SLADO

Semantic Lexical Alignment for Domain-specific Ontologies

Ahmad Adel Abu-Shareha, Rajeswari Mandava, Dhanesh Ramachandram
Computer Vision Research Group, School of Computer Science
Universiti Sains Malaysia
Penang, Malaysia
{adel, mandava,dhaneshr}@cs.usm.my

Abstract— **Domain-specific ontologies encode reusable domain vocabulary and represent established domain semantics. The alignment of such ontologies requires an approach based on a semantic analysis of its components. This paper presents SLADO, Semantic Lexical Alignment for Domain-specific Ontologies. The proposed approach aims to use the available dictionaries and lexical resources in the underlying domain and to mine the ontology information to extract words from the string features of the input ontologies. The string features are processed as word wrappers, where every string is assumed to contain some word(s). The architecture of SLADO uses a mixture of lexical and knowledge-based methods in a semantic approach. It first identifies the semantic words from the ontology labels, and then determines the output by comparing the extracted words and their synonyms using lexical resources such as WordNet.**

Keywords-component; Domain-specific ontologies, ontology alignment, lexical resources, word-based alignment

I. INTRODUCTION

An ontology is a conceptualization and representation of knowledge. Domain-specific ontologies encode reusable domain vocabulary representing established domain semantics. Domain specific ontologies have been created in many fields, such as the semantic web, artificial intelligence, gene representations and health care. Consequently, many ontologies have been created for the same domain. [1] Due to the variety of ontology design considerations and circumstances, besides the lack of formal restrictions on their engineering, heterogeneity problems have arisen among the different ontologies even for those representing a simple and tiny domain. Ontology alignment was born to overcome the drawbacks of such heterogeneity. Ontology Alignment is the process of developing a correspondence between two or more ontologies by identifying their identical elements. The alignment produces a dialogue that maximizes the interaction consistency between the underlying ontologies.[2]

Ontology alignment is carried out using two major approaches, the lexical and the structural. The lexical approach compares the strings associated with the nodes (labels, names, identity, etc) of each ontology, relying on the assumption that real objects have the same name in different contexts. The

structural approach matches the nodes based on their adjacency relationships. The structural method must be initialized with a set of identical pairs; based on these pairs, other pairs can be matched by comparing their affinities to the initial pairs. The relationships (e.g., subClassOf and is-a) that are frequently used in the ontology serve as the foundation of the structural matching.

Each of these methods has advantages and disadvantages: the output results from the structural methods might contain many errors and create inconsistency problems, whereas the output results from the lexical alignment are incomplete because such alignment is limited by the naming mechanism. The ontology developers might choose two different names or expressions for the same thing. Thus, the lexical methods might neglect some actually identical elements.

Looking on the bright side, the structural alignment is more comprehensive in that it is able to align identical elements with several names, while the lexical alignment is safer in that it deals with names, which are mostly consistent. [3], [4] This paper presents SLADO, Semantic Lexical Alignment for Domain-specific Ontologies. SLADO is an alignment approach that combines safety (by using lexical methods semantically in a novel word-base mechanism) and comprehensiveness (by using the available resources in the domain of interest to match identical elements with different names which also can be categorized as lexical method). Thus, the implemented approach doesn't use any structural method.

The rest of the paper is organized as follows. Section 2 highlights the lexical alignment approach, Section 3 discusses previous work, and Section 4 presents the proposed system. The implementation details and the results are discussed in Section 5 and Section 6 respectively. Finally, we present our conclusion in Section 7.

II. LEXICAL ALIGNMENT

The lexical alignments of interest to us can be classified into two categories, syntactic-based and semantic-based. The syntactic-based alignment compares the strings as sequences of ordered characters, and sets the output similarity based on the similarities or dissimilarities of these characters. The syntactic approach for ontology alignment relies on the string distance, which uses exact and approximate string matching methods.

