On the Utility of WordNet for Ontology Alignment: Is it Really Worth It?

Uthayasanker Thayasivam and Prashant Doshi
THINC Lab
Department of Computer Science
University of Georgia
Athens, Georgia 30602
Email: {*uthayasa,pdoshi*}*@cs.uga.edu*

***Abstract*—Many ontology alignment algorithms augment syntactic matching with the use of WordNet (WN) in order to improve their performance. The advantage of using WN in alignment seems apparent. However, we strike a more cautionary note. We analyze the utility of WN in the context of the reduction in precision and increase in execution time that its use entails. For this analysis, we particularly focus on real-world ontologies. We report distinct trends in the performance of WN-based alignment in comparison with alignment that uses syntactic matching only. We analyze the trends and their implications, and provide useful insights on the types of ontology pair for which WN-based alignment may potentially be worthwhile and those types where it may not be.**

***Keywords*-ontology alignment; performance; WordNet;**

I. Introduction

Multiple ontologies often exist for overlapping domains. These need to be aligned in order to promote interoperability on the semantic Web. Several ontology alignment algorithms are now available [1], [2], [3], [4], [5] that utilize varying techniques to semi- or fully automatically generate mappings between entities in the ontology pair. One popular technique is the use of large lexical databases such as *WordNet (WN)* [6]. This enhances the traditional syntactic or string-based matching between the labels of entities with the ability to match words that could be synonyms, hypernyms, and in other lexical senses. Alignment algorithms utilize WN due to the potential improvement in recall of the alignment. This predicted improvement is reinforced by previous studies of using WN [7], which cite the improved recall to unconditionally recommend using WN in alignment.

However, we strike a more cautionary note on the utility of WN in ontology alignment. Although its use may improve recall, one trade off is that precision typically suffers. This has been studied by Mandala et al. [8] in the context of information retrieval with the revelation that WN's significant negative impact on precision cannot be ignored while deciding on its use. Additionally, in contrast to the previous studies [7], [9], we consider the increased computational expenditure in the form of execution time as well while evaluating the performance gains. We think that execution time is a critical component of the evaluation because automatically aligning ontologies is computationally intensive, which is exacerbated as the ontologies become larger. *While alignment is often viewed as an offline and one-time task, continuously evolving ontologies and applications involving real-time ontology alignment such as semantic search and Web service composition stress the importance of computational complexity considerations [10].* Consequently, we position the possibly improved performance gains from using WN in the context of the increased computational time that the enhanced alignment entails. We select a recognized ontology alignment algorithm based on iterative expectation-maximization, which produces the most likely match between two given ontologies [3]. This algorithm uses both the structure of the ontologies and their lexical similarity in arriving at the match. We perform this experiment comprehensively using ontology pairs that appear in the real-world ontologies track of the Ontology Alignment Evaluation Initiative (OAEI) 2009 edition [11]. For our analysis, we think that the real-world ontology pairs are most appropriate due to the nature of our study.

We uncover some surprising trends while comparing the performance of ontology alignment enhanced with WN and that of alignment that uses syntactic matching only. While, in many cases, the WN-enhanced alignment expectedly achieved a better recall and F-measure, it did so while taking significantly more time and aligning without it achieved nearly identical performance in less time. We also report on several pairs where the WN-enhanced alignment did not improve on the performance of the original alignment algorithm. Consequently, we investigate characteristics of the ontology pair that would likely facilitate improved performance when a lexical database such as WN is used during the alignment, and particularly those which would hinder its performance. We think that many of the outcomes of this analysis are novel and useful in evaluating the use of computationally intensive add-ons such as WN.

This study has insights for both ontology alignment researchers and users, and provides useful guidance on utilizing lexical knowledge sources for ontology alignment. Its results provide clear evidence against commonly held beliefs that, (a) the use of WN in ontology alignment *always* improves the recall of the alignment; and (b) any improvement in the recall supersedes the loss in precision that WN may bring, and this is notwithstanding the excessive execution time due to using WN. The contributions of this novel study in the context of alignment are two-fold: First, it shows that the utility of WN in aligning ontologies is not always clear, and the use of WN not always advisable. This is demonstrated by comparing the performance of ontology alignment with WN and that of alignment without WN. For example, we show that multiple benchmark ontology pairs do not exhibit improvements in recall when WN is used despite the larger execution time. More importantly, several benchmark ontology pairs do not show a marked improvement in F-measure when WN is utilized to help the alignment process. Second, it recommends a set of "rules of thumb" for ontology alignment users in order to decide whether WN would be worthwhile for a given ontology pair. For example, we discover that ontologies with deep hierarchies take far more time when aligned with WN than ontologies with shallow hierarchies.

