Pattern-based core word recognition to support ontology matching

Fuqi Song, Grégory Zacharewicz, David Chen

Pattern-Based Core Word Recognition to Support Ontology Matching

Fuqi Song[1], Gregory Zacharewicz, and David Chen

Univ. Bordeaux, IMS, UMR 5218
351 Cours de la Libération, F-33400 Talence, France

Abstract
Ontology matching is a crucial issue in semantic web and data interoperability. In this paper, we describe a core word based method for measuring similarity from the semantic level of ontology entities. In ontology, most of labels of entities are compound words rather than single meaningful words. However, the main meaning usually is represented by one word, 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. Also, the other information is noted as complementary information. An algorithm is given to measure the similarity between a pair of compound words or short texts. In order to support diverse situation, especially when no core words could be recognized, non semantic based ontology matching techniques are applied from lexical and structural aspects 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; pattern recognition; core word

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 semantic correspondence between entities. Ontology matching has been studied for years, many matching techniques have been proposed. The purpose of all the work is trying to discover the matches from semantic 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 process (NLP), especially, information extraction (IE) to ontology matching. Ontology is usually created to represent domain concepts and relations. We noticed that the labels used for naming entities are alike natural language, normally, consisted with several single meaning words. These compound words or short phrases represent one core meaning, unlike the normal complete sentences, which contains several meanings. The main meaning is usually denoted in one or two words, which is called “core word”. Thus, if we could know the core word, it would be easier to find equivalent semantic correspondence. This is the base of our research work, form this point, we propose to use pattern recognition with part of speech (POS) to learn the core word, and measure the semantic similarity with core word and complementary information.

[1] Corresponding author. E-mail: fuqi.song@ims-bordeaux.fr

The hypothesis to apply the method is that the labels of entities in ontology should be alike natural language. For the situation with random generated strings or less meaningful compound word, the method is not 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 the correspondences from lexical and structural level of ontology.

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

2 Related work

2.1 Ontology matching

Ontology matching seeks to find semantic correspondences between a pair of ontology entities by identifying semantic relations. A definition of correspondence [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:

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

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 $n \in \Xi$. There are two types of correspondences for multiple matchers-based approaches: intermediate and final. Intermediate correspondence is discovered by a specific matcher, and then several intermediate correspondences are combined into a final correspondence. Namely, final correspondences are used for aligning, whereas intermediate correspondences are used for generating final correspondences.

The similarity-based approaches seek to discover the equal relation between ontology. Hierarchical relation, such as super class and child class, and the other relations are beyond the ability of similarity-based approaches. In this paper, we focus on discovering equal relation between ontology, the other types of relation are not considered.

The entities to be matched most commonly in ontology include: class, instance and property. In some approaches, more information of ontology is adopted, such as, in [3], data type and value are used to calculate the similarity. However, usually this kind of information plays a role of complementary information to support match the above three entities. In this article, this information is not investigated.

The authors of [2, 4, 5]gave comprehensive introduction and comparison of different basic matching techniques and applications. In [6], the authors classify the ontology matching approaches by considering the three levels of source ontology to be matched. At element level, the class itself is treated as the studied object; the label, comment and internal information of it are investigated. The most used techniques are string metric [7], string similarity, domain, property and data type comparison [8], etc. At local level, the objects and the relations linked to the studied element are taken into account, such as similarity flooding [9], graph-based, taxonomic-based, etc. At global level, the whole ontology is taken as a context and environment, the relation and affect between it and the studied object is

investigated, machine learning, artificial neural network [10] are some methods applied at this level.

2.2 Information Extraction (IE)

"*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 a certain pre-defined rules. NER is a subtask of IE. NER seeks to find and recognize the atomic elements in text. For example, "*the book title*" will be recognized as *the (article) book (noun) title (noun)*. The recognition rules are various, in this example, we identify by the part of speech of words.

Some related work in this area is listed in table 1. In [11], the author investigated the different extraction patterns in information extraction (IE). [12-14] applied extraction patterns to free text and documents. In [12, 14], the patterns are focused on specific information, such as the date and location, which are important in accident report. [13] used patterns to extract and create ontology from free text. It could build semantic relations in ontology. [15, 16] adopted patterns to perform ontology matching for discovering complex correspondences, which are in the relevant research domain to ours. They define a set of patterns from the one or several related entities in ontology and use the pattern to find correspondences. The patterns are learnt from the mostly used forms when creating ontology. In our work, we try to obtain the core word, which represents the main meaning of a compound work or short text, by applying the certain patterns. These patterns are defined based on POS and linguistics. We don't use pattern directly to find the matches, rather a way to find the core word.

