International Journal of Knowledge-based and Intelligent Engineering Systems 17 (2013) 167–176
DOI 10.3233/KES-130270
IOS Press

Pattern-based core word recognition to support ontology matching

Fuqi Song*, Gregory Zacharewicz and David Chen
University Bordeaux, IMS UMR, Talence, France

Abstract. Ontology matching is a crucial issue in the domain of semantic web and data interoperability. In this paper, a core word based method for measuring similarity from the semantic level of ontology entities is described. In ontology, most labels of entities are compound words rather than single meaningful words. However, the main meaning is represented usually by one word of them, which is called core word. The core word is learned by investigating certain patterns, which are defined based on part of speech (POS) and linguistics knowledge. The other information is noted as complementary information. An algorithm is given to measure the similarity between a pair of compound words and short texts. In order to support diverse situation, especially when core words cannot be recognized, non semantic based ontology matching techniques are applied from lexical and structural level of ontology. The described method is tested on real ontology and benchmarking data sets. It showed good matching ability and obtained promising results.

Keywords: Ontology matching, core word, pattern recognition

1. Introduction

Ontology matching is a crucial issue in the domain of semantic integration for data interoperability, which is an essential part of Enterprise Information System (EIS) interoperability [1]. The major issue of ontology matching is to find correspondences between entities. Ontology matching has been studied for years, many matching techniques have been proposed. They try to discover the matching from lexical and structural level. An intuitive idea is why not perform the matching just from the semantics and try to understand the entities like human beings.

With this idea, we apply the research in the domain of natural language processing (NLP), especially, information extraction (IE) to ontology matching. Ontology is usually created to represent specific concepts and relations in a domain. The labels used for naming entities are alike natural language. Normally, they are consisted with several single meaning words. These compound words or short phrases focus on expressing one core meaning, unlike the normal complete sentence, which may intend to express several meanings. The main meaning is denoted in one word, which is called "core word". Thus, if the core word can be identified, it would be easier and helpful to find equivalent semantic correspondence. This is the base of this work, form this point, we propose to use pattern recognition with part of speech (POS) to learn the core word, and then measure the semantic similarity with core word and complementary information.

The hypothesis to apply the method is that the labels of entities in ontology should be alike natural language. For the situation with randomly generated strings and less meaningful compound words, the method is less applicable. To adapt the diverse situation, especially for the case that no core words could be recognized, two non-semantic based matchers are applied. The two matchers seek to discover the correspondences from lexical and structural level of ontology.

The reminder of the paper is organized as follows. Section 2 recalls the related work of ontology matching and some related work to pattern recognition. Section 3 describes the proposed method of pattern recog-

*Corresponding author: Fuqi Song, University Bordeaux, IMS UMR 5218, 351 Cours de la Libération, F-33400 Talence, France. E-mail: song.fuqi@gmail.com.

ISSN 1327-2314/13/$27.50

nition and core word identification, also the algorithm for measuring semantic similarity. Section 4 introduces two non-semantic based matching techniques from lexical and structural level of source ontology, as well as the aggregation process. Section 5 evaluates the proposed approach with an illustrative case and benchmark testing. A brief discussion is given. Section 6 draws some major conclusions.

2. Related work

Ontology matching seeks to find semantic correspondences between a pair of ontology entities by identifying semantic relations. A definition of correspondence from Euzenat and Shvaiko [2] is: given two ontology o and o' with associated entity languages O_L and $O_{L'}$, a set of alignment relations Θ and a confidence structure over Ξ, a correspondence is a 5-uple:

$$\{id, e, e', r, n\},$$

such that id is a unique identifier of the given correspondence, $e \in Q_L(o)$, $e' \in Q_{L'}(o')$, $r \in \Theta$ and $n \in \Xi$.

2.1. Ontology matching

The similarity-based matching approaches seek to discover the equal relations between entities in ontology. Hierarchical relation, such as, super class and child class, and the other relations are beyond the ability of similarity-based approaches. This paper focuses on discovering equal relation between ontology, the other types of relation are not considered.

The entities to be matched include: classes, instances and properties. In some approaches, more information of ontology is adopted, for example, data type and value are used to calculate the similarity in Euzent and Shvaiko [3]. However, usually this kind of information plays a role of complementary information to support match the above three types of entities. In this article, this information is not investigated.

Song et al. [4] classify the ontology matching approaches by considering three levels of source ontology to be matched. At entity level, the class itself is treated as the object of study; the label, comment and internal information of it are investigated. The mostly used techniques are string metric [5], string similarity, domain, property and data type comparison [6]. At local level, the objects and the relations linked to the studied entity are taken into account, such as, similarity flooding [7], graph-based approach and taxonomic-based approach. At global level, the whole ontology is taken as a semantic context, and the approach uses this context to seek the semantic correspondence. Machine learning, artificial neural network [8] are some methods applied at this level. Granitzer et al. [9] and Yan et al. [10] gave comprehensive introduction and comparison of different basic matching techniques and applications. The methods can be classified into the above three levels.

2.2. Information extraction (IE)

"*Natural Language Processing (NLP) strives to enable computers to make sense of human language*". NLP has been proposed and studied more than half century. It seeks ways to make computers understand human natural language. The input resources could be speech, text and multimedia. In the domains of artificial intelligence (AI) and human-computer interaction (HCI), NLP is a major research topic. In NLP, there are many research issues involved. Concerning to identifying core word in ontology, a few topics are involved: information extraction (IE) and named entity recognition (NER).

