The Properties of Property Alignment on the Semantic Web

Michelle Cheatham[1]

Computer Science and Engineering
Wright State University
Dayton OH, USA
michelle.cheatham@wright.edu

Catia Pesquita, Daniela Oliveira

Helena B. McCurdy

Abstract

The performance of alignment systems on property matching lags behind that on class and instance matching. This work seeks to understand the reasons for this and consider avenues for improvement. The paper contains an exploration of the performance of current alignment systems on the only commonly accepted alignment benchmark that involves matches between properties. A second benchmark involving properties from DBPedia and YAGO, scaled to be within the capabilities of most existing alignment systems, is also proposed. A basic approach focused on aligning properties is then presented and evaluated using both benchmarks to serve as a baseline against which to compare more complex matchers on the property alignment task. The results show that even a relatively simplistic approach can achieve a significantly higher F-measure than current matchers. Finally, an existing full-featured alignment system is augmented with the basic property matching approach and the difference in performance is assessed.

Keywords: Ontology alignment, Semantic data integration, Semantic web, Ontology alignment benchmark

1. Introduction

Until recently, many ontology alignment systems focused first and foremost on aligning classes, and performance on property alignment lagged behind accordingly. However, as ontology-based applications mature, property matching becomes crucial to support the required reasoning and querying capabilities.

This paper provides quantitative support for the claim that current alignment systems perform significantly worse on properties than on classes and explores the reasons behind this performance gap. These include the differences in property labeling conventions that limit the effectiveness of the string similarity metrics that are traditionally employed in alignment systems on properties compared to classes, as well as the common lack of a rich property hierarchy comparable to that often found for classes, which limits the utility of structural matching approaches. We show that it *is* possible to achieve reasonable performance on property matching if string metrics are chosen to maximize performance on the type of labels given to properties rather than classes, and if domain and range information, which is of course not applicable to classes, is considered. This basic approach provides a useful baseline against which alignment systems can be compared on property matching tasks.

Another limiting factor in the development of strong property alignment techniques is the lack of suitable benchmarks. The OAEI Conference track is currently the only commonly used non-synthetic benchmark that involves any matches between properties, and these are a small percentage of the overall matches within the benchmark. We therefore introduce a new property-centric alignment benchmark based on DBPedia and YAGO and evaluate the performance of existing alignment systems and our baseline property matcher on it.

This paper significantly expands on our work presented in a workshop paper on the same topic [1] by evaluating the accuracy of the confidence values assigned by the baseline approach, strengthening the rigor of the real-world evaluation of the approach, and by incorporating this approach to property matching into a full-featured alignment system and evaluating the impact on performance.

The central contributions of this paper are:

- In-depth analysis of the performance of current alignment systems on properties, including common false positives and false negatives.
- Introduction of a baseline property alignment ap-

[1]Corresponding author

Figure 1: Performance of string metrics on classes versus properties

proach designed for accurate similarity computation between properties, and evaluation of its performance in order to serve as a baseline for full-featured systems with respect to property alignment tasks.

- Proposal of a second real-world property-centric alignment test set that is scaled to be within the capabilities of most existing alignment systems and presentation of results on this task for several systems.

2. Property Matching is Different

Our work in [2] indicates that string metrics are less effective when comparing properties than when comparing classes. Figure 1 shows this difference in performance for eleven string metrics on the Conference track within the OAEI. Others have noted the challenge of aligning properties as well. For example, this is stated without additional detail by Maedche and Staab in [3], while Pernelle et al. note that human experts had a more difficult time agreeing on when properties match than on when classes do [4].

2.1. Related Work

Previous work has considered why property matching