Table 1. Investigation of IE and NER based approaches in ontology matching domain

Author(s)	Type	Pattern recognition	Source	Application
(Muslea, 1999) [11]	survey	-	-	-
(Ceausu, 2007) [12]	framework	POS-based	Accident report	Text categorization Accident report
(Maynard, 2009) [13]	tool	NER Hearst pattern Lexical-syntactic pattern Contextual pattern	Free text	Ontology extraction ontology creation
(Sari, 2010) [14]	method	Date and time Location Accident effect	Free text, document Structured documents	Creating extraction pattern
(Ritze, 2008) [15]	method	Class by Attribute type pattern (CAT)….	Ontology	Detecting complex correspondences
(Svab-zamazal, 2011) [16]	theory	NER	OWL ontology	Ontology matching

3 Matching with Pattern-based Core Word (PCW)

Core word is one or more words, which represent the main meaning in a compound word or short phrase. A process (Figure 1) is proposed to measure similarity confidence between two compound words or short phrases. A pair of compound words or short phrases is as input. First, the stop words and unnecessary information are eliminated from label, and then the

string is tokenized into single words. With pre-defined patterns, the short text is recognized into each catagory. 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. Details of each step are illustrated in following sections.

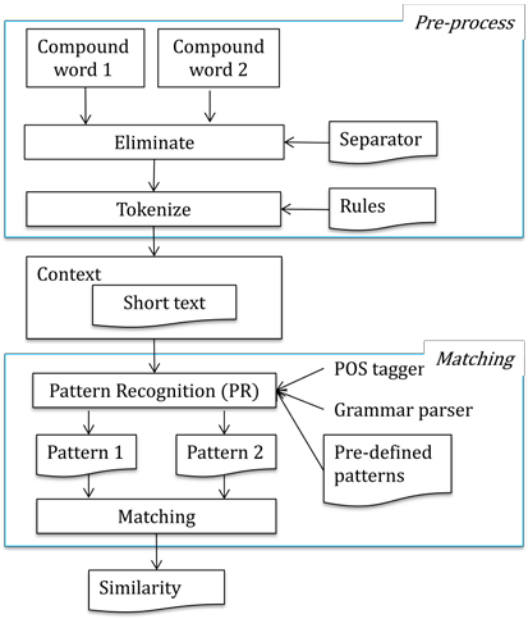

Figure 1. Process of core word-based similarity measurement

3.1 Elimination and tokenization

Before pattern recognition, elimination and tokenizatioin core words are performed as pre-process. 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 makes the compound word split into single ones. The compound word is tokenized by rules: 1) stop words, e.g. dash, underscore, and dot; 2) capitalized word, such as “numberOfTelephone”.

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 carrier of text. The labels commonly follow specific rules and most of them are compound words or short phrases. Usually verb-based label 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, we believe that certain patterns could be concluded from source ontology labels. Unlike complete phrases, the label concentrates on representing one simple meaning. So it is important to find out which word is the core word. It is helpful to understand the whole meaning.

The types of part-of-speech (POS) used in the approach are listed in table 2. To tag the POS of words, we use postagger [17] from Stanford University. Mainly nouns, verbs, adjectives and part of preposition are tagged. The words with the other POS are ignored, such as article and conjunction, because they don’t represent much real 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 tense and participle. In prepositions, only “of” and “by” are tagged, and 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 edited book. For adjectives and this kind of verb, they are called as “modifier” in general.

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

In order to obtain the patterns mostly used, we studied some real-life ontology and experimental ontology. The most commonly used patterns are concluded in table 3. The first column shows the composition mode of word, 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, the passive tense, etc. The representation of these information is denoted as *(core word, <type, complement info. 1, type, complement info. 2, …>).* For example, *(conduct, < form, pass>)* denotes that the core word is “*conduct*” with a passive form.

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 multiple-noun mode. For example, the core word of “*titleOfBook*” is “*title*” and the core word of “*titleBook*” is “*book*”. “*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.

Table 3. Recognition pattern and core word

		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*

3.3 Semantic similarity measuring with PCW

Two similarity measuring methods are used in Semantic matching (SMA). Lin model [18] is a reused and adapted method. We propose a homonym checker to solve homonym issues in semantic matching.