IE refers to extracting structured information from information sources automatically. The extraction process respects to certain pre-defined rules. NER is a subtask of IE. NER seeks to find and recognize the atomic elements in text. For instance, "*the book title*" will be recognized as *the (article) book (noun) title (noun)*. The recognition rules are various, in this example, it is recognized by the part of speech (POS) of words.

Some related work in this area is listed in Table 1. Muslea [11] investigated the different extraction patterns in information extraction. The authors [12–14] applied extraction patterns to free text and documents. In Ceausu [12] and Sari et al. [14], the patterns are focused on specific information, such as, the date and location, which are important in accident report. Maynard et al. [13] used patterns to extract and create ontology from free text. It could build semantic relations in ontology. Ritze et al. [15] and Svab-Zamazal et al. [16] adopted patterns to perform ontology matching for discovering complex correspondences, which are in the relevant research domain to our work. They defined a set of patterns from one or several related entities in ontology and used the pattern to find correspondences. The patterns are learned from the mostly used forms when creating ontology. In this paper, the patterns are

Table 1
Investigation of IE and NER based approaches

Author(s)	Type	Pattern recognition	Extraction source	Application
(Muslea, 1999) [11]	Survey	–	–	–
(Ceausu and Despre, 2007) [12]	Framework	POS-based	Accident report	Text categorization Accident report
(Maynard et al., 2009) [13]	Tool	NER hearst pattern Lexical-syntactic pattern Contextual pattern	Free text	Ontology extraction ontology creation
(Sari et al., 2010) [14]	Method	Date and time Location Accident effect	Free text, document Structured documents	Creating extraction pattern
(Ritze et al., 2008) [15]	Method	Class by attribute type pattern (CAT)....	Ontology	Detecting complex correspondences
(Svab-zamazal and Svaek, 2011) [16]	Theory	NER	OWL ontology	Ontology matching

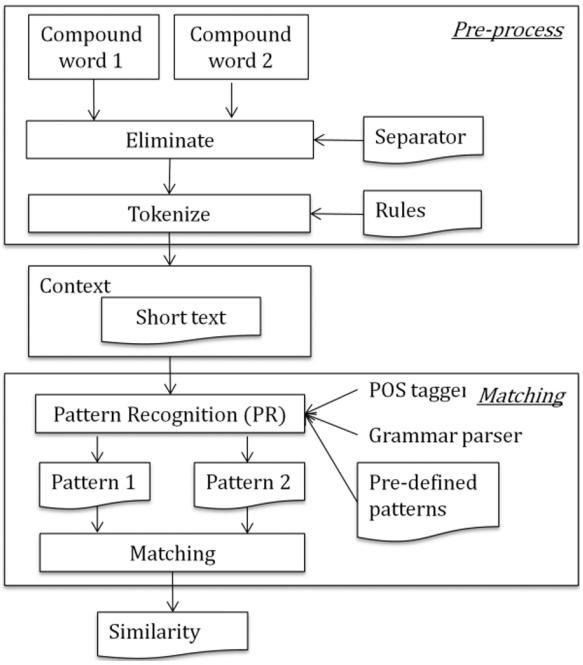

Fig. 1. Process of core word recognition.

recognized based on POS and linguistics. The purpose of obtaining patterns is not to find the matching directly, rather a way to find the core word. The core word is used to aid discover the correspondences.

3. Matching with pattern-based core word (PCW)

Core word is one or more word(s), which represent the main meaning in a compound word or short phrase. A process (see Fig. 1) is given to measure the similarity confidence between two compound words. A pair of compound words or short phrases is as input. First, the stop words and superfluous information are eliminated from the label, and then the label is tokenized into several single words. With pre-defined patterns, the short text is recognized into each category. In this process, POS tagger and grammar parser are applied. At last, the recognized pattern and core word will be used to measure the similarity.

3.1. Elimination and tokenization

Before pattern recognition, elimination and tokenization of core words are performed as the preprocess. Most of labels are composed of several words with stop words and separators. First, elimination helps to eliminate the unnecessary information which could confuse the matching task. Then tokenization splits the compound word into atomic ones. The compound word is tokenized by rules: 1) Stop words, such as, dash, underscore and dot; 2) Capitalized word, for example,
"*numberOfTelephone*" is tokenized into "*number, of, telephone*".

3.2. Pattern recognition and core word identification

Ontology, as the text source, is different from free text and document. The labels of entities are the main carriers of text. The labels commonly follow specific rules and most of them are compound words and short phrases. Usually verb-based labels are used for labeling object property (relation), such as, *hasName* and *applyTo*. Noun-based labels are used for labeling class and data property, such as, *blackBook* and *conferenceMember*. From this perspective, certain patterns could be concluded from labels of source ontology. Unlike complete phrases, the label concentrates on representing one simple meaning. Thus it is important to find

Table 2
Part-of-speech (POS) tagging

POS	Prefix	POS tagging type	Remark
Noun	NN-	NN, NNP, NNPS, NNS	Noun and proper noun, singular and plural
Verb	VB-	VB, VBP, VBZ, VBD	Verb base form, singular present, past tense
		VBG	Verb, Present participle
		VBN	Verb, past participle
Preposition	IN	IN	Preposition, of, by,
Adjective	JJ-	JJ, JJR, JJS	Adjective, comparative form, superlative form
Other	O-		Except the above POS