The exact matching methods compare the input strings and produce as output either "true" if the input strings are exactly the same, or "false" otherwise. Thus, these methods are considered harsh and are hardly ever used. The approximate matching method is used often because of its flexibility in dealing with slight differences in the input strings. The slight differences between words are assumed to reflect word variations such as corruption, noun affixing and prefixing, compound words and verb tenses. The output of the approximate method is the ratio of the similar/dissimilar characters in the input strings to the total number of characters. In some cases of different words that have slightly different string representations, the approximate method fails to capture the dissimilarity. [5], [6]

The semantic approach compares the strings as words or dictionary entities. The semantic alignment depends on natural language processing methods, in which the words mostly lemmatized (reduced to its base meaning), and then the lemmas are compared. The problem with the existing semantic approach is its inability to process corrupted words, compound words, spelling reform and sometimes misspelled words.[7]

Recently, a new approach has been developed that can be classified under the lexical category. The new approach uses background knowledge to set up semantic matching. The knowledge that can be used in this context has to be general and comprehensive. Based on the background-based approach, the similarity of two input strings depends on their relationships and positions in the background knowledge. [8] The background-based method has, to some extent, made the lexical approach a more comprehensive approach.

Each of the previously discussed approaches has its own advantages. The new method, SLADO, shares most of the advantages of each. SLADO is as flexible as the approximate methods that it uses to extract words from strings, as accurate as the semantic methods, and as comprehensive as the background-based approach since it makes use of the available resources in constructing the alignment.

III. RELATED WORK

Based on the two major approaches to ontology alignment, the existing alignment systems can be classified into lexical-based, structural-based and hybrid (lexical and structural). As mentioned earlier, structural alignment depends on identifying initial identical pairs, which normally are produced using some lexical method. Thus, the hybrid is the most commonly used approach. Pure structural systems are rare; they use user input or learning mechanisms to identify the initial pairs. Lexical-based systems mostly adopt a combination of syntactic (string shape) and semantic (string meaning) methods, with priority given to the syntactic method because of its flexibility and ability to process any arbitrary string.

As a pure lexical based, John Li [9] proposed a system that implemented a combination of several lexical methods: whole term matching, word constituent matching, synset matching using WordNet and type matching. The synset and type matching represent the semantic part: synset matching matches different words with equal meanings, while type matching matches words that belong to the same type. These semantic methods process words that are extracted directly from the strings or their constituents. The word constituent method depends on the internal punctuation delimiters that appear in the string, such as white space and dashes. This system performs well due to the variety of the implemented methods. However, the ontology developers might not use such boundary delimiters, and compound words are sometimes written without punctuation. Besides, word variations might also appear. In such cases, more practical approaches are required for the fragmentation and processing of the input string.

Zharko et al. [10] used background knowledge to align poor structure input ontologies. The proposed alignment substituted the single matching step (aligning the input ontologies) with two matching steps (aligning each of the input ontologies to the background ontology). This increased the accuracy of the matching for specific types of data, namely patient information in intensive care units. In the Cupid system [11], the linguistic matching consists of three phases: normalization (which includes tokenization, expansion and elimination using a thesaurus), categorization and comparison (string based comparison).

Previous works on ontology alignment have focused either on directly performing syntactic comparisons among the input strings, or on performing a semantic comparison after some shallow preprocessing such as lemmatization and affix elimination. However, the properties of the strings in the ontologies might have huge variations (verb tenses, spilling reform, compound words and numbering), which have not previously been addressed in the ontology alignment. Semantically, such cases cannot be solved using only shallow preprocessing. Syntactically, the difference between a misspelling and the correct word (winner, winer) is not easily distinguished from the difference between two distinct words with slightly different shapes (winner, winter). A rich set of alignment systems can be found in the proceedings of EON 2004-2008 [12].

A part away from lexical-based alignment in the schema-based category, some motivation for this work arises from the research on instance-based alignment. The instance-based alignment is limited to the lexical approach wherein the matching depends on analyzing the instance texts. D. Fossati et al. [13] implement four different algorithms that used natural language processing for instance-based ontology alignment. The processing steps includes tag removal, tokenization, lowercase conversion and other advanced NLP methods such as part of speech, synset matching, etc. Lexical-based approaches achieve more precision in the instance-based than schema-based ones, because the texts are being interpreted as meaningful text rather than arbitrary stings.

In this paper, a comprehensive lexical system, SLADO, is presented. The developed system processes the input strings to identify the meaningful words prior to the matching process. The advantages of using the pre-processing is to eliminate the unexpected varying in the string's shape, and to extract the individual components of compound words and corrupted phrases using the available dictionaries and background knowledge sources such as WordNet [14].