II. Background: Ontology Alignment Algorithm

Multiple algorithms exist for aligning ontologies, some of which are tailored to specific domains. For our study, we select a general-purpose ontology alignment algorithm that formulates the problem of inferring a match between two ontologies as a maximum likelihood problem, and solves it using the technique of expectation-maximization (EM). This algorithm is available in an ontology alignment tool called Optima [3]. Our choice of the algorithm is driven by its competitive performance and the accessibility of Optima, which is available as an intuitive API. Many other ontology alignment tools were either inaccessible or we found no evidence in their documentation that their use of WN could be smoothly switched off, which is required in this study.

Optima adopts directed graphs as its model for ontology schemas and employs a generalized version of EM to arrive at a map between the nodes of the graphs. This is appropriate because contemporary ontology description languages such as RDFS [12] and OWL [13] allow ontology schemas to be modeled as directed labeled graphs. Optima exploits the structural, lexical and instance similarity between the graphs, and differs from the previous approaches in the way it utilizes them to arrive at, a possibly inexact, match. Inexact matching is the process of finding a best possible match between the two graphs when exact matching is not possible or is computationally difficult.

The iterative alignment algorithm requires a seed map. This is an initial list of mappings between concepts often provided to iterative algorithms. While the seed map could be generated manually, Optima additionally utilizes a simple technique of mapping leaf nodes (if they exist) across the ontologies whose labels are syntactically similar. We ensure that the seed map does not exceed 10% of the nodes in the smallest ontology. Candidate alignments are generated using simple but intuitive heuristics. For example, given each previously mapped node pair, their parents are considered for a match. Additionally, their sibling nodes could be considered. Analogous to the seed map, node pairs among the parents that are sufficiently similar are matched. Different potential alignments are generated based on how many parent nodes are matched and whether siblings are matched as well. These candidate alignments are considered during each iteration of Optima. More details about Optima are available in [3].

III. Integrating WordNet

Similarity measures may be broadly categorized into syntactic and semantic. Syntactic similarity between concepts is entirely based on the string similarity between the concepts' names, labels and other associated text. Semantic similarity measures attempt to utilize the meaning behind the concept names to ascertain the similarity of the concepts. A popular way of doing this is to exploit lexical databases such as WN, which provide words related in meaning.

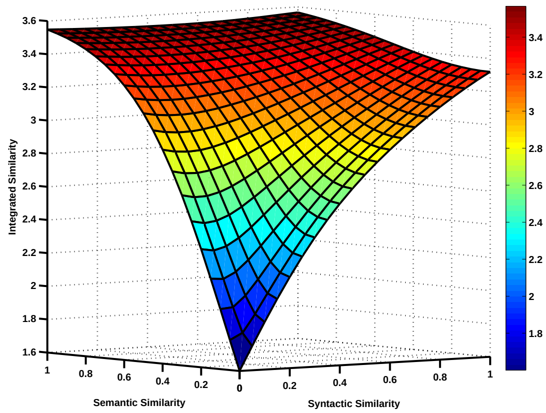
Optima utilizes the well-known Smith-Waterman [14] technique for ascertaining the syntactic similarity between concept and relationship names. We enhance the syntactic similarity to include knowledge from WN [6] as a representative lexical database, popularly used by many ontology alignment tools. In a comparison of different ways of using WN to match concept names, Yatskevich and Giunchiglia [7] demonstrate that gloss-based similarity measuring algorithms (matchers) showed the best matching performance. These matchers compute the cosine similarity between the glosses (definitions) provided by WN for the given words. Consequently, we integrate these matchers with the syntactic matching in Optima. However, these matchers do not utilize the structure of WN – *synsets* and how they relate to each other – and associated statistical knowledge. Hence, we also include another popular and competitive method [15], which uses WN's structure. As we seek to evaluate the incremental utility of WordNet, we augment the existing syntactic similarity in Optima with these WN-based similarity measures.

A. Adding WordNet-based Similarity

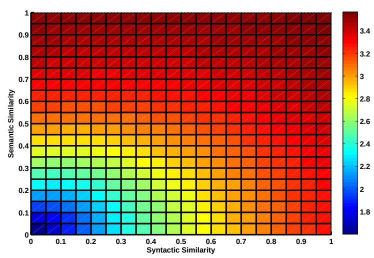
A known limitation of Lin's method [15] is its poor performance when the concept labels are word phrases instead of single words. In this case, we evaluate the WN-based similarity using the gloss-based matcher that accumulates the glosses of each word in the phrase. Consequently, we use Lin's approach if both labels are single words, otherwise the gloss-based matcher is utilized. We denote this way of utilizing WN using *Sem*.

Lin proposes the use of information content in computing the semantic similarity between labels using WN:

$$Lin(x_a, y_\alpha) = \frac{2 \times IC(lcs(x_a, y_\alpha))}{IC(x_a) + IC(y_\alpha)} \quad (1)$$



(a)



(b)

Figure 1. Our integrated similarity measure as a function of the WN-based semantic similarity (*Sem*) and Smith-Waterman based syntactic similarity (*Syn*). Notice that the value is lower if semantic similarity is low but syntactic is high compared to vice versa.