Table 3
Recognition patterns

		Composition mode	Pattern	Com. info.	Remark
Noun-based	Nouns group (NNG)	Single noun	NN*	–	The noun
		Multi-nouns	NN(+)-NN*	NNs	The last noun
		Multi-nouns with 'of'	NN*-of-NNG	NNs	Noun before 'of'
	Modifier-noun (MM-NNG)	Adjective-noun(s)	JJ-NNG*	JJ, NNs	The noun
		Past participate-noun(s)	VBN-NNG*	VBN, NNs	The noun
		Present participle-noun(s)	VBG-NNG*	VBG, NNs	The noun
Verb-based	Verb	Single verb	VB*	NNs	Verb
	Verb-object	Verb-noun	VB*-NNG	NNs	Verb
		Verb-prep-noun	VB*-PP(-NNG)	NNs	Verb
		Passive form	VBN-by(-NNG)	NNs	Verb

* core word + one to more.

out which word is the core word. It helps to understand the semantics.

The types of part-of-speech (POS) used in the approach are listed in Table 2. To tag the POS of words, postagger [17] from Stanford University is used. Mainly nouns, verbs, adjectives and part of prepositions are tagged. The words with the other POS are ignored, such as, article and conjunction, because they do not help significantly in representing the major meaning. For nouns, there are four types: singular noun (NN), singular proper noun (NNP), plural noun (NNS) and plural proper noun (NNPS). For verbs, there are different tenses and participles. In preposition, only "of" and "by" are tagged, the others are ignored. For adjectives, there are base form (JJ), comparative form (JJR) and superlative form (JJS). Sometimes present and past participate are used as adjectives, such as "*edited book*". For adjectives and this kind of verbs, they are called as "modifier" in general.

In order to obtain the patterns mostly used, the real-life ontology and experimental ontology are studied. The most commonly used patterns are concluded in Table 3. The first column shows the composition mode of word, and then the pattern. A star symbol (*) indicates that the tagged word is identified as core word. Besides the core word, complementary information is also noted, such as, multiple nouns and the passive tense. The representation of this information is denoted as *(core word, <type, complement info. 1, type, complement info. 2, ... >)*. For instance, *(conduct, < form, pass>)* denotes that the core word is "*conduct*" with a passive voice.

NNG is used to represent a group of nouns, including one or more nouns. *NNs* represents the complementary information, it is composed with several nouns in sequence. There are two special cases with preposition "of" and "by". "Of" changes the position of core word in a multiple-noun mode. For example, the core words of "*titleOfBook*" and "*bookTitle*" are both "*title*", but the position is different. "By" is used to identify whether a verb is past form or modifier. For example, in "*editedBook*", "*edited*" is a modifier. In "*editedByAuthor*", "*edited*" is a past form of "*edit*". The details and examples of each pattern are given in Table 4.

3.3. Semantic similarity measuring with PCW

Two similarity measuring algorithms are used in Semantic MAtching (SMA). Lin model [18] is a reused and adapted method. A homonym checker is proposed to solve homonym issue in semantic matching.

Lin model [18] is a taxonomy-based model for measuring semantic similarity. Lin model takes the taxonomy as a tree and returns the semantic similarity by measuring communality between two words in the taxonomy tree. WordNet [19] is used as the taxonomy in this paper.

Table 4
Examples of patterns and core words

Type	Composition mode	Example	Core word	Compl. info.
Nouns group	Single noun	Book, books	Book	–
	Multi-nouns	Book_title, BookTitle	Title	Book
	Multi-nouns with 'of'	TitleOfBook	Title	Book
Modifier-noun	Adjective-noun(s)	ShortName	Name	Short
	Past participate-noun(s)	PublishedBook	Book	Published
	Present participle-nouns(s)	PublishingManagerBook	Book	Publishing, manager
Verb	Single verb	Uses	Use	–
Verb-object	Verb-noun	HasSiblingsOf	Have	Siblings
	Verb-prep(-noun)	Submits_to_conference	Submit	Conference
	Passive form	WrittenByAuthor	Write	Author

Homonym checker: Homonym is a special case in semantic matching. The same word represents different meanings in different contexts. For example, "*article*" may refer to a publication or refer to a product. First whether the two ontologies, where the homonyms are occurred, belong to the same context is measured. A semantic similarity indicator I_s helps to examine whether they belong to the same context. I_s is computed based on the identified core words, not on the original labels. I_s is defined in Eq. (1), where *#synonym* is the number of synonyms identified between O_1 and O_2, and *tcp* is the number of total concepts and properties. For a word in ontology O_1, if there is a synonym existing in O_2, then *#synonym* count adds 1.

$$I_s = \#\text{synonym}/\min(\#\text{tcp}_1, \#\text{tcp}_2), \tag{1}$$

A threshold *th* is set. If the indicator I_s is greater than the threshold *th*, then the two ontologies are considered as belonging to same context. In this case, the two words are considered as identical and the similarity is assigned to 1.0. Otherwise, a formula (see Eq. (2)) is applied for computing the similarity of a pair of homonyms, where *#m* is the number of different explanations (retrieved from WordNet) that the word has. In this work, the threshold is set manually as $th = 0.2$.

$$H(e) = (\#m - 1)/\#m \tag{2}$$

An overall similarity measurement between two single concepts of *SMA* is as Eq. (3).

$$\text{SMA}(e_1, e_2) = \begin{Bmatrix} \text{LinModel}(e_1, e_2), \text{not homonym} \\ H(e), \text{homonym; th} < I_s \\ 1, \text{homonym; th} \geqslant I_s \end{Bmatrix} \tag{3}$$

In order to measure the similarity between two short texts, a pair of patterns with core word and complementary information is as input. The format of input is defined as

(type[noun-based, verb-based], pattern, core word, <complementary info.1, type1>, <complementary info.2, type2>...).