Lin model [18] is a taxonomy-based model for measuring semantic similarity. In Lin model, the taxonomy is taken as a tree, WordNet [19] is used as the taxonomy. It returns a semantic similarity by measuring communality between two words in the taxonomy tree.

Table 4. Examples of patterns and core words

	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 applyWith	submit apply	conference -
	Passive form	writtenByAuthor	write	author

Homonym checker: Homonym is a special case in semantic matching. The same word doesn't always represent the same meaning. It represents different meanings in different contexts, such as "*article*" may refer to a publication or refer to a product. First we measure whether the two ontology, where the homonyms occurred, belong to the same context. A semantic similarity indicator $\boldsymbol{I_s}$ helps to examine whether they belong to the same context. I_s is computed based on the identified core words, not the original labels. For a word in ontology O_1, if there is a synonym existing in O_2, then *#synonym* count adds 1. It is defined as

$$I_s = \#synonym/min(\#tcp_1, \#tcp_2), \qquad (1)$$

in which *#synonym* is the number of synonyms identified between O_1 and O_2, *tcp* is the number of total concepts and properties.

A threshold *th* is set, if the indicator I_s is greater than the threshold *th*, then the two ontology are considered as belonging to same context. In this case, the two words are considered to represent the same meaning and the similarity is assigned as 1.0. Otherwise, a formula

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

is applied for computing the similarity of a pair of homonyms. In the equation, #m (meaning) is the number of different explanations (retrieved from WordNet) the word has.

An overall similarity measurement between two entities $e_1$and e_2 of *SMA*(eq. 3) is as

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

In order to measure the similarity between two short text. 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, the label "_theShortTitle_OfBook" generates 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 >> JJ-NN-IN-NN >> 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 text with above given format is based on SMA (eq. 3). SMA aims to measure the similarity between a pair of single words in a semantic context. PCW (eq. 4) utilizes SMA as a component of the algorithm. There are two parts involved: core word part M1 (eq. 5) and complementary information part M2 (eq. 6). We consider that core word is more important in representing the semantic, the weight of M1 and M2 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,…} of *e* with length *l*. ci1k is one arbitrary element of set CI1. If two inputs share the same core word and one complementary word, the confidence equals 1. Otherwise, the similarity is accumulated based on each pair of them.

$$PCW(e_1, e_2) = 0.7 * M_1(cw_1, cw_2) + 0.3 * M_2(CI_1, CI_2) \quad (4)$$

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

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

4 Non-semantic based matching and aggregation

From non-semantic aspects to perform ontology elements matching is important, since source ontology usually has complex situations. There are string-based and structure-based matching techniques, which are regardless the meaning of the elements represented. Two matchers are used in the approach: 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 elements are treated only as a sequence of letters, without considering the meaning represented and structure contained. We use string metric, which measure similarity or distance between two plain strings. Distance functions map 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. We will use a broadly string metric Jaro-Winkler [21] distance proposed by Winkler based on Jaro distance [22, 23]. Jaro distance is defined as

$$\mathrm{Jaro}(\mathrm{e}_1, \mathrm{e}_2) = \frac{1}{3} * \left(\frac{\mathrm{m}}{|\mathrm{e}_1|} + \frac{\mathrm{m}}{|\mathrm{e}_2|} + \frac{\mathrm{m}-\mathrm{t}}{\mathrm{m}}\right) \quad (7)$$

where e_1 and e_2 are string from O_1 and O_2, *m* is the number of matching character, *t* is half of the transportation number. Two characters are matched only when the distance is not beyond the matching window, i.e. take 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 $g=\max(|s_1|,|s_2|)/2 - 1$. Jaro-Winkler distance added a weight for common prefix, defined as,

$$\mathrm{ED}(\mathrm{e}_1, \mathrm{e}_2) = \mathrm{Jaro}(\mathrm{e}_1, \mathrm{e}_2) + \frac{\min(\mathrm{P},4)}{10} * (1 - \mathrm{Jaro}(\mathrm{e}_1, \mathrm{e}_2)) \quad (8)$$

where P is the length of longest common prefix of e_1 and e_2. min(P,4)/10 for assuring the coefficient not exceeding 0.25, which may cause consequently $ED(e_1, e_2) > 1$.

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

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

4.2 Directed graph (DG)

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

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

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 (#indegree1, #indegree2)/ max (#indegree1, #indegree2). If both of values are equal to 0, then *inR* = 0. In figure 2, an illustration 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.