IV. SLADO

The architecture of the proposed system uses a mixture of lexical and knowledge-based methods in a semantic system. The lexical methods process the input elements' strings prior to the matching step. These methods jointly aim at identifying words in the input strings, relaying on the dictionary and on the accumulated extracted words from these strings. The knowledge-based part uses lexical resources to retrieve the relationships between words. Up to this point, WordNet is used for synonym retrieval.

The SLADO architecture, as illustrated in Fig. 1, has three processing levels. The system begins by collecting and arranging the string properties from the input ontologies. Then, a word extraction process takes place which aims to extract words from the previously extracted strings. Finally, the system semantically compares the strings of the ontology elements based on their word-base.

The word extraction process which formulates the backbone of SLADO is a novel sophisticated combination mechanism that uses several string distance methods and dictionaries. The string distance method is responsible for identifying strings and substrings in the inputs that have exact and approximate similarities with some dictionary entities.

The dictionaries used in this process are classified into multi-purpose dictionaries and domain-specific dictionaries (if available). The ontology-words dictionary is an extra dictionary that is built during the extraction process by mining the words that are extracted sequentially from the ontology elements. The ontology-words dictionary recurrence information is used for confidence calculation, based on the assumption that words in the ontology are more likely to be repeated. For example, the extracted word 'parent' can be used to deconstruct the string 'hasparent' into its true component words 'has' and 'parent' not 'has' 'pa' 'rent'.
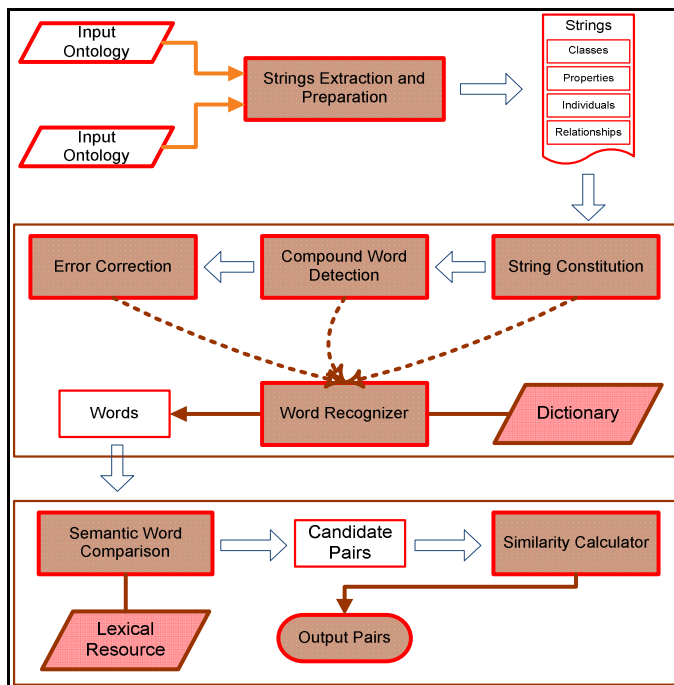

Figure 1. SLADO architecture.

In the matching process, the comparison process carried out over the extracted words prioritizes the normal words over stop words, distinguishes between repeated words and unique ones, and gives more weight to the base words than to other information, such as numbers. In general, the comparison method as implemented helps to meet the following goals:

- Words that have corresponding dictionary inputs can be matched to each other only through exact matching and synonyms in WordNet.

- Arbitrary strings and words can be matched to each other based on the misspelling assistance; strings that have no entry in the dictionary are assumed to be the results of corruption such as typing mistakes.

- Compound words components are matched to both arbitrary strings, via misspelling assistance, and words, via exact matching and synonyms in WordNet.

- The component of compound strings are matched to each other via synonyms in WordNet and exact matching, if the components are words, and otherwise, based on misspelling assistance.

- The rest of the unmatched strings are matched using Edit-Distance through the error correction mechanism.