Here, the information content (IC) is computed by looking up the frequency count of its argument word in standard corpora [16]. The term, $lcs(x_a, y_\alpha)$, is the least common subsumer of the two words, x_a and y_α, within the WN hierarchy. Lin is guaranteed to be between 0 and 1.

Let x_a, y_α be the two concepts for which the similarity to be measured and the number of words in each concepts be w_a and w_α respectively then the time complexity of the Lin similarity is $\mathcal{O}(w_a.w_\alpha.s_a.s_\alpha.h)$ [17]. Here the number of senses in WN for x_a, y_α are s_a and s_α with h being the maximum depth of both the concepts in WN hierarchy. The time complexity of the gloss based similarity would be then, $\mathcal{O}(w_a.w_\alpha.s_a.s_\alpha.g_a.g_\alpha)$, where g_a and g_α are the maximum number of words in any single gloss in WN for concepts x_a and y_α respectively. Note that the number of words in a concept and the depth of the words in the WN hierarchy determines the complexity of computing its similarity using WN.

There is no standard way of integrating WN-based similarity with syntactic measures. We define a normalized 3D function that maps a given pair of semantic and syntactic similarity to the integrated value. In order to generate this function, we observe that labels that are syntactically similar (such as *cat* and *bat*) may have different meanings.

Table I

ONTOLOGIES FROM OAEI 2009 PARTICIPATING IN OUR EVALUATION AND THE NUMBER OF NAMED CLASSES AND PROPERTIES IN EACH. NOTICE THAT OUR EVALUATION INCLUDES REASONABLY LARGE ONTOLOGIES AS WELL.

Ontology	**Named Classes**	**Properties**
101	37	70
205	36	69
301	16	40
302	14	30
303	57	72
304	41	49
ekaw	74	33
sigkdd	50	28
iasted	150	41
cmt	30	59
edas	104	50
confOf	39	36
conference	60	64

Because we wish to meaningfully map entities, semantic similarity takes precedence over syntactic. Consequently, high syntactic but low semantic similarity results in a lower integrated similarity value in comparison to low syntactic but high semantic similarity. We model such an integrated similarity measure as shown in Fig. 1 and give the function in Eq. 2. Our integrated similarity function is similar to a 3D sigmoid restricted to the quadrant where the semantic and syntactic similarities range from 0 to 1. One difference from the exact sigmoid is due to the specific property it must have because semantic similarity takes precedence over syntactic.

$$Int(x_a, y_\alpha) = \gamma \frac{1}{1 + e^{t \cdot r - c(Sem)}} \tag{2}$$

Here, γ is a normalization constant; $r = \sqrt{Syn^2 + Sem^2}$, which produces the 3D sigmoid about the origin; t is a scaling factor and $c(Sem)$ is a function of the semantic similarity as shown below: $c(Sem) = \frac{2}{1 + e^{t' \cdot Sem(x_a, y_\alpha) - c'}}$ where t' is the scaling factor and c' is the translation factor, if needed. The specific function in Fig. 1 is obtained when $t = 4$, $t' = 3.5$, and $c' = 2$.

IV. EXPERIMENTS

As we mentioned previously, alignment algorithms have used lexical databases such as WN based on the potential improvement in the alignment that it could generate. Furthermore, past studies of using WN do not take into account the increased computational load that utilizing WN entails. We analyze the implications of using WN on the alignment performance in the context of Optima.

A. Methodology

We utilized execution time as an indicator of the computational load. In order to incorporate execution time within our experimentation, we measure the maximum *recall* and *F-measure* that Optima attains on a pair of ontologies given varying execution time. We evaluated recall and F-measure because integrating WN typically results in improved recall but reduced precision, which would be collectively reflected in the F-measure. We measured recall and F-measure as follows:

$$\text{Recall} = \frac{\text{Number of true mappings discovered by Optima}}{\text{Total number of true mappings between the ontologies}}$$

$$\text{Precision} = \frac{\text{Number of true mappings discovered by Optima}}{\text{Total number of mappings discovered by Optima}}$$

$$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}$$

The alignment performance was measured with the integrated similarity measure and independently using just the syntactic similarity between node labels, in order to evaluate the utility of WN. We used OAEI in its recent version, 2009, as the testbed for benchmarking. Within the benchmark, we mostly focus on the track that involves real-world ontologies for which the reference (true) alignment was provided by OAEI. These ontologies are not created or altered for purposes related to the benchmark and were obtained by OAEI from the Web. This includes all ontology pairs in the 300 range which relate to *bibliography*, and expressive ontologies in the *conference* track all of which structure knowledge related to conference organization. Because we wish to evaluate the utility of WN in practical use, we focused on real-world ontologies. However, we selected one pair of ontologies specifically tailored by the benchmark that contained synonyms of node labels. We list the ontologies participating in our evaluation in Table I and provide an indication of their sizes.

We ran each execution – with WN and without – until there was no improvement in the performance. During the execution, we recorded the recall and F-measure every time it changed along with the time consumed till then. Because of the iterative nature of Optima, the alignment performance usually improves as more time is allocated until the EM converges to a maxima. We note that we seed both executions with the same initial alignment to facilitate comparison.