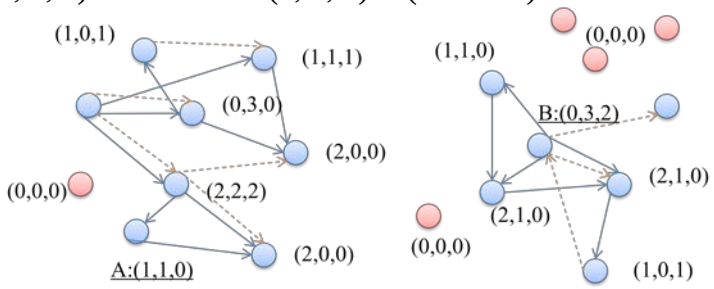

Figure 2. Example of directed graph representing ontology

4.3 Aggregation

So far the matching techniques have been described. In order to select and aggregate them, a flow process is given in figure 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, such as both entities are classes.

With pattern-based core word identification, we examine whether there is a core word existing. If core word is identified, then we will use PCW matcher to match. Otherwise non-semantic matchers ED and DG will be applied. Each of them will take up 50% weight. 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*, we consider that no core word has been recognized.

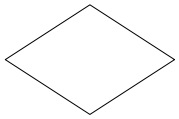

Figure 3. Aggregation process

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, postagger [17] for identifying core words. The java APIs used in implementation are JWI [25], JWS [26] and Alignment API [27]. First, we use a pair of real ontology to illustrate the proposed matching method, and then perform benchmarking test with test cases of OAEI [28].

5.1 Illustrative case

First we use an ontology *EKAW* to test the pattern recognition. The ontology is available at http://oaei.ontologymatching.org/2012/conference/data/ekaw.owl, which contains 74 classes and 33 object properties. The ontology is in the domain of conference and publication. There are total 106 elements recognized, and we keep part of the results without changing. In table 6, there are original labels, identified patterns, core word and complementary information.

Most of the elements 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, such as “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. Manually, we count the incorrectly identified core word and patterns regarding to their real semantics. There are 9 misidentified patterns out of 106, the precision is 91.5%.

We use another real ontology *OpenConf*, which is also in the domain of conference organization and available at http://oaei.ontologymatching.org/2012/conference/data/OpenConf.owl, to perform ontology matching. In *OpenConf*, there are 61 classes, 21 datatype properties and 24 object properties. We obtained 106 correspondences, with threshold = 0.7 (set manually), there are 24 correspondences filtered as shown in table 7.

Table 6. Pattern and core word recgonition with ontology

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. Benchmark data set biblio

Entity in *EKAW*	**Entity in *OpenConf***	**Similarity**
Demo_Chair	Program_chair	0.74
Document	Text	0.86
Event	Result_of_Advocate	0.91
Industrial_Paper	Paper	0.70
OC_Member	Member	0.75
PC_Member	Member	0.75
Paper	Paper	1.00
Paper_Author	Contact_Author	0.75
Research_Topic	Domain_Topic	0.71
SC_Member	Member	0.75
Scientific_Event	Result_of_Advocate	0.70
Session_Chair	Program_chair	0.71
Social_Event	Result_of_Advocate	0.70
Submitted_Paper	Submitted_Paper	1.00
Tutorial_Abstract	Conference_Program	0.72
Tutorial_Chair	Program_chair	0.70
Workshop_Chair	Program_chair	0.70
Workshop_Paper	Paper	0.70
hasEvent	has_Result	0.86
hasPart	has_made_review	0.83
hasReview	has_Review	1.00
hasReviewer	has_Review	0.75
hasUpdatedVersion	has_Result	0.82
reviewWrittenBy	is_written_by	0.78
…	…	…

5.2 Benchmarking

The data set for experiment is from OAEI 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 4 test cases T1 to T4 as table 8. Test case T1 contains 3 ontology with small changes in labels and structure. Test case T2 contains 10 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: 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, and 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 set of alignments identified by our approach is denoted as A_d, and the set of reference alignments is denoted as A_r. Then the measurements are denoted as,

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

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 high, and the average precision is 0.83. Recall of test cases T1 and T3 are 1. 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.

Table 9. Results of 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

5.3 Discussion

The aim of PCW is to identify core words from natural language alike compound word or 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. Such as "*MS word*", which is should be a name of word processing software, but WordNet cannot identify correctly the meaning. A solution to this issue is to define a special name list, which contains the unusual meanings and uncommon words, for example "*PDF*", "*MS word*", etc. Then assign these names with a commonly used equivalent concept, such as 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 depend on the language environment. So this also allows the room to improve and extend the patterns in order to adapt to different situations.