For example, after a series of processes mentioned above, the label "_theShortTitle_OfBook" generates the input as *(Noun-based, JJ-NN-of-NN, title, <short, MODIFIER>, <book, MULTI_NOUN>)*.

a) Original label	_theShortTitle_OfBook
b) Elimination and tokenization	short title of book
c) Pattern recognition	JJ-NNG $\gg$ JJ-NN-IN-NN $\gg$ JJ-NN1-of-NN2
d) Core word	title
e) Complementary information	short, MODIFIER; book, MULTI_NOUN

The algorithm of measuring the similarity of two short texts with above given format is based on *SMA* (see Eq. (3)). SMA aims to measure the similarity between a pair of single concepts in a semantic context. PCW (see Eq. (4)) utilizes SMA as a component to compose the algorithm. There are two parts involved: core word part M_1 (see Eq. (5)) and complementary information part M_2 (see Eq. (6)). Core word is considered more important in representing the semantic, thus the weight of M_1 and M_2 are set to 0.7 and 0.3 manually. In the algorithm, *cw* denotes the core word of e, *CI* denotes the set of complementary information $\{ci1, ci2, \ldots\}$ of e with length l. $ci1_k$ is one arbitrary entity of set CI_1. If two inputs share the same core word and one complementary word, the confidence is assigned to 1. Otherwise, the similarity is accumulated based on each pair of them.

$$\begin{aligned} \text{PCW}(e_1, e_2) = {} & 0.7 * M_1(cw_1, cw_2) \\ & + 0.3 * M_2(CI_1, CI_2) \end{aligned} \tag{4}$$

$$M_1 = \begin{Bmatrix} 1, cw_1 = cw_2 \\ \text{SMA}(cw_1, cw_2), cw_1 \neq cw_2 \end{Bmatrix} \tag{5}$$

Table 5
Sample of Jaro-Winkler distance between "winkler" and "wenklir"

	W	I	N	K	L	E	R
W	*1*	0	0	0	0	0	0
E	0	0	0	0	0	~~*0*~~	0
N	0	0	*1*	0	0	0	0
K	0	0	0	*1*	0	0	0
L	0	0	0	0	*1*	0	0
I	0	~~*0*~~	0	0	0	0	0
R	0	0	0	0	0	0	*1*

$$M_2 = \left\{ \begin{array}{l} 1, ci1_k = ci2_j \\ \sum \mathrm{SMA}(ci1_k, ci2_j)/(l_1 * l_2), \\ ci1_k \neq ci2_j \\ ci1_k \in CI_1, ci2_j \in CI_2, \\ 0 < k < l_1, 0 < j < l_2 \end{array} \right\} \quad (6)$$

4. Non-semantic based matching and aggregation

Since source ontology usually has complex situations, it is important to perform ontology matching from non-semantic levels. There are lexicon-based and structure-based matching techniques, which are regardless of the semantics that the entities represented. Two matchers are used in the work: edit distance (ED) and directed graph (DG).

4.1. Edit distance (ED)

String matchers are designed based on the string, which presents the labels of concepts and properties. These entities are treated only as a sequence of letters, without considering the meaning represented and structure contained. String metric measures similarity or distance between two plain strings. Distance function maps a pair of string s_1 and s_2 to a real number r, where a smaller value of r indicates greater similarity between e_1 and e_2 [20].

Levenshtein distance (also known as edit distance) is the mostly known distance function, in which distance is the cost of operations, including insertion, deletion and substitution, for converting s_1 to s_2 in a best sequence. A broadly string metric Jaro-Winkler [21] distance is applied. It was proposed by Winkler based on Jaro distance [22,23]. Jaro distance is defined in Eq. (7), where e_1 and e_2 are string from O_1 and O_2, m is the number of matching character and t is half of the transportation number. Two characters are matched only when the distance is not beyond the matching window, i.e. taking a_i and b_j (i, j denotes the sequence in the string) character from e_1 and e_2, if $a_i = b_j$ and $j - g < i < j + g$, where $g = \max(|s_1|, |s_2|)/2 - 1$.