B. Results and Analysis

While we ran our evaluations on 23 pairs of ontologies, in this section we focus on a set of 6 pairs, which are representative of the different trends that we obtained. We show our evaluations on some of the remaining pairs in the Appendix. Because of the large number of pairs that we evaluated on (23 in all), we ran the tests on three different computing platforms. Two of these were Red Hat machines with Intel Xeon Core 2, processor speed of about 3 GHz with 4GB of memory, while the third was a Windows Vista machine with Intel Xeon Core 2, 2.4 GHz processor and 4GB of memory.

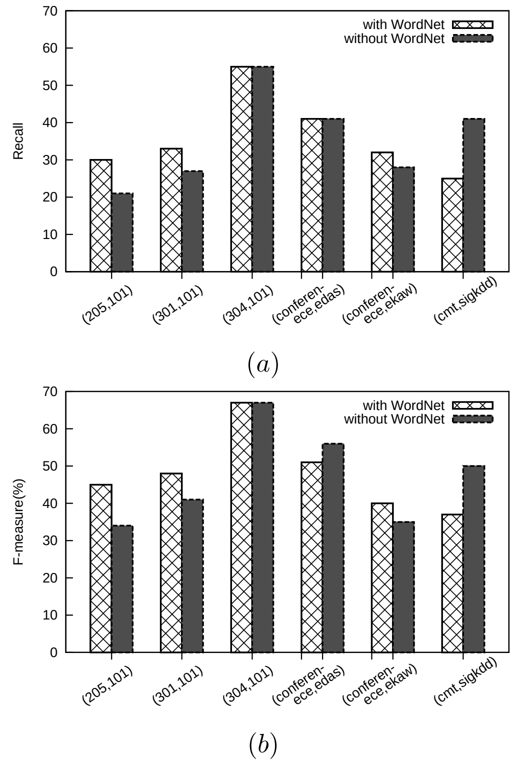


Figure 2. (a) Final recall and (b) final F-measure generated by Optima on 6 representative ontology pairs, with the integrated similarity measure and with just the syntactic similarity between entity labels.

We show a summary of the final recall and F-measure that was obtained on the 6 pairs with WN integrated and with just the syntactic similarity measure, in Figs. 2 (a, b). Our focus is on the change in these measures, and not their overall values which could be poor for some ontology pairs. As we may expect, for many of the ontology pairs, the final recall with WN integrated is higher than the recall with just the syntactic similarity. For example, while aligning the ontology pair *(101, 301)*, the alignment process with WN matches the concept *Monograph* against the concept *Book*, which is not possible with using just the syntactic similarity. The difference in performance is statistically significant with p-value of 0.057 as measured using a paired Student's t-test. On the other hand, integrating WN decreased the recall for a single pair, *(cmt,sigkdd)*. However, the improvement in F-measure due to WN reduces to the extent where it loses significance (p-value=0.184).

In Fig. 3, we detail the performances w.r.t. execution time. Each data point is the maximum recall or F-measure, as appropriate, that could be obtained given the execution time. Notice that Figs. $3(a, b, e)$ all show an improved recall with WN integrated. In particular, ontology 205 in the pair (205, 101) is altered by OAEI to include synonyms of labels