6 Conclusion

In this paper, we described a pattern-based approach to recognize the core word of compound word. 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. We apply this method 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. We think that the method could also be applied to support these areas and have certain contributions.

References

[1] Song F, Zacharewicz G and Chen D. 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] Euzenat J andShvaiko P. *Ontology matching*. Heidelberg: Springer, 2007, p.341.

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

[4] Granitzer M, Sabol V, Onn KW, et al. Ontology Alignment—A Survey with Focus on Visually Supported Semi-Automatic Techniques. *Future Internet* 2010; 2(3): 238-258.

[5] Yan W, Zanni-Merk C and Rousselot F. 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, pp.445-454.

[6] Song F, Zacharewicz G and Chen D. An ontology-driven framework towards building enterprise semantic information layer. *Advanced Engineering Informatics* 2013; 27(1): 38-50.
[7] Stoilos G, Stamou G and Kollias S. A String Metric for Ontology Alignment. In *Proceedings of 4th international conference on The Semantic Web*. Galway, Ireland: Springer, 2005, pp.624-637.
[8] Ehrig M andStaab S. QOM – Quick Ontology Mapping. In: McIlraith SA, Plexousakis D and Harmelen Fv (eds) The Semantic Web – ISWC 2004. Heidelberg: Springer, 2004, pp.683-697.
[9] Melnik S, Garcia-Molina H and Rahm E. 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, pp.117-128.
[10] Huang J, Dang J, Vidal JM, 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.
[11] Muslea I. 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, pp.1-6.
[12] Ceausu V andDesprès S. *A semantic case-based reasoning framework for text categorization*, in *6th international The semantic web and 2nd Asian conference on Asian semantic web conference*. 2007, Springer-Verlag: Busan, Korea. p. 736-749.
[13] Maynard D, Funk A and Peters W. W.: 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] Sari Y, Hassan MF and Zamin N. Rule-based pattern extractor and named entity recognition: A hybrid approach. In *Proceedings of Information Technology (ITSim), 2010 International Symposium in*, 2010, pp.563-568.
[15] Ritze D, Meilicke C, Sváb-Zamazal O, et al. A Pattern-based Ontology Matching Approach for Detecting Complex Correspondences. In *Proceedings of OM*: CEUR-WS.org, 2008.
[16] Šváb-Zamazal O andSvátek V. OWL Matching Patterns Backed by Naming and Ontology Patterns. In *Proceedings of 10th Czecho-Slovak Knowledge Technology Conference*. Stara Lesna, Slovakia, 2011.
[17] Toutanova K, Klein D, Manning CD, 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, pp.173-180.
[18] Lin D. An Information-Theoretic Definition of Similarity. In *Proceedings of 5th International Conference on Machine Learning*. Wisconsin, USA: Morgan Kaufmann, 1998, pp.296-304.
[19] Fellbaum C. *WordNet and wordnets*, in *Encyclopedia of Language and Linguistics*, Brown K, Editor. 2005, Elsevier: Oxford. p. 665--670.
[20] Cohen W, Ravikumar P and Fienberg S. A comparison of string distance metrics for name-matching tasks. In *Proceedings of IJCAI-03 Workshop on Information Integration*, 2003, pp.73-78.
[21] Winkler WE. The state of record linkage and current research problems. In *Proceedings of*: Statistical Research Division, U.S. Bureau of the Census, 1999.
[22] Jaro M. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association* 1989; 84(406): 414-420.
[23] Jaro MA. Probabilistic linkage of large public health data files. *Statistics in Medicine* 1995; 14(5-7): 491-498.
[24] Pedersen T, Patwardhan S and Michelizzi J. WordNet::Similarity: measuring the relatedness of concepts. In *Proceedings of NAACL-Human Language Technology Conference*. Boston, Massachusetts: Association for Computational Linguistics, 2004, pp.38-41.
[25] CSAIL-MIT. *JWI (the MIT Java Wordnet Interface).* 2012 [cited 2012 January]; Available from: http://projects.csail.mit.edu/jwi/.
[26] Hope D. *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] Euzenat J, Ferrara A, Hage WRv, 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] Euzenat J. Semantic precision and recall for ontology alignment evaluation. In *Proceedings of 20th International Joint Conference on Artificial Intelligence*. Hyderabad, India: Morgan Kaufmann, 2007, pp.348-353.