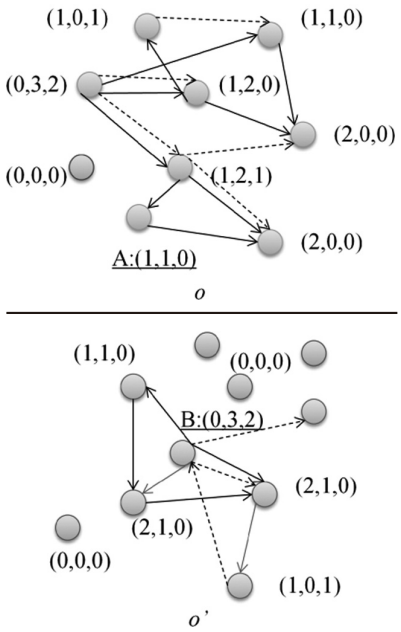

Fig. 2. Sample of directed graph. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/KES-130270)

$$\mathrm{Jaro}(e_1, e_2) = \frac{1}{3} * \left(\frac{m}{|e_1|} + \frac{m}{|e_2|} + \frac{m-t}{m} \right) \quad (7)$$

Jaro-Winkler distance adds a weight for common prefix. It is defined in Eq. (8). P is the length of longest common prefix of e_1 and e_2, $\min(P, 4)/10$ is for assuring the coefficient not exceeding 0.25, which may cause consequently *ED*(e_1, e_2) greater than 1.

$$\mathrm{ED}(e_1, e_2) = \mathrm{Jaro}(e_1, e_2) + \frac{\min(P, 4)}{10} * (1 - \mathrm{Jaro}(e_1, e_2)) \quad (8)$$

For example, given strings $e_1 =$ "winkler" and $e_2 =$ "wenklir", then $|e_1| = 7, |e_2| = 7$ and $g = \max(7, 7)/2 - 1 = 2$. The matching process is shown in Table 5, the shadowed table cell represents the matching window. For 'E' and 'I', they cannot be matched because they are beyond of the matching window. Then $m = 5$, the matched string is "WNKLR" and "WNKLR". The sequence are the same, thus no transportation is needed, then $t = 0$. The distance *Jaro*("*winkler*", "*wenklir*") $= 1/3 * (5/7 + 5/7 + (5-0)/5) = 17/21 = 0.809$. The longest prefix is "w", then $P = 1$, thus *ED*("*winkler*", "*wenklir*") $= 0.809 + 0.1 * (1 - 0.809) = 0.828$.

4.2. Directed graph (DG)

Directed graph (or digraph) G is represented as $G =< V, E >$, V is a set of vertices (or nodes) and E

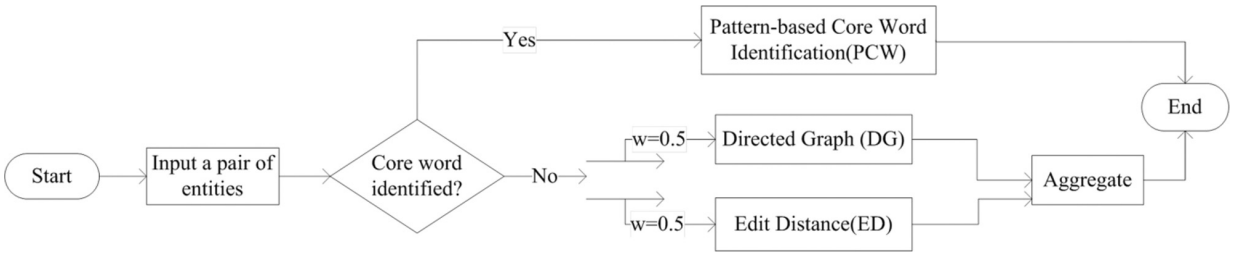

Fig. 3. Aggregation process.

is a set of edges with ordered pairs of vertices (v_i, v_j) from V. A vertex in ontology is described as (*#indegree, #outdegree, #subclass*). The similarity between two vertices is defined in Eq. (9), where *inR, outR and subR* denote the ratio between *#indegree, #outdegree, #subclass* of two vertices v_1 and v_2 from O_1 and O_2. Taking *inR* for example, *inR* = min (*#indegree*$_1$, *#indegree*$_2$)/max (*#indegree*$_1$, *#indegree*$_2$). If both values equal to 0, then *inR* = 0.

$$DG\,(e_1, e_2) = (inR + outR + subR)/3 \qquad (9)$$

In Fig. 2, an illustration sample is presented to show the directed graph of ontology o and o'. The solid line denotes the *sub-class* relation and dotted line denotes the relation, for example, the similarity between vertex $A(1, 1, 0)$ and vertex $B(0, 3, 2)$ is $(0 + 1/3 + 0)/3 = 1/9$.

4.3. Aggregation

So far the matching techniques have been described. In order to select and aggregate them, a flow process is given in Fig. 3. A pair of entities is as input. The source ontology is processed into a set of entities, including classes and properties (datatype property and object property). The matching algorithm is performed only between entities with the same type, for example, both entities are classes.

With pattern-based core word identification, whether there is a core word existing is checked. If the original label is a compound word and can be tokenized into several single words, then a core word should be identified. If the label can neither be tokenized nor be found in the lexical database (WordNet), such as *txdf*, then it is considered that no core word has been recognized. If core word is identified, then PCW (Eq. (4)) matcher will be used to match. Otherwise non-semantic matchers ED (Eq. (8)) and DG (Eq. (9)) will be applied. Each of them takes up 50% weight.

5. Evaluation

To test and validate the proposed approach, a software prototype was developed in Java. It uses WordNet [24] as lexical database for checking synonyms and homonyms, and postagger [17] for identifying core words. The java APIs used in implementation are JWI [25], JWS [26] and Alignment API [27]. First, a pair of real ontology is used to illustrate the proposed matching method, and then benchmarking with test cases of OAEI [28] is performed.