A. Word extraction

The proposed mechanism deals with strings as word wrappers, with every string assumed to contain some word(s). The word extraction process, which is the cornerstone of the proposed system, relies on the assumption that ontology encodes knowledge rather than raw data. Thus, the ontology elements' labels refer to existing real world entities, which are normally included in the dictionary. This assumption is extended to characterize cases wherein the strings in the ontology might appear as arbitrary strings, which must be processed before being matched to return them to their original state. These cases include the following:

- Corrupted and misspelled words.

- Compound words with no internal punctuation delimiters.

- Combination of words and numbers.

- Combination of corrupted words and correct words.

Word extraction processing is achieved using a set of methods, which form the backbone of the system. Due to the flexibility of the implemented system, these methods can be extended, modified and enriched with other methods in the future if necessary.

1) Preprocessing: In the pre-processing step, as illustrated in Fig. 2, the strings are extracted from the ontology elements and arranged in groups based on the types of their underlying elements. Anonymous classes and individuals with no names/labels are discarded at this stage.

2) String constituent: The string constituent step segments the string based on any clear and existing internal punctuation delimiters, such as spaces, underscores, etc. Numbers and other special characters are also extracted and saved at this stage.
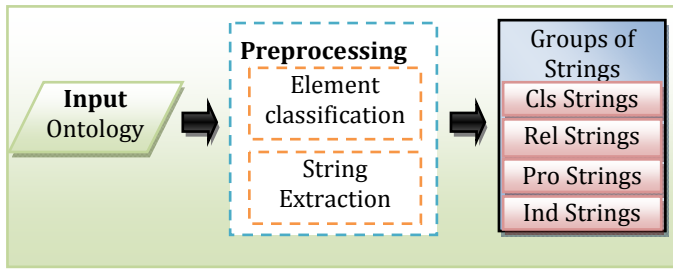
Figure 2. Pre-processing in SLADO

3) Word recognizer: The directed word recognizer extracts the meaningful words that have corresponding dictionary entries using dictionary lookup and exact matching against the dictionary entities, as illustrated in Fig. 3. Besides serving as base words for the matching process, the directly extracted words constitute the ontology-words dictionary that is later used in the word extraction method to weight the different words in multiple choice solutions.

4) Lemmatization: Each detected word is lemmatized to eliminate the mismatching that is caused by word variations.

5) Compound word analyzer: Generally, the Compound Word Analysis and Word Segmentation are used for processing the natural languages that do not use spaces as boundaries between words, such as Chinese and Thai. [15], [16] In other languages, such as English, words are compounded together with no boundaries to express specific meanings (e.g., housewife) [17]. In such cases, word constituent analysis and word identification are required. The word identifier in such cases uses the dictionary and some probability theory to identify the component words.

Ontology developers tend to express meanings using nouns, noun phrases and compound words. The compound words might or might not have clear punctuation delimiters (capital letters, underscores, dashes, etc) as boundaries between their component words. Thus, identifying the component words of such combinations requires word detectors. There may be several possibilities for the identified components, with or without slight editing. At this point, we consider the splitting of compound words with no further editing. The edited splitting process is taking place in the error correction method.

The compound word analyzer has a set of steps to detect the words that might be juxtaposed. The analyzer divides the input string into multiple words without any extra steps (insertion, deletion, change, substitute or re-ordering), using the multipurpose dictionary entities as illustrated in Fig. 4.
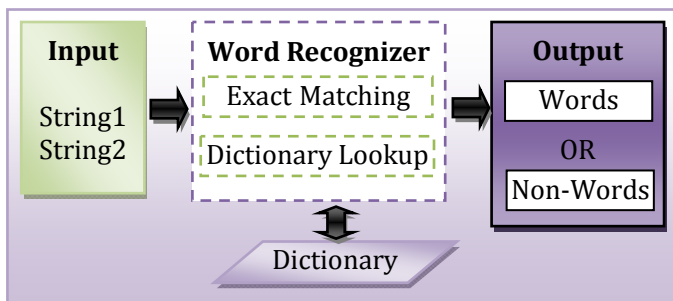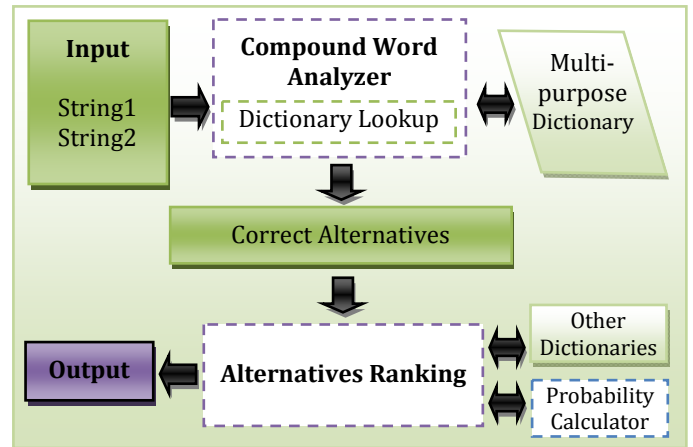

