$

$$P@10 = \frac{\sum_{i=1}^{N} \text{precision at 10 for question-}i}{N} \quad (7)$$

where, precision at 10 for a question is 1 if the answer to this question is found in top-10 documents and 0 otherwise.

The second evaluation is performed by conducting experiments that retrieve documents satisfied why-questions, where the questions are generated randomly 20, 40, 60, 80, and 100 questions. The experiments are conducted in 20 iterations. The evaluation performances are the average values of each measure (i.e., MRR and P@10) from the 20 iteration results.

The proposed method has been tested on 5367 why-questions in *Text Retrieval* domain. In addition, for the first evaluation, the experiments are also conducted by inputting some questions manually, especially the questions out of well-ordered forms, in order to analyze errors further.

6. RESULTS AND DISCUSSION

Table 8 presents the evaluation results of the first evaluation, which is the evaluation of each phase of the proposed why-question analysis method (see Chapter 5 on evaluation method). As can be seenin Table 8, for all phases, the results show good performance for all measures. The average values of the precision and the recall measures are greater than 99%. Moreover, the average values of the undergeneration and the overgeneration measures are less than 1%.

Table 8. Evaluation results for phase 1, phase 2, and phase 3 in 20 iterations

	Metrics	Phase 1	Phase 2	Phase 3
Data = 100	Precision	99.35%	99.34%	99.40%
	Recall	99.31%	99.30%	99.29%
	Undergeneration	0.72%	0.73%	0.71%
	Overgeneration	0.14%	0.15%	0.00%
Data = 200	Precision	99.24%	99.23%	99.26%
	Recall	99.22%	99.21%	99.21%
	Undergeneration	0.80%	0.81%	0.79%
	Overgeneration	0.06%	0.07%	0.01%
Data = 300	Precision	99.58%	99.57%	99.61%
	Recall	99.55%	99.55%	99.54%
	Undergeneration	0.48%	0.49%	0.46%
	Overgeneration	0.10%	0.12%	0.02%

The performance of the proposed method is very good because the questions generated are in well-ordered forms (see chapter 5 on question collection construction). However, there were still few errors occurred because of misclassification of words. Some verbs such as 'utilize', 'utilized' and 'applied' often are classified as noun, or adjective instead of verb. Words such as 'imprecise' and 'popular' sometimes are classified as noun instead of adjective. These misclassifications cause the method cannot identify question topic, and cannot extract terms and relations, properly.

The noticeable error is in using determinant. If a noun phrase (i.e., especially that has role as subject or object) is not preceded by a determinant, the evaluation performance decreases drastically. Table 9 shows the results of the experiments while a noun phrases are not preceded by a determinant. As shown in Table 9, the average values of the precision and the recall measures decreaseinto around74% for phase 1. It means that the proposed method cannot extract terms and relations properly, so that the intermediate representation of a question (i.e., the query triples) cannot be generated correctly.Much misclassification of noun phrases occurred. The noun phrases that have a last word is a gerund such as 'lexical_matching', 'stemming', and 'TF-IDF_weighting'usually areclassified as verbs. Other case, some noun phrases such as 'interpolated precision statistic', 'query reformulation method', and 'partial matching method' are classified as adjectives.

Table 9. Evaluation results for phase 1 in 20 iterations

Metrics	Data = 100	Data = 200	Data = 300
Precision	75.25%	74.82%	74.31%
Recall	75.25%	74.81%	74.30%
Undergeneration	24.89%	25.34%	25.86%
Overgeneration	0.29%	0.31%	0.33%

Furthermore, by conducting experiments using some questions out of the well-ordered forms, the resultsshow more detailed error analyses. The analyses reveal four main causes of errors,

1. Misspelling of words, words with misspelling out of the proposed threshold cannot be recognized or recognized incorrectly.
2. Misclassification of words, POS tagging tool cannot tag some words properly.
3. Unrecognized question pattern, the proposed method has limited defined question patterns.
4. Undefined terms and relations, the proposed method is applied in a specific domain and the domain ontology is not complete yet.