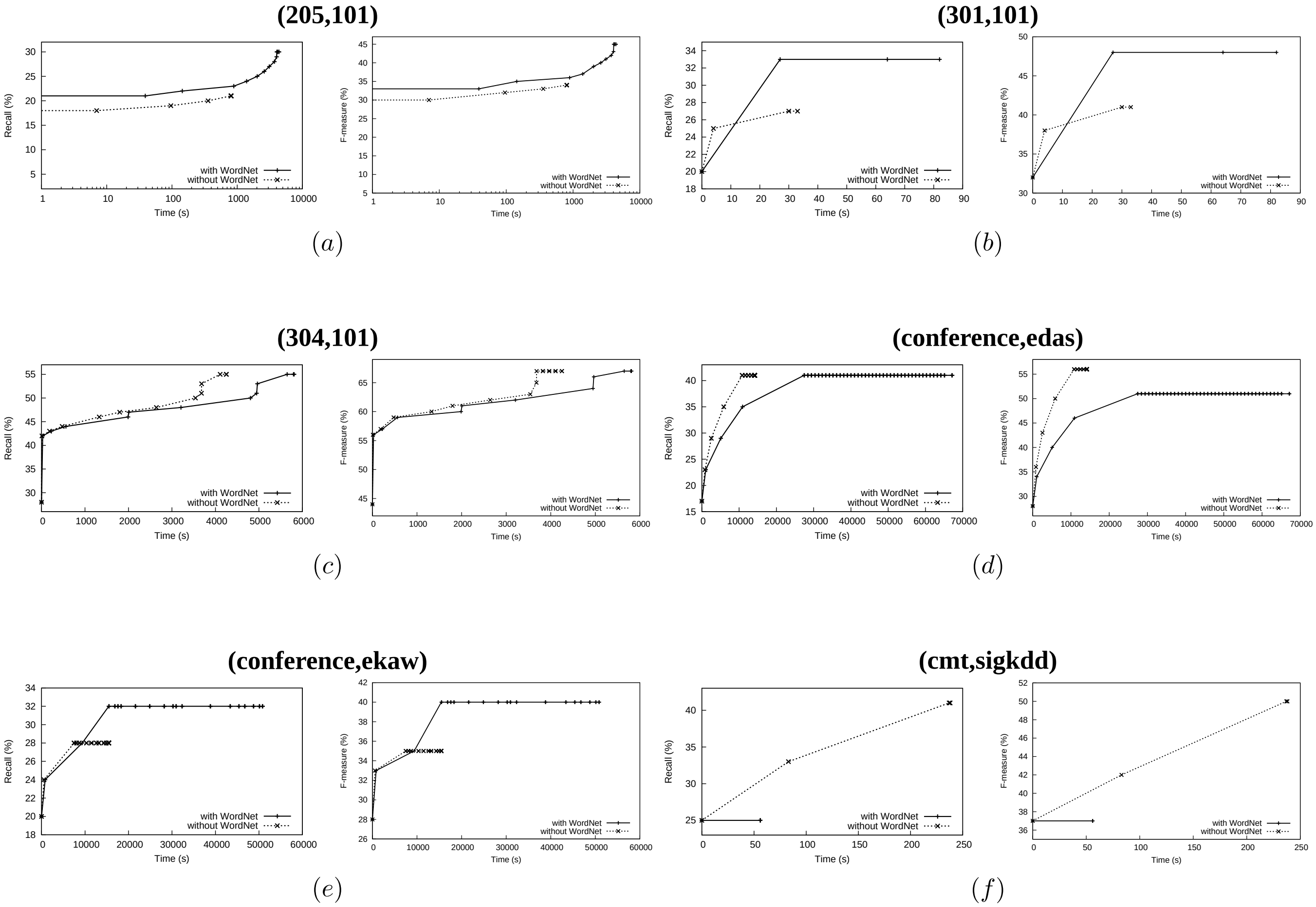


Figure 3. $(a)-(f)$: Recall (left) and F-measure (right) for 6 of the 23 ontology pairs that we used in our evaluations. We show the evaluations when the alignment algorithm utilized an integrated similarity involving WN and just the string-based similarity without WN. Notice the different trends in our evaluations. Ontologies related to *conference* consume more time because they are larger.

in 101, as its entity labels. For example, *title* is altered to *heading*. In some cases, the WN-based integrated similarity leads to better recall eventually. However, the improvement is obtained after spending significantly more time on the alignment process; in some cases approximately an order of magnitude more time was consumed to achieve a significant increase, as in Fig. $3(a)$. The additional time is spent on initializing WN and querying the database. Further, in two of these, aligning without WN results in better recall for an initial short time span (Fig. $3(b,e)$), before the performance with WN exceeds it.

On the other hand, some ontology pairs did not exhibit an improved recall with WN (see Figs. $3(c,d,f)$). Surprisingly, conference ontology pair *(cmt, sigkdd)* results in worse recall with WN integrated (Fig. $3(f)$). This is because *(cmt, sigkdd)* pair has several concepts with compound words or phrases as labels. As one example, *Meta-Review* appears in *cmt* ontology and *Registration_Non–Member* appears in *sigkdd* ontology. Tokenizing these correctly and locating individual glosses in WN is often challenging[1], resulting in low semantic and therefore low integrated similarity. However, the string-based similarity resulted in better label matching.

Our F-measure evaluations of the alignments tell another story. We predominantly found that the improvement in F-measure due to WN was smaller in comparison to the improvement in recall. Thus, the use of WN often leads to reduced precision than if we did not use it. Due to its consideration of synonyms and other lexical senses, semantic similarity is often high for multiple concepts across the

[1] The concept *Meta-Review* should be tokenized into two words *(Meta, Review)* while *Registration_Non–Member* needs to be tokenized into two words *(Registration, NonMember)* but should not be tokenized into three words *(Registration, Non, Member)*. The hyphen (–) is a delimiter in the former concept but should be just ignored in the later concept. This tokenization is demanded by WN matchers since *MetaReview* does not exist in WN but the word *NonMember* exists in WN.

Table II
THE DIFFERENT ONTOLOGY PAIRS COULD BE GROUPED INTO 4 TRENDS OF ALIGNMENT PERFORMANCE BASED ON THE RECALL AND F-MEASURE EVALUATIONS.