Figure 3. Word recognizer in SLADO

Figure 4. Compoun-word analyzer in SLADO

If there are multiple correct choices, then the choice that has entries in the other dictionaries (the domain-specific dictionary and the ontology-words dictionary) is considered based on probability calculation. The most common way to calculate the confidence value for the suggestions produced by the error correction in the literature is to use Bayesian probability, as in (1).

$$p(c|w) = p(w|c)p(c). \qquad (1)$$

where, $p(c|w)$ is the probability of the suggestion 'c' given the corrupted word 'w'. $p(w|c)$ is the error model representing the confidence that the suggestion represents the misspelled word. This confidence normally stands for the similarity value produced by the utilized approximate string distance method. $p(c)$ is the language model which represents the reputation of the word in the corresponding language.

In SLADO, the language model depends on the entries of the ontology itself and the utilized dictionary. Thus, the language model is calculated based on (2), which combines the dictionary model represented as $p(w|D)$ and the ontology model represented as $p(w|O)$.

$$p(c) = p(w|D)p(w|O). \qquad (2)$$

As a result, the probability of each given suggestion in SLADO is calculated based on (3).

$$p(c|w)=p(w|c)p(w|D)p(w|O). \qquad (3)$$

The dictionary model is calculated as the ratio between the number of appearances of the word in the dictionaries and the total number of dictionaries used. At the moment, with a single domain dictionary, each suggested word can take either the value of ½ if it appears only in the multipurpose dictionary (the suggestions were extracted using the multipurpose dictionary, so each suggestion will at least appear in the multipurpose dictionary) or 2/2 if it appears also in the domain-specific dictionary. Thus, the proposed system can handle more dictionaries, especially domain–specific ones. These dictionaries will reflect the value of the words contained. The

words that appear in more domain-specific dictionaries are assumed significantly representing the domain.

The ontology model represents the reputation of the word in the input ontologies based on the ontology-words dictionary. The examined word is also counted to avoid zero value.

6) Error correction: The error correction technique processes the strings that have no corresponding entity in the dictionary. The error correction tries to find the closest dictionary entity (or entities) for the input string, using a similarity calculation that normally uses the approximate string matching method. As discussed in our previous work [18], the error correction technique provides a good enhancement to the alignment problem.

Like the compound word analyzer, the error correction technique might produce multiple results. These results are ranked based on the similarity value produced by the approximate string matching, ontology-words dictionary and the words in the domain-specific dictionaries using probability calculation in (4).

$$p(w_i) = avg \; max \; p(w_i|D) * p(w_i|O) \qquad (4)$$

The probability $p(w_i|D)$ is the dictionary model, similar to the dictionary model used in ranking the error correction suggestions, which is defined as the number of appearances of the word in the different dictionaries, divided by the number of dictionaries used. $p(w_i|O)$ is the ontology model, reflecting the fact that words that appear in the ontologies are more likely to be repeated in compound words. The word that is currently being tested is considered in order to avoid zero values.

Because the compound word contains more than a single word, the probability of each compound solution is the average probability of its components excluding the stop words.

7) Stop word recognition: The stop word recognition marks the words that are considered stop words. The stop words sometimes have lesser effects on the meaning of the noun phrase than the main noun. Thus, they have less significance on the comparison process. A list of English stop words is used in this process.

8) Synonym retrieval: Using external lexical resources such as WordNet for English, the synonym relationships between the extracted words are retrieved. We are only concerned with the bilateral synonym relationships. At each word extraction step, all the synonyms of the extracted word are retrieved. Then, the synonymous entities that have corresponding words in the ontology-word dictionary (which is updated sequentially) are saved. In such a way, only exchange synonyms for the extracted words are saved.