The second evaluation shows that the proposed ontology-based why question analysis method obtained better performance than baseline methods, the term-based and the phrase-based method. As can be seen in Table 10, the average values of MRR ofthe proposed method are greater than 0.4. It means that, in average, the searching documents by using the proposed ontology-based question analysis can results the most relevant documents at position 2 or 3 (i.e., position in ranked documents retrieved). Comparing with the searching document that uses the term-based method, where the average value of MRR are smaller than 0.16 (i.e., in average, the term-based method can retrieve the most relevant documents at position 6 or below), the proposed method shows the significant improvement, see Table 10. The worst results are shown by the searching documents that uses the phrase-based method, where the average value of MRR are smaller than 0.1 (i.e., in average, the term-based query method can retrieve the most relevant documents at position 10 or below), see Table 10.

Table 10. Comparison results with the baseline methods

	Metrics	The Proposed Ontology-Based Method	The Term-Based Method	The Phrase-Based Method
Data = 20	MRR	0.47357	0.15958	0.09192
	P@10	0.72250	0.27500	0.11375
Data = 40	MRR	0.44888	0.13939	0.08287
	P@10	0.67937	0.26813	0.10000
Data = 60	MRR	0.45953	0.15931	0.09524
	P@10	0.68625	0.27625	0.11542
Data = 80	MRR	0.47556	0.13541	0.08545
	P@10	0.69375	0.23875	0.10031
Data = 100	MRR	0.48032	0.13899	0.08009
	P@10	0.70425	0.24400	0.09700

Similar to the MRR results, the average values of P@10 of the proposed ontology-based why-question analysis method are the highest ones, which are greater than 0.65, and the average values of P@10 of the phrase-based method are the worst ones, which are smaller than 0.12.It means that,by using the proposed method, in average, more than 65% of the most relevant documents are the top-10 documents retrieved, but by using the phrase-based method, only less than 12% of the most relevant documents are the top-10 documents retrieved. Furthermore, the term-based method obtained P@10 smaller than 0.28. It means that, in average, only less than 28% of the relevant documents retrieved by this method are the top-10 documents.

From the results, we can see that even the proposed ontology-based why-question analysis method shows the better performance in term of MRR and P@10than the baseline methods, actually, the proposed method does not result good performance yet. The average value of MRR only around 0.45 (see Table 10) or in the other words, it has to look at 2 or 3 documents total until it find the most relevant one, yielding an efficiency of only 45%. Furthermore, the average value of P@10 only around 0.65 (see Table 10), it means only around 65% of the most relevant documents are top-10 documents. These results are not goodenough for retrieving answers of why-questions. The ideal method should be able to position the most relevant document in the top-1 document; hence the question answering system can retrieve the appropriate answers. It is because most of questions in the evaluation dataset have only one relevant document containing answers.

There are two factors that cause the worsts of the proposed ontology-based why-question analysismethod. The first factor is the indexing. We evaluate the proposed method using a semantic index constructed based on semantic entities of the domain ontology. The indexing system cannot identify all semantic entities contained by documents especially documents that contain answers. Consequently, the system cannot retrieve properly the most relevant documents. The second factor is the position of concepts in a text. In this research, the positions between concepts in a text (i.e., the proximity of concepts) are not involved. However, in fact, the proximity of concepts, especially concepts involved in a causal relation affect significantly the relevance of documents retrieved. The smaller of the proximity of concepts (e.g., concepts in a why-question and in its expansion are in one paragraph), the more relevant documents will be retrieved.