5.1. Illustrative case

Ontology *EKAW*is used to test the pattern recognition approach. The ontology is available at http://oaei.ontologymatching.org/2012/conference/data/ekaw.owl, it contains 74 classes and 33 object properties. The ontology is about the domain of conference and publication. There are total 106 entities recognized, part of the results are kept without changing in Table 6. There are original label, identified pattern, core word and complementary information.

Most of the entities can be identified correctly as expected. However, a few of them cannot be recognized correctly (in italic font in Table 6). The reason is that the precision of postagger is not 100%. For the words which have several POS, for instance, “industrial” and “abstract” are both nouns and adjectives; the precision of postagger relies much on the context. Also, for some compound word, the precision is relatively affected, such “early-registered” and “camera-ready”, these words should be taken as one word, but in current approach, it is difficult to tokenized and recognize automatically. The incorrectly identified core word and patterns are counted manually, regarding to their real semantics. There are nine misidentified patterns out of 106, and then the precision is 91.5%.

Another ontology *OpenConf*, which is also in the domain of conference organization and available at http://oaei.ontologymatching.org/2012/conference/

Table 6
Recognized pattern and core word

Original label	Pattern	Core word	Complementary information
Abstract	*JJ-*	*(Abstract, MODIFIER)*	
Academic_Institution	NN-NN-	(Institution, MULTIPLE_NOUNS)	<MULTI_NOUN,Academic>
Accepted_Paper	JJ-NN-	(Paper, SINGLE_NOUN)	<MODIFER,Accepted>
Agency_Staff_Member	NN-NN-NN-	(Member, MULTIPLE_NOUNS)	<MULTI_NOUN,Agency> <MULTI_NOUN,Staff>
Camera_Ready_Paper	*NN-NN-NN-*	*Camera-Ready-Paper-*	*(MULTIPLE_NOUN, Paper)*
Conference_Banquet	NN-NN-	(Banquet, MULTIPLE_NOUNS)	<MULTI_NOUN,Conference>
Demo_Chair	NN-NN-	(Chair, MULTIPLE_NOUNS)	<MULTI_NOUN,Demo>
Early-Registered_Participant	*O-NN-NN-*	*Early-Registered-Participant-*	(*MULTIPLE_NOUN,Participant)*
Organising_Agency	NN-NN-	(Agency, MULTIPLE_NOUNS)	<MULTI_NOUN,Organising>
Paper	NN-	(Paper, SINGLE_NOUN)	
Proceedings_Publisher	NN-NN-	(Publisher, MULTIPLE_NOUNS)	<MULTI_NOUN,Proceedings>
Submitted_Paper	NN-NN-	(Paper, MULTIPLE_NOUNS)	<MULTI_NOUN,Submitted>
Tutorial_Chair	NN-NN-	(Chair, MULTIPLE_NOUNS)	<MULTI_NOUN,Tutorial>
authorOf	NN-IN-	(Author, SINGLE_NOUN)	
coversTopic	NN-NN-	(Topic, MULTIPLE_NOUNS)	<MULTI_NOUN,covers>
paperPresentedAs	NN-VBN-O-	(Paper, SINGLE_NOUN)	<MODIFER,Presented>
referencedIn	VBN-O-	(Referenced, MODIFIER)	
writtenBy	VB-IN-	(Written, VERB_BASED)	
.....	.....	.....	.....

Table 7
Discovered correspondences with threshold = 0.7

Entity in *EKAW*	Entity in *OpenConf*	Sim.	Entity in *EKAW*	Entity in *OpenConf*	Sim.
Demo_Chair	Program_chair	0.74	Social_Event	Result_of_Advocate	0.70
Document	Text	0.86	Submitted_Paper	Submitted_Paper	1.00
Event	Result_of_Advocate	0.91	Tutorial_Abstract	Conference_Program	0.72
Industrial_Paper	Paper	0.70	Tutorial_Chair	Program_chair	0.70
OC_Member	Member	0.75	Workshop_Chair	Program_chair	0.70
PC_Member	Member	0.75	Workshop_Paper	Paper	0.70
Paper	Paper	1.00	HasEvent	Has_Result	0.86
Paper_Author	Contact_Author	0.75	HasPart	Has_made_review	0.83
Research_Topic	Domain_Topic	0.71	HasReview	Has_Review	1.00
SC_Member	Member	0.75	HasReviewer	Has_Review	0.75
Scientific_Event	Result_of_Advocate	0.70	HasUpdatedVersion	Has_Result	0.82
Session_Chair	Program_chair	0.71	ReviewWrittenBy	Is_written_by	0.78
...	...	...	...	...	...

data/OpenConf.owl, is used to perform ontology matching. In *OpenConf*, there are 61 classes, 21 datatype properties and 24 object properties. 106 correspondences are discovered, with threshold $th = 0.7$ (set manually), there are 24 correspondences filtered as shown in Table 7.

5.2. Benchmarking

The data set for experiment is from OAEI 2011 benchmark [28,29]. Data set *biblio* has been used since 2004 and the seed ontology concerns bibliographic references, which contains 33 named classes, 24 objet properties, 40 data properties, 56 named individuals and 20 anonymous individuals. The data sets are generated based on the seed ontology. Data set are grouped into four test cases T1 to T4 in Table 8. Test case T1 contains three ontology with small changes in labels and structure. Test case T2 contains ten ontology with same structure and different lexical labels. Test case T3 has many variations in structure. Data set #248 to #266 has variations in both aspects, especially, the labels are randomly generated strings. So in the test, this group of test cases is not chosen, because the pattern and core word recognition are based on meaningful compound words. Test cases #301 to #304 are four real-life ontology created by different institutions.