B. Matching

1) Word significance: The repetition of a word defines the significance of its word base. The less it is repeated, the more significant the word is, unless the repeated word appears on its own in the underlying ontology element, unaccompanied by other words or numbers, in which case it will be of high significance even if it is repeated with other words in other elements. Counting the repetitions of each word represents mining the ontology words to select the best and most appropriate matching. Notice that the role played by repetition here is the opposite of its role in the processing of the compound words.

2) Weighted matching: Finally, the matching process takes place, based on the base-words that were previously extracted. The initial selection of candidate pairs is based on matching the extracted words and their synonyms. In this initial matching, the ontology elements are intended to have multiple matching with each other. Thus, the final matching is performed over these matched pairs, by comparing the rest of the string components.

The final matching considers the following criteria for the initially matched pairs:

- The significance of each word.

- The similarity of the accompanying stop words if they exist.

- The similarity of the accompanying numbers if they exist.

These criteria are used to rank the candidate matchings for each element, and then to select the final matching.

The final alignment output can include many scenarios. The alignment dialogue between the ontology elements can be one-to-one, one-to-many and many-to-many, due to the variety of levels of expressiveness of words in most natural languages. It is desirable for a system to be able to address such variety. However, constructing such a system is a non-trivial task and requires an intelligent approach. In this work, we limit our experience to one-to-one dialogues. However, the framework is flexible enough to be expanded to process the other matching possibilities in future work. Moreover, the matching can be type oriented, in which only elements with the same type can be compared to each other (class vs. class, relationships vs. relationships, etc), or it can be generic, in which the elements are compared regardless of types. The technique as developed here can work with both mechanisms. However, the generic matching is more intuitive, since humans might address the same reality in different ways.

V. IMPLEMENTATION

SLADO has been implemented Using Java and the NetBeans 5.5 Integrated Development Environment. The implementation of the proposed system is based on some of existing APIs. The alignment API[1] is the foundation of the implementation. It provides a set of rich tools to construct, manipulate, and evaluate a variety of alignment methods [19]. The open source spell checker Jazzy provides a complete set of error correction methods [20]. The Java WordNet library JWNL is used as a dictionary reference[2] [21].

---

[1] http://alignapi.gforge.inria.fr/
[2] http://sourceforge.net/projects/jwordnet

VI. RESULTS

The data sets used for the experimental results are the data set provided by EON [12] in the EON 2008 Ontology alignment contest[3]. The output results of the proposed system are compared with the results of the leading systems participated in EON 2004, 2005, 2006 and 2008. The EON data set contains a variety of rich ontologies that have different variations in perspective. However, we have selected tests with relevant to our system capabilities, other tests that heart on other issues like foreign names have not been considered.

In the provided test, a single ontology, ontology 101, is aligned to another one, and the true alignments for these tests are also provided. Ontology 101, which is included in all the tests, describes bibliographic references. Each of the testing cases is numbered by the second input ontology number (the first input ontology is 101). In general, tests 1xx are simple tests, tests 2xx are advanced while 3xx are real case tests. Ontology 103,104 differ in their language generalization level, accordingly, those test the ability to handle language generalization levels. Ontologies 201 and 202 use anonymous elements with no names. Test 204 examines the string conversation handling by using capital letters and special characters like underscores. Ontology 205 includes words that are synonyms to those in Ontology 101. Tests 221, 222 and 223 examine the handling of structural variation. Ontology 230 expands the class components. Ontologies 301, 302, 303 and 304 are real ontologies representing bibliographic references in the computer science field. Tests with high similarities to those mentioned above have not been considered in the discussion. As shown in the result table (Table 1), wherein the proposed system is compared to the top systems in the EON 2008 (ASMOV, Lily and RiMOM) [20], the proposed system has achieved high precision and high recall for the provided tests.

In Table 1, in tests 103 and 104, full recall, is tested with simple, similar names. For 201 and 202 the alignment dialogue is null because no names are used. 204 is a naming conversation test, and the result of this test is encouraging; an example of a good result is matching 'MSc_thesis' with 'MasterThesis'. 205 is a synonyms test, the implemented system shows good results based on the retrieved synonyms from WordNet, in which, for example, 'frequency' and 'periodicity' are matched. The tests 221, 222 and 223 achieve full recall and precision. Some of the matched elements deal with the definitions of words (synonyms), such as 'Book' with 'Reference' and 'Article' with 'journalPart'.