However, out of the weaknesses, the proposed ontology-based why-question analysis method has been able to improve significantly the baseline methods, the keyword-based ones, where the proposed method can improve efficiency until 350%. This fact can be seen from Table 10, where efficiency of the proposed method are around 45% (i.e., the average value of MRR around 0.45), and efficiency of the phrase-query based method only around 10% (i.e., the average value of MRR around 0.1). In addition, the proposed method can also improve the ranking of the most relevant documents until 400% (i.e., can be seen from the average value of P@10 of the proposed method, that is around 0.65, and the phrase-based method, that is 0.12, see Table 10).

7. CONCLUSION

From the first evaluation, as already discussed in chapter 6, the proposed method can be implemented and can result good performance (i.e., in the first evaluation results). The lexico-syntactic patterns over the typed dependency parse trees by considering POS tagging have been implemented to extract terms and relations for constructing query-triples of why-questions (i.e., output of phase 1). This implementation shows high average values ofthe precision and the recall measures, and small average values of the undergenerationand the overgeneration measures (see Table 8). Moreover, the verb classification also has been implemented for identifying the expected answer of why-questions, where the performance can be seen implicitly from the constructed query-triples (i.e., output of phase 1). The semantic mapping using Damerau-Levenstein edit distance and the list of synonymies(i.e., referring tothe list of labels) also has been applied for constructing theontology-compliant query-triples (i.e., output of phase 2). This implementation also shows good value of performance measures (see Table 8). In addition, the SPARQL templates also have been employed for processing the query over the domain ontology, in order to identify the additional semantic annotations of why-questions (i.e., output of phase 3), and it also showsthe good value of performance measures not much different from the previous phases (see Table 8).

However, the proposed method has some drawbacks such as, the lexico-syntactic patterns are constructed manually, and hence it is time consuming. The pattern number is limited, hence the patterns cannot recognize all real question patterns. Furthermore, the limitation of the proposed method is the questions must be in correct English grammar. The implementation of the proposed method is in a specific domain (e.g., text retrieval domain). Thus, terms and relations queriedare restricted in the specific scope.

On the other hand, from the second evaluation results (see Table 10), as have been discussed in chapter6, the performance of the proposed ontology-based why-question analysis method is not good enough for retrieving the most relevant documents that can answer the why questions. The evaluation shows small

value of efficiency, only around 45% (or the position of the most relevant answer is at position 2 or 3) and only around 65% of the most relevant documents are the top-10 documents.

Out of the drawbacks and the limitations of the proposed method, this research has proved that even though it is simple, by only relying on the lexico-syntactic patterns, by considering some assumptions such as the correct English grammar, the recognized question patterns, and the restricted terms and relations of the questions, the proposed method can be implemented and results good performance. Few errors are only caused by misclassification of words performed by POS tagging tool. Moreover, even the proposed method has no high efficiency; the method has been able to improve significantly the baseline methods, the keyword-based one, in both, efficiency and ranking of the most relevant documents (see chapter 6).

For improving the proposed method, it will be better to utilize the machine learning technique for generating the lexico-syntactic patterns automatically. In addition, it needs more efforts to expand the domain ontology so that it covers more complete knowledge, more concepts/instances and relations. Moreover, it is important to improve the indexing technique (i.e., for constructing semantic index) and it is also crucial to involve proximity of concepts for retrieving the most relevant documents.

Thus, our future works are to develop an indexing method that can construct semantic index properly, and to develop a searching method that involves proximity of concepts. Hopefully they can improve the performance of document retrieval system as implemented of the proposed ontology-based why-question analysis method.

ACKNOWLEDGEMENTS

We would like to thank Suzan Verberne, a researcher at Faculty of Arts, Centre for Language & Speech Technology, University of Nijmegen, Netherlands. We use her datasets to compile the why-question datasets that are used for developing the proposed method.