Table 8
Benchmark data set *biblio*

Test case #	Data set	No. of ontology	Description
T1	#101 – #104	3	Simple ontology
T2	#201 – #210	10	Variations in lexical aspect
T3	#221 – #247	18	Variations in structural aspect
T4	#301 – #304	4	Real-life ontology

Three measurements are used to evaluate the match-

Table 9
Results of test case T2

Threshold	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	Average
Precision	0.65	0.65	0.65	0.65	0.68	0.75	0.83	0.90	0.95	0.98	0.99	0.79
Recall	0.65	0.65	0.65	0.65	0.65	0.64	0.63	0.61	0.60	0.53	0.41	0.60
F-Measure	0.65	0.65	0.65	0.65	0.66	0.69	0.72	0.73	0.73	0.69	0.58	0.68

Table 10
Evaluation results of test case T1 to T4

Test case	Data set	Precision (average)	Recall	F1-Measure
T1	#101 – #104	1.00	1.00	1.00
T2	#201 – #210	0.79	0.60	0.68
T3	#221 – #247	0.95	1.00	0.97
T4	#301 – #304	0.58	0.44	0.50
	Average	0.83	0.76	0.80

ing results: precision (P), recall (R) and F1-measure ($F1$). According to Euzenat [30], precision measures the ratio of correctly found correspondences over the total number of returned correspondences, while recall measures the ratio of correctly found correspondences over the total number of expected correspondences. In logical term, precision and recall are supposed to measure the correctness and completeness of method respectively. F1-measure combines and balances between precision and recall. The measurements are denoted in Eq. (10). The set of alignments identified by our approach is denoted as A_d, and the set of reference alignments is denoted as A_r.

$$P = \frac{|A_d \cap A_r|}{|A_d|} R = \frac{|A_d \cap A_r|}{|A_r|} F1 = \frac{2 * P * R}{P + R} \tag{10}$$

For each data set, the results are generated into 10 groups by respecting to the threshold, which distributing from 0.0 to 1.0 with interval 0.1. In Table 9, the results of test case T2 is listed. In the last column, the average precision, recall and F1-measure is given. In Table 10, the average precision, recall and F1-measure of all test cases are listed. The precisions of all test cases T1 to T4 are relatively high, and the average precision is 0.83. Recall of test cases T1 and T3 are 1.0. Recall of test cases T2 and T4 are relatively low, 0.60 and 0.44 respectively. The average recall of all test cases is 0.76 and average F1-measure is 0.80.

5.3. Discussion

The aim of PCW is to identify core words from natural language alike compound word or short phrases, thus the hypothesis of usage and application of the method is that the description of ontology should be alike natural languages. The ontology, which is constructed by random strings or few meaningful words, is not applicable to use the method. Another issue about the precision is caused by the limitations of the lexical database, which is WordNet in our approach. Some words and their special meanings may not be included in the database, so that the algorithm could not generate accurate results. For example, the meaning of "*MS word*", which is a name of word processing software, cannot be identified correctly with WordNet. A solution to this issue is to define a special name list, which contains the unusual meanings and uncommon words, such as, "*PDF*" and "*MS word*". Then assign these names with a commonly used equivalent concept, for example, using "*format*" to replace "*PDF*" and "*software*" to replace "*MS word*". Because of the complexity and diversity of language environment, the patterns can vary tremendously. The patterns defined in this article are commonly used in general domain. It may work differently on some specific domains. This issue allows the room to improve and extend the patterns in order to adapt to different language environment.

6. Conclusion

In this paper, a pattern-based approach to recognize the core word of compound word is described. This method allows measuring the semantic similarity between a pair of compound words. It emphasizes on extracting the main meaning of one compound word, and uses it to find similar entities. This method is applied to support ontology matching, and it showed good matching ability and obtained promising results. However, semantic measurement of short compound words and short phrases is a basic issue in the domains of semantic web and semantic interoperability. It is believed that the method could also be applied to support this research and have certain contributions.

References

[1] F. Song, G. Zacharewicz and D. Chen, An architecture for interoperability of enterprise information systems based on

SOA and semantic web technologies, in: *Proceedings of 13th International Conference on Enterprise Information Systems*, Beijing, SciTePress, 2011, pp. 431–437.

[2] J. Euzenat and P. Shvaiko, *Ontology Matching*, Heidelberg, Springer, 2007, p. 341.

[3] J. Euzenat and P. Valtchev, Similarity-based ontology alignment in OWL lite, in: *Proceedings of 16th European Conference on Artificial Intelligence*, Valencia, Spain, IOS Press, 2004.

[4] F. Song, G. Zacharewicz and D. Chen, An ontology-driven framework towards building enterprise semantic information layer, *Advanced Engineering Informatics* **27**(1) (2013), 38–50.

[5] G. Stoilos, G. Stamou and S. Kollias, A string metric for ontology alignment, in: *Proceedings of 4th International Conference on The Semantic Web* Galway, Ireland: Springer (2005), 624–637.