The strength of the proposed system can be observed in the tests 230, 301, 302, 303, 304 which represent naming conversation and synonyms, for example, 'Organization' with 'organizationName', 'annote' with 'hasAnnotation', 'Chapter' with 'Inbook', 'reference' with 'entry', 'address' with 'location', 'name' with 'eventTitle' and 'Book' with 'inChapter', this shows the strength of using word-base as a basic for semantic alignment, whereas in such cases, sole synonyms matching, syntactic matching or semantics cannot find the desirable matching. As observed from the table, the proposed system has stability characteristics with the different types of tests (except those have no labels).

---

[3] http://oaei.ontologymatching.org/2008/

TABLE I. PRECISION AND RECALL OF ASMOV, LILY, RIMOM AND THE DEVELOPED TECHNIQUE FOR THE ALIGNMENT TESTS

Test #	Alignment Approach							
	ASMOV		Lily		RiMOM		SLADO	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
101	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
103	1.00	1.00	1.00	1.00	1.00	1.00	0.97	1.00
104	1.00	1.00	1.00	1.00	1.00	1.00	0.97	1.00
201	1.00	1.00	1.00	1.00	1.00	1.00	Nan	Nan
202	0.92	0.80	1.00	0.84	1.00	0.81	Nan	Nan
204	1.00	1.00	1.00	1.00	1.00	1.00	0.96	0.78
205	1.00	0.99	1.00	0.99	1.00	0.99	0.79	0.80
221	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
222	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.99
223	0.99	0.99	0.98	0.98	1.00	1.00	0.93	0.99
230	1.00	1.00	0.94	1.00	0.94	1.00	0.97	0.97
301	0.89	0.77	0.94	0.82	0.76	0.69	0.94	0.78
302	0.61	0.46	0.89	0.65	0.72	0.65	0.95	0.78
303	0.73	0.83	0.65	0.71	0.76	0.88	0.95	0.96
304	0.90	0.92	0.95	0.97	0.90	0.97	0.97	.096

Moreover, it is clear that the system is efficiently processing tests 301, 301, 303 and 304 which represent real ontologies. Thus, this proves that the semantically word-base approach developed in SLADO is appropriate for real alignment. Figure 5 shows the precision and recall of the proposed system and the participated systems in EON [20] for tests 301,302,303,304.

Finally, as illustrated, the developed system has performed well in handling naming variations, synonyms and word definition. Thus, it is suggested to enrich the system with more dictionaries and lexical resources on the one hand, and to add more methods to the system cycle on the other hand. However, the proposed system is not operating well over the rest of the tests which have not been addressed in our experimental because these tests concentrate on the structural aspects and which enclose no labels.
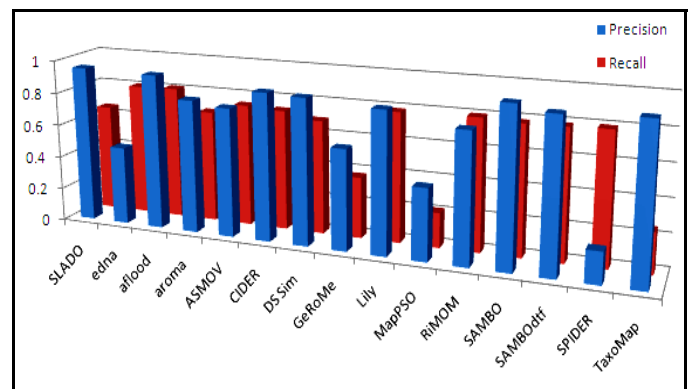

Figure 5. Comparative evaluation

VII. CONCLUSION

In this paper, we have developed a word-base alignment technique for ontology alignment, SLADO. The proposed technique is a semantic lexical-based approach that assumes that each string property in the ontology is a word wrapper and includes words in some way. Thus, the proposed technique extracts the words from the strings and compares the base-words. The experimental results, although limited, show the efficiency of the proposed technique. The proposed approach over roll the classical alignment methods based on blind string matching methods and sharp semantic-based methods. The implemented approach can be enriched with more methods in both of the word extraction steps, the dictionaries and matching steps.