REFERENCES

[1] J. Lear. *Aristotle. The Desire to Understand.* Cambridge University Press, Cambridge, UK. 1988.
[2] S. Verberne. *Developing an Approach for Why-Question Answering*. in the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2006). Trento, Italy. 2006: 39-46.
[3] R. Higashinaka and H. Isozaki. *Corpus-based question answering for why-questions*. in International Joint Conference on Natural Language Processing (IJCNLP-08). Hyderabad, India. 2008:418-425.
[4] S. Nakakura and J. Fukumoto. *Question Answering System beyond Factoid Type Questions*. in the 23rd International Technical Conference Circuits/Systems, Computers and Communications (ITCCSCC-2008). Yamaguchi, Japan. 2008:617-620.
[5] T. Mori, et al. *Answering Any Class of Japanese Non-Factoid Question by Using the Web and Example Q&A Pairs from a Social Q&A Website"*. in IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. Sydney, Australia. 2008; 3: 59-65.
[6] Jing Yu, Dongmei Li, Shudong Hao, Jiajia Hou, Jianxin Wang. Applying Ontology and VSM for Similarity Measure of Test Questions. *Institute of Advanced Engineering and Science (IAES)*. 2014; 12(9): 6932-6939.
[7] Jianghua Li, Qing Cao. DSRM: An Ontology Driven Domain Scientifitic Data Retrieval Mode. *Institute of Advanced Engineering and Science (IAES)*. 2014; 12(2): 1462-1470.
[8] K. Barker et al. *Learning by Reading: A Prototype System, Performance Baseline and Lessons Learned*. in the 22nd Conference on Artificial Intelligence (AAAI-07). Vancouver, British Columbia. 2007:280-286.
[9] A.B. Abacha and P. Zweigenbaum, *Medical QA: Translating Medical Questions into SPARQL Queries.* International Health Informatics Symposium (IHI'12). Florida, USA. 2012.
[10] O. Ferrandez, et al. Addressing Ontology-Based QA with Collection of User Queries. *International Journal of Information Processing and Management*. 2009; 45: 175-188.
[11] P. Adolphs, et al. YAGO-QA: Answering Questions by Structured Knowledge Queries. The *5th IEEE International Conference on Semantic Computing*. Palo Alto, USA. 2011: 158-161.
[12] Unger et al. *Template-based Question Answering over RDF Data.* The 21st International Conference on Word Wide Web (WWW 2012) – Session: Ontology Representation and Querying: RDF and SPARQL. Lyon, France. 2012:639-648.
[13] V. Lopez, et al. AquaLog: An ontology-driven Question Answering System for Semantic intranets. *Journal of Web Semantics*. 2007; 5 (2): 72-105.
[14] D.S. Kim, *et al. Knowledge Integration across Multiple Texts.* The 5th International Conference on Knowledge Capture (KCAP-09).California, USA. 2009.
[15] A. Zouaq, *et al. Linguistic Patterns for Information Extraction in Onto Cmaps.* The 3rd Workshop on Ontology Patterns (WOP 2012), in conjunction with the 11th International Semantic Web Conference (ISWC 2012).2012:61-72.
[16] A. Akbik and J. Brob. *Wanderlust: Extracting Semantic Relations from Natural Language Text Using Dependency Grammar Patterns*. Semantic Search 2009 Workshop (SemSearch '09), pp.6-15, April 21, 2009.