	Max. recall +WN $>$ Max. recall -WN	**Max. recall +WN $=$ Max. recall -WN**	**Max. recall +WN $<$ Max. recall -WN**	**Count**
Max. F-measure +WN $>$ Max. F-Measure -WN	(205, 101); (301, 101); (confOf, ekaw); (edas, iasted); (cmt, conference); (conference, iasted); (conference, ekaw)	none	none	7
Max. F-measure +WN $=$ Max. F-Measure -WN	none	(304,101); (cmt, confOf); (ekaw, iasted); (ekaw, sigkdd); (iasted, sigkdd); (conference, sigkdd)	none	6
Max. F-measure +WN $<$ Max. F-Measure -WN	none	(302, 101); (303, 101); (confOf, edas); (edas, sigkdd); (edas, ekaw); (cmt, edas); (cmt, ekaw); (conference, confOf); (conference, edas)	(cmt, sigkdd)	10
Count	7	15	1	23

two ontologies. However, not all of these possible matches appear in the true alignment. For example, while the final recall in Fig. $3(d)$ does not change when WN is utilized, the final F-measure drops to below what we could get when just the syntactic similarity is used in Optima for the alignment. The mapping between concepts *Conference_part* and *ConferenceEvent* in ontologies *Conference* and *edas*, respectively, is one such example that is found by Optima with WN but is incorrect and therefore leads to lower precision. Furthermore, the increased execution time due to WN for achieving an F-measure is significant (p-value = 0.013).

Overall, we saw general trends where, (i) the final recall and F-measure due to WN improved considerably although the lower values of recall and F-measure were achieved without the use of WN in much less time; (ii) alignment with WN exhibited similar or better recall but poorer F-measure due to reduced precision; and (iii) integrating WN degraded the alignment performance, although this was rare. We tabulate the alignment performance on all the 23 different pairs based on the trends, in Table II. Interestingly, 15 of the 23 pairs that we used did not exhibit an increase in recall due to the additional use of WN, and 9 of these showed a decrease in overall F-measure.

C. Discussion

Our results in the previous section demonstrate that integrating a lexical database such as WN may not always be worthwhile especially if the execution time is a concern as well. In particular, the performance in terms of recall or F-measure did not improve for 15 of the 23 ontology pairs when an integrated similarity measure involving WN was utilized. However, the execution time increased considerably. Clearly, the utility of WN for these ontology pairs is negligible. We investigated these pairs in greater detail to ascertain the differential properties that could lead to minimal performance improvement on using WN. These would allow us to make an informed decision on whether WN would be worthwhile for a given ontology pair.

• Interestingly, ontologies that have a deep hierarchy ("tall" ontology) may consume an excessive amount of time when aligned using WN. This is because such ontologies tend to have several specialized classes, and identifying the least common subsumer in WN required by algorithms such as Lin [15] requires traversing a large portion of the WN hierarchy (see section III-A). An example of this is the ontology pair, *(conference, edas)*, in which the ontology *edas* is a tall ontology.
• Furthermore, if such ontologies need to be aligned with those that have a shallow hierarchy ("short" ontology), WN will likely suggest several matches between the specific concepts[2] of the tall ontology and more general concepts of the short ontology, thereby leading to reduced precision.
• We may search WN using single words only. Consequently, compound words or phrases appearing as entity labels in an ontology need to be appropriately tokenized and a single representative word or WN-based similarity measure must be obtained. This is further complicated if the phrases are not formatted in a uniform manner making tokenization challenging. An example of this is the ontology pair, *(cmt, sigkdd)*, which leads to poor performance with WN due to the difficulty in improving over the seed map (see Fig. $3(f)$).

V. RELATED WORK

Giunchiglia et al. [7] studied the alignment performance in terms of recall and precision of a set of semantic matchers between concept labels, which use WN as a source of background knowledge. While this study reported significant improvement in alignment quality on small OAEI 2006 ontologies due to WN, it did not evaluate the increased execution time. Aleksovski and Harmelen [18] investigated the impact of using a background knowledge ontology in ontology matching. Again, this effort did not explore the computational trade off in the form of increased execution

[2]Specific concepts (e.g., *Presenter* in "tall" *edas* ontology) appear at the lower part of the WN hierarchy tree compared to general concepts (e.g., *Person* in "short" *confOf* ontology) which stay closer to the root of the WN tree.

time required for leveraging an external knowledge store for ontology alignment. Similarly, Lin and Sandkuhl [9] surveyed different ways of utilizing WN in aligning ontology pairs. These approaches have predominantly focused on comparing different ways of integrating WN.

While the previous investigations have compared various ways of exploiting WN for ontology alignment, a comprehensive study of the utility of WN especially keeping in mind the increased computational load has not been performed. This is a pertinent question in the context of emerging evidence that lexical databases may not always improve results [8]. Indeed, OAEI questions the utility of background knowledge due to the relative poor performances of ontology alignment algorithms that utilize background knowledge in the anatomy track of its 2009 campaign [11]. Because automatically aligning ontologies is computationally intensive, which is exacerbated as the ontologies become larger and real-time needs for ontology alignment emerge, we think that execution time is a critical component.