[6] M. Ehrig and S. Staab, QOM – quick ontology mapping, in: *The Semantic Web – ISWC 2004*, S.A. McIlraith, D. Plexousakis and F.V. Harmelen, eds, Heidelberg, Springer, 2004, pp. 683–697.

[7] S. Melnik, H. Garcia-molina and E. rahm, Similarity flooding: A versatile graph matching algorithm and its application to schema matching, in: *Proceedings of the 18th International Conference on Data Engineering* Washington, DC, USA, IEEE Computer Society, (2002), 117–128.

[8] J. Huang, J. Dang, J.M. Vidal et al., Ontology matching using an artificial neural network to learn weights, in: *Proceedings of 20th International Joint Conference on Artificial Intelligence* Hyderabad, India, (2007).

[9] M. Granitzer, V. Sabol, K.W. Onn et al., Ontology alignment – a survey with focus on visually supported semi-automatic techniques, *Future Internet* **2**(3) (2010), 238–258.

[10] W. Yan, C. Zanni-Merk and F. Rousselot, Matching of different abstraction level knowledge sources: The case of inventive design, in: *Proceedings of 15th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems* Kaiserslautern Germany: Springer, (2011), 445–454.

[11] I. Muslea, Extraction patterns for information extraction tasks: A survey, in: *Proceedings of 6th National Conference on Artificial Intelligence Workshop on Machine Learning for Information Extraction*, (1999), 1–6.

[12] V. Ceausu and S. Desprè, A semantic case-based reasoning framework for text categorization, in: 6th *International The Semantic Web and 2nd Asian Conference on Asian Semantic Web Conference* Springer-Verlag: Busan, Korea, 2007, pp. 736–749.

[13] D. Maynard, A. Funk and W.W. Peters, Sprat: A tool for automatic semantic patternbased ontology population, in: *Proceedings of International Conference for Digital Libraries and the Semantic Web* Trento, Italy, (2009).

[14] Y. Sari, M.F. Hassan and N. Zamin, Rule-based pattern extractor and named entity recognition: A hybrid approach, in: *Proceedings of Information Technology (ITSim), 2010 International Symposium in*, (2010), 563–568.

[15] D. Ritze, C. Meilicke, O. Svá-Zamazal et al., A pattern-based ontology matching approach for detecting complex correspondences, in: *Proceedings of OM: CEUR-WS Org* (2008).

[16] O. Šá-Zamazal and V. Sváek, Owl matching patterns backed by naming and ontology patterns, in: *Proceedings of 10th Czecho-Slovak Knowledge Technology Conference*, Stara Lesna, Slovakia, (2011).

[17] K. Toutanova, D. Klein, C.D. Manning et al., Feature-rich part-of-speech tagging with a cyclic dependency network, in: *Proceedings of NAACL-Human Language Technology Conference*, Edmonton, Canada: Association for Computational Linguistics (2003), 173–180.

[18] D. Lin, An information-theoretic definition of similarity, in: *Proceedings of 5th International Conference on Machine Learning* Wisconsin, USA: Morgan Kaufmann (1998), 296–304.

[19] C. Fellbaum, WordNet and wordnets, in: *Encyclopedia of Language and Linguistics*, K. Brown, ed., Elsevier: Oxford, (2005), 665–670.

[20] W. Cohen, P. Ravikumar and S. Fienberg, A comparison of string distance metrics for name-matching tasks, in: *Proceedings of IJCAI-03 Workshop on Information Integration* (2003), 73–78.

[21] W.E. Winkler, The state of record linkage and current research problems, in: *Proceedings of: Statistical Research Division*, U.S. Bureau of the Census, (1999).

[22] M. Jaro, Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida, *Journal of the American Statistical Association* **84**(406) (1989), 414–420.

[23] M.A. Jaro, Probabilistic linkage of large public health data files, *Statistics in Medicine* **14**(5–7) (1995), 491–498.

[24] T. Pedersen, S. Patwardhan and J. Michelizzi, WordNet: Similarity: measuring the relatedness of concepts, in: *Proceedings of NAACL-Human Language Technology Conference* Boston, Massachusetts: Association for Computational Linguistics, (2004), 38–41.

[25] CSAIL-MIT, JWI (the MIT Java Wordnet Interface) 2012 [cited 2012 January], available from: http://projects.csail.mit.edu/jwi/.

[26] D. Hope, JWS (Java WordNet::Similarity), 2008 [cited 2011 December], available from: http://www.sussex.ac.uk/Users/drh21/.

[27] INRIA, Alignment API 2012 [cited 2012 January], Available from: http://alignapi.gforge.inria.fr/.

[28] OAEI, Ontology Alignment Evaluation Initiative(OAEI) 2011 Benchmarking Data Sets, 2011 [cited 2012 Febrary], Available from: http://oaei.ontologymatching.org/2011/benchmarks/.

[29] J. Euzenat, A. Ferrara, W. Hage et al., Results of the ontology alignment evaluation initiative 2011, in: *Proceedings of the 6th International Workshop on Ontology Matching*, Bonn, Germany: CEUR Workshop Proceedings, (2011).

[30] J. Euzenat, Semantic precision and recall for ontology alignment evaluation, in: *Proceedings of 20th International Joint Conference on Artificial Intelligence*, Hyderabad, India: Morgan Kaufmann, (2007), 348–353.