[17] A. Zouaq, *et al.* Semantic Analysis using Dependency-based Grammars and Upper-Level Ontologies. *International Journal of Computational Linguistics and Applications.* 2010; 1(1-2): 85-101.
[18] F. Reichartz. *Semantic Relation Extraction with Kernels Over Typed Dependency Trees.* The 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2010).Wahington, USA. 2010:773-782.
[19] H. Mousavi, *et al.* Mining Semantic Structures from Syntactic Structures in Free Text Documents. *2014 IEEE International Conference on Semantic Computing*. Newport Beach, USA.2014: 84-91.
[20] M. de Marneffe, et al. *Generating Typed Dependency Parses from Phrase Structure Parses.* The 5th Edition of the International Conference on Language Resources and Evaluation (LREC-06).Genoa, Italy. 2006:449-454.
[21] J. Eastwood.*Oxford Guide to English Grammar.* Oxford University Press. 1994.
[22] B. Levin. *English Verb Classes and Alternations - A Preliminary Investigation.* The University of Chicago Press. 1993
[23] V.I. Levenstein. Binary Codes Capable of Correcting Deletions, Insertions and Reselvals*. Cybernetic and Control Theory*. 1966; 10(8): 707-710.
[24] F. Damerau. A Technique for Computer Detection and Correction of Spelling Errors. *Communications of the ACM*. 1964; 7(3): 171-176.
[25] T.R. Gruber. *A translation approach to portable ontology specifications.* Knowledge Acquisition.1993; 5(2): 199-220.
[26] http://www.w3.org/TR/rdf-sparql-query/#initDefinitions
[27] M.P. Álvarez. The Four Causes of Behavior: Aristotle and Skinner. *International Journal of Psychology and Psychological Therapy*. 2009; 9(1): 45-57.
[28] R. Quirk, et al. *A Comprehensive Grammar of the English Language.* Longman, London. 1985.
[29] M. Collins. *Head-Driven Statistical Models for Natural Language Parsing.* Ph.D. Thesis, Computer and Information Science, University of Pennsylvania. 1999.
[30] R. Levy, G. Andrew. *Tregex and Tsurgeon: Tools for Querying and Manipulating Tree Data Structures.* The 5th Edition of the International Conference on Language Resources and Evaluation (LREC-06). 2006:2231-2234.
[31] M.P. Marcus, et al. Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistic*. 1993; 19(2): 313-330.
[32] M. de Marneffe and C. D. Manning.*Stanford Typed Dependencies Manual.* 2008.
[33] sverberne.ruhosting.nl/
[34] G. Canfora and L. Cerulo. A Taxonomy of Information Retrieval Models and Tools. *Journal of Computing and Information Technology (CIT 12).* 2004; 3: 175–194.
[35] C. D. Manning, *et al*. *Introduction to Information Retrieval.* Cambridge University Press, New York. 2008.
[36] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press, New York. 1999.
[37] J.A. Thom, F. Scholer. *A Comparison of Evaluation Measures Given How Users Perform on Search Tasks*. 12th Australasian Document Computing Symposium. 2007: 100-103.

BIOGRAPHIES OF AUTHORS

A.A.I.N. Eka Karyawati, S.Si., M.Eng. She received her S1 degree in Mathematics from Bogor Agricultural University, Indonesia, and MEng. in Information Science and Systems Engineering from Ritsumeikan University, Japan. Now, she is studying in Doctoral Program of Computer Sciences, Gadjah Mada University, Indonesia. Her research interests are in text mining, natural language processing, information retrieval, and knowledge representation.

Drs. Edi Winarko, M.Sc., Ph.D. He received his S1 degree in Statistics from Gadjah Mada University, MSc. in Computer Sciences from Queen's University, Canada, and PhD. in Computer Sciences from Flinders University, Australia. His research interestsare in data warehousing and data mining, and information retrieval.

Dr. Drs. Azhari, M.T. He received his S1 degree in Statistics from Gadjah Mada University, Indonesia, MT. in Informatics from Bandung Institute of Technology, Indonesia, and Dr. in Computer Sciences from Gadjah Mada University, Indonesia. His research interests are in intelligent autonomous system, intelligent agent and multiagent system, knowledge management system, intelligent enterprise system, intelligent information system, software engineering methodology and application, and object oriented methodology and application.

Drs. Agus Harjoko, M.Sc., Ph.D. He received his S1 degree in Electronics and Instrumentation from Gadjah Mada University, Indonesia, MSc. and PhD. in Computer Sciences from University of New Brunswick, Canada. His research interests are in multimedia processing, computer/machine vision, and medical instrumentation.