VI. Conclusion

While using WN in addition to syntactic string-based similarity measures does improve the quality of the alignment in many cases, it does so after consuming significantly more time. Furthermore, the precision of the alignment typically reduces leading to much reduced improvement in F-measure. We also reported on many ontology pairs where WN did not improve on the final recall or F-measure, but consumed more time. Clearly, the utility of WN is questionable in these cases. We analyzed the ontologies for which using WN did not improve the performance, and provided a few rules of thumb related to characteristics of ontologies for which WN should be utilized cautiously. Based on our results and analysis, our recommendation to the ontology alignment research community is not to discourage the use of WN but allow WN usage within the alignment process to be *optional*, and its use be recommended after analyzing the characteristics of the ontologies.

Of course, our study could be enhanced by evaluating the utility of WN in the context of multiple alignment algorithms and more ways of using WN. However, our focus on the relative change in performance due to WN reduces the effect of the choice of the underlying algorithm on the results, and we sought to select multiple competitive WN-based matchers with prior support. As such, we think that our results reflect the general pattern. Additionally, we used 23 independently developed real-world ontology pairs from two distinct domains (bibliography and conferences), which we think is a relatively versatile dataset from which to generalize our conclusions. Furthermore, emerging applications of ontology alignment such as in semantic Web services and search bring new emphasis on alignment execution time.

Acknowledgment

This research is supported in part by grant number R01HL087795 from the National Heart, Lung, And Blood Institute. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Heart, Lung, And Blood Institute or the National Institutes of Health.

References

[1] W. Hu, N. Jian, Y. Qu, and Y. Wang, “Gmo: A graph matching for ontologies,” in *Workshop on Integrating Ontologies, K-Cap*, 2005, pp. 43–50.

[2] Y. Li, J. Li, and J. Tang, “Rimom: Ontology alignment with strategy selection,” in *ISWC & ASWC*, 2007, pp. 51–52.

[3] P. Doshi, R. Kolli, and C. Thomas, “Inexact matching of ontology graphs using expectation-maximization,” *Web Semantics*, vol. 7, no. 2, pp. 90–106, 2009.

[4] Y. R. Jean-Mary, E. P. Shironoshita, , and M. R. Kabuka, “Ontology matching with semantic verification,” *Web Semantics*, vol. 7, no. 3, pp. 235–251, 2009.

[5] P. Wang and B. Xu, “Lily: Ontology alignment results for oaei 2008,” in *OM Workshop ISWC*, 2009.

[6] G. A. Miller, “Wordnet: A lexical database for english,” in *CACM*, 1995, pp. 39–41.

[7] M. Yatskevich and F. Giunchiglia, “Element level semantic matching using wordnet,” University of Trento, Tech. Rep., 2007.

[8] R. Mandala, T. Tokunaga, and H. Tanaka, “Improving information retrieval system performance by combining different text-mining techniques,” in *Intelligent Data Analysis*, vol. 4, no. 6, 2000, pp. 489–511.

[9] F. Lin and K. Sandkuhl, “A survey of exploiting wordnet in ontology matching,” in *Artificial Intelligence in Theory and Practice II*, M. Bramer, Ed., 2008, vol. 276, pp. 341–350.

[10] T. Hughes and B. Ashpole, “The semantics of ontology alignment,” in *I3CON*, 2004.

[11] (2009) Ontology alignment evaluation initiative 2009. [Online]. Available: http://oaei.ontologymatching.org/2009/

[12] B. McBride and R. Guha, “Rdf vocabulary description language 1.0: Rdf schema,” W3C, Tech. Rep., 2004.

[13] D. McGuinness and F. Harmelen, “Owl web ontology language overview,” W3C, Tech. Rep., 2004.

[14] T. F. Smith and M. S. Waterman, “Identification of common molecular subsequences,” in *JMB*, vol. 147, no. 1, 1981, pp. 195–197.

[15] D. Lin, “An information-theoretic definition of similarity,” in *ICML*, 1998, pp. 296–304.

[16] M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz, “Building a large annotated corpus of english: The penn treebank,” in *Comp. Linguistics*, vol. 19, no. 2, 1993, pp. 313–330.

[17] A. Islam and D. Inkpen, “Semantic text similarity using corpus-based word similarity and string similarity,” *ACM Trans. Knowl. Discov. Data*, vol. 2, pp. 10:1–10:25, 2008.

[18] Z. Aleksovski, W. ten Kate, and F. van Harmelen, “Ontology matching using comprehensive ontology as background knowledge,” in *OM Workshop,ISWC*, 2006, pp. 13–24.

APPENDIX
REST OF THE RESULTS

We evaluated the recall and F-measure of the alignment generated by Optima when WordNet is integrated and that of the alignment when just the syntactic similarity is used. While we showed the results and discussed the trends for 6 representative ontology pairs out of 23 in Section IV-B, the results for the rest of the ontology pairs are given below for completeness.

