Text Based Similarity Metrics and Deltas for Semantic Web Graphs

Krishnamurthy Koduvayur Viswanathan and Tim Finin
University of Maryland, Baltimore County
Baltimore, MD 21250 USA
{krishna3, finin}@umbc.edu

Abstract. Recognizing that two Semantic Web documents or graphs are similar and characterizing their differences is useful in many tasks, including retrieval, updating, version control and knowledge base editing. We describe several text-based similarity metrics that characterize the relation between Semantic Web graphs and evaluate these metrics for three specific cases of similarity: similarity in classes and properties, similarity disregarding differences in base-URIs, and versioning relationship. We apply these techniques for a specific use case – generating a delta between versions of a Semantic Web graph. We have evaluated our system on several tasks using a collection of graphs from the archive of the Swoogle Semantic Web search engine.

Keywords: Semantic Web graphs, similarity metrics, delta

1 Introduction and Motivation

Semantic Web search engines can benefit from recognizing nearly duplicate documents [2] for many of the same reasons that text search engines do. Comparing Semantic Web documents or graphs, however, is more complicated. In natural language text local word order is important to meaning while the order of triples in a Semantic Web document (SWD) is not important. As a result, equivalent Semantic Web documents may have completely different statement ordering. It is also possible to have two different SWDs, which become identical after performing inference. The presence of "blank nodes" in SWDs further complicates their comparison.

We explore three different ways in which a pair of Semantic Web graphs can be similar to each other: similarity in classes and properties used while differing only in literal content, difference only in base-URI, and implicit versioning relationship [4, 3]. We define text-based similarity metrics to characterize the relation between them. As a part of this, we identify whether they may be different versions of the same graph. Furthermore, if we determine a versioning relationship between a candidate pair, then we generate a *delta*, i.e., a detailed description of their differences at the triple level delta between them.

These methods enable a Semantic Web search engine to organize its query results into groups of documents that are similar with respect to the different metrics and also generate deltas between documents judged to have a versioning relationship.

2 Objective and Approach

Our objective is identify pairs of documents that are similar to each other in a collection of SWDs. We characterize SWD similarity along three dimensions: a similarity in classes and properties used while differing only in literal content, difference only in base-URI, and versioning relationship. For pairs of SWDs exhibiting a versioning relationship we compute a triple-level delta for them.

Our input corpus is in the form of a set of RDF documents. Our approach involves the following steps:

Convert to canonical representation: We convert all the documents into a uniform n-triple serialization. Since equivalent semantic web graphs may have different n-triple serializations, we apply the algorithm described in [4] which assigns consistent IDs to blank nodes and lexicographically order the triples.

Generate reduced forms: In order to compute the similarity measures, the canonical representations are decomposed into the following four reduced forms. Each is a document with:

1. only the literals from the canonicalized n-triples file
2. literals replaced by the empty string[1]
3. the base-URI of every node replaced by the empty string
4. literals and the base-URI of every node are replaced by the empty string

Thus, each Semantic Web graph has a canonical representation, and four reduced forms i.e. five forms in all.

Compute similarity measures: Given the text-based reduced forms, we can use the the following similarity measures from the information retrieval field.

1. **Jaccard similarity and containment**: We construct sets of character 4-grams from the input documents which are used to compute the measures. A high value for both Jaccard and containment metrics indicates a strong possibility of a versioning or equivalence relation between two.
2. **Cosine Similarity between semantic features**: Each SWD is represented as a vector of terms. The non-blank, non-literal nodes from each SWD are extracted and their term-frequency in the SWD is used as the feature weight.
3. **Charikar's Simhash**: We compute the Hamming distance between the simhashes of the documents being compared. Simhash [1] is a locality sensitive hashing technique where the fingerprints of similar documents differ in a small number of bit positions.