REFERENCES

[1] Raj, S., Rajiv, K., and Ram, R., "Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems". Integrated Series in Information Systems, Vol. 14: Springer-Verlag New York, Inc., 2006, 930.

[2] Ehrig, M., "Ontology alignment: bridging the semantic gap", Springer-Verlag New York, Inc., 2007.

[3] Choi, N., Song, I.-Y., and Han, H., "A Survey on ontology mapping", ACM SIGMOD Record, 35, September 2006, pp. 34-41.

[4] Yi Li, J.T., Zhang, D., and Li, J., "Toward Strategy Selection for Ontology Alignment", the 4th European Semantic Web Conference, Austria, 3-7 June 2007.

[5] Petteri, J., Jorma, T., and Esko, U., "A comparison of approximate string matching algorithms", Softw. Pract. Exper., 26, (12), 1996, pp. 1439-1458.

[6] Navarro, G., "A guided tour to approximate string matching", ACM Computing Surveys, 33, (1), 2001, pp. 31-88.

[7] Euzenat, J., and Shvaiko, P., "Ontology Matching": Springer-Verlag New York, Inc., 2007.

[8] Aleksovski, Z., Kate, W.t., and Harmelen, F.v., "Ontology matching using comprehensive ontology as background knowledge", the International Workshop on Ontology Matching at ISWC, 2006, pp. 13-24.

[9] Li, J., "LOM: A Lexicon-based Ontology Mapping Tool", Performance Metrics for Intelligent Systems (PerMIS. '04), 2004.

[10] Aleksovski, Z., Klein, M., Kate, W.t., and Harmelen, F.v., "Matching unstructured vocabularies using a background ontology ", Knowledge Engineering and Knowledge Management (EKAW), 2006, pp. 182-197.

[11] Madhavan, J., Bernstein, P.A., and Rahm, E., "Generic schema matching with Cupid." 27th Intl. Conference on Very Large Databases (VLDB), , Rome, Italy, Sep. 2001, pp. 49-58.

[12] Caracciolo, C., Euzenat, J., Hollink, L., Ichise, R., Isaac, A., Malaisé, V., Meilicke, C., Pane, J., Shvaiko, P., Stuckenschmidt, H., Šváb-Zamazal, O.r., and Svátek, V.e., "Results of the Ontology Alignment Evaluation Initiative 2008", 3rd ISWC workshop on ontology matching (OM), Karlsruhe (DE), 2008, pp. 73-119.

[13] Fossati, D., Ghidoni, G., Eugenio, B.D., Cruz, I., Xiao, H., and Subba, R., "The problem of ontology alignment on the web: a first report", 2nd Web as Corpus Workshop In conjunction with the 11th Conference of the European Chapter of the Association for Computational Linguistics, Trento, Italy, 2006.

[14] FELLBAUM, C.e., "WordNet: An Electronic Lexical Database", MIT Press, 1998.

[15] Xin-Jing, W., Yong, Q., and Wen, L., "A search-based Chinese word segmentation method", the 16th international conference on World Wide Web, Banff, Alberta, Canada, 2007, pp. 1129 - 1130.

[16] Haruechaiyasak, C., Kongyoung, S., and Dailey, M., "A comparative study on Thai word segmentation approaches", 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. ECTI-CON, 2008, pp. 125-128.

[17] Rudolf, F., and Antonio, Z., "Spelling assistance for compound words", IBM Journal of Research and Development, 32, (2), 1988, pp. 195-200.

[18] Abu-Shareha, A.A., Rajeswari, M., and Ramachandram, D., "Two-way Dictionary-based Lexical Ontology Alignment", International Conference on Computer Engineering and Applications (ICCEA), Manila- Philippines, 6-8 June 2009, pp. 151-157.

[19] Euzenat, J., "An API for Ontology Alignment", International semantic web conference (ISWC), Hiroshima, Japan, 2004, pp. 698-712.

[20] SourceForge.net., "Jazzy - Java Spell Check API", pp.

[21] http://sourceforge.net/projects/jwordnet