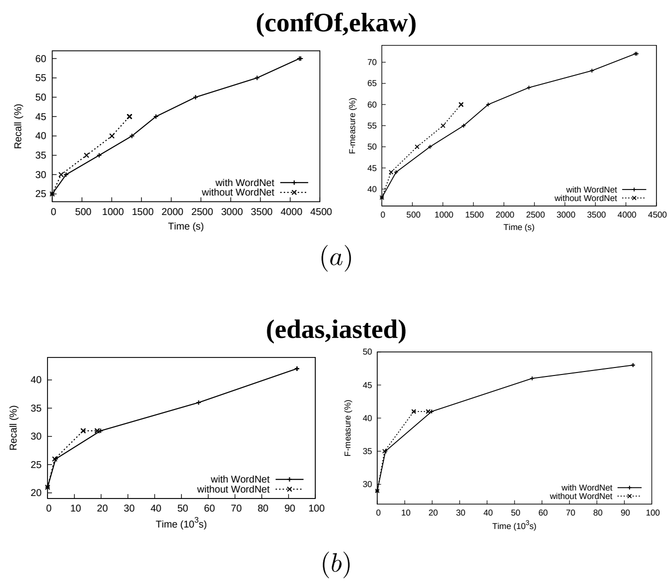


Figure 4. Recall and F-measure for 2 ontology pairs of the same trend where the final recall and F-measure with WN integrated is higher than the recall and F-measure with just syntactic similarity.

We categorized the ontology pairs based on the trends that their recall and F-measure exhibited. In Fig. 4, we show another 2 out of 7 of those pairs for which the final recall and F-measure due to WN improved considerably although, in some cases, the intermediate values of recall and F-measure were achieved by Optima without WN in less time.

Next, we show pairs for which the alignment with WN showed similar recall and F-measure as achieved by aligning with just string similarity. Six ontology pairs exhibit this trend and we show 2 of them in Fig. 5. Notice the increased execution time due to WN for similar recall and F-measure values.

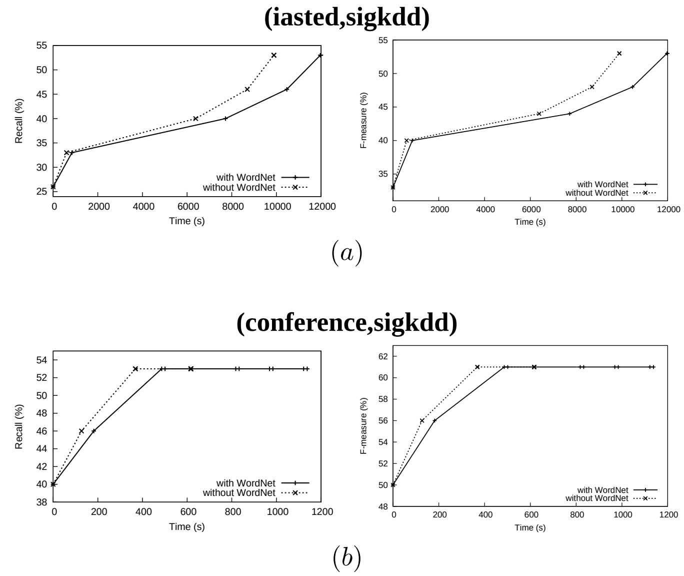


Figure 5. Recall and F-measure for 2 ontology pairs of the same trend where the final recall and F-measure with WN integrated did not improve on the recall and F-measure without WN.

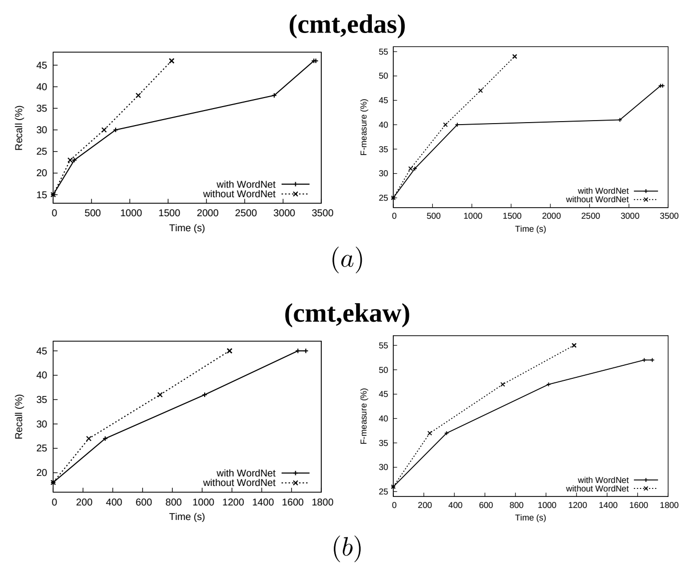


Figure 6. Both the ontology pairs shown here exhibit a final recall with WN that is same as the recall without it. However, the F-measure with WN is less than the F-measure without WN.

Finally, 10 ontology pairs resulted in recall with WN that was similar to recall with just the syntactic string similarity, but poorer F-measure while aligning with WN due to reduced precision. When the additional execution time is taken into consideration, the utility of WN is questionable in these cases. We show 2 of these pairs in Fig. 6.