Pairwise computation of metrics: Given an input of Semantic Web documents, we need to find all pairs that are similar to each other. The total number of metrics computed for each pair of SWDs is 17: two kinds of cosine similarity, and three other metrics for each reduced form pair. The process is as follows:

1. Compute the two cosine similarity values between the canonical representations (already generated) of both the SWDs.

[1] Another viable approach is to replace each literal string by its XSD data type

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.973	0.000	1.000	0.973	0.986	1.000	yes
	1.000	0.027	0.973	1.000	0.987	0.996	no
Weighted Avg.	0.986	0.014	0.987	0.986	0.986	0.998	

Table 1: Accuracy by class (classes and properties used), using Naive Bayes

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	1.000	0.040	0.962	1.000	0.980	0.979	yes
	0.960	0.000	1.000	0.960	0.980	0.990	no
Weighted Avg.	0.980	0.020	0.981	0.980	0.980	0.985	

Table 2: Accuracy by class (pairs different only in base URI), using Naive Bayes

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.864	0.045	0.950	0.864	0.905	0.909	yes
	0.955	0.136	0.875	0.955	0.913	0.909	no
Weighted Avg.	0.909	0.091	0.913	0.909	0.909	0.909	

Table 3: Accuracy by class (versioning relationship), using SVM with linear kernel

.

2. If the cosine similarity values are below a pre-determined threshold, then eliminate this pair from further consideration, else add this pair to a list of candidate pairs. The threshold for this step was determined empirically by performing pilot experiments, and was set at 0.7.
3. For all candidate pairs, compute the remaining three pairwise similarity metrics for each reduced form.

Thus the cosine similarity metric is used as an initial filter to reduce the remaining computations. It is to be noted that this pairwise comparison approach entails a quadratic number of comparisons.

3 Classification and Delta

We trained three classifiers (one for each kind of similarity defined) with a dataset collected from the Swoogle, annotated with human judgements of the three kinds of similarity. Pairwise similarity measures are computed for each candidate pair in the labeled dataset and three different feature vectors (one for each classifier) are constructed for each candidate pair, using appropriate attributes. The attributes used are the similarity measures that have been computed. These classifiers are then used to detect the three forms of similarity that we have defined. For a list of attributes used for each classifier, see [4].

Once it is determined that two SWDs have a versioning relationship between them, we compute the set of statements that describe the change between successive versions of the SWD. For details on the specific deltas that we compute between two versions, see [4].

4 Evaluation and Conclusion

Our system is based on several informal types of similarity that we have observed among documents in Swoogle's repository. In addition, there exists no standard labeled dataset of similar Semantic Web documents that we could use for the purpose of evaluating our system. Hence we constructed a collection of Semantic Web documents from Swoogles Semantic Web archive and cache services. Swoogle periodically crawls the Semantic Web and maintains several snapshots for each indexed SWD. We added such versions to our data-set and labeled them as having a versioning relationship. We constructed a labeled dataset of 402 SWDs (over 161,000 candidate pairs) from Swoogle's Semantic Web archive and cache services[2]. The results of the classification process are as shown in Tables 1, 2, and 3.

The results shown in Tables 1, and 2 were obtained by performing a ten-fold stratified cross validation using the labeled dataset. Table 3 shows results of evaluation on a test set that was constructed in the same way as the training data-set. The high true positive rate for determining version relationships between SW graph pairs allows us to develop applications like generation of deltas. For details on attributes used for each classifier, refer to [4]

We developed techniques to recognize when two Semantic Web graphs are similar and characterize their difference in three ways. When the system detects a versioning relationship between a pair, we also generates a delta in terms of triples to be added or deleted. One of the future directions is to increase scalability. We intend to implement a principled filtering mechanism to reduce the number of pairwise comparisons. We might use the Billion Triples Challenge dataset for experiments[3]. We also plan to develop a global similarity and ranking scheme where, given a sample, we would be able to find and rank the most similar matches. This would require representation of the similarity measures by more complex schemes, e.g., lattices.

References

1. M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC '02: Proc. of the thiry-fourth annual ACM symposium on Theory of computing*, pages 380–388, New York, NY, USA, 2002. ACM.
2. G. S. Manku, A. Jain, and A. D. Sarma. Detecting near-duplicates for web crawling. In C. L. Williamson, M. E. Zurko, P. F. Patel-Schneider, and P. J. Shenoy, editors, *WWW*, pages 141–150. ACM, 2007.
3. K. Viswanathan. Text Based Similarity Metrics and Delta for Semantic Web Graphs. Master's thesis, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore MD 21250, June 2010.
4. K. Viswanathan and T. Finin. Text Based Similarity and Delta for Semantic Web Graphs. Technical report, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore MD 21250, August 2010.

[2] http://swoogle.umbc.edu/index.php?option=com_swoogle_service&service=archive
[3] http://km.aifb.kit.edu/projects/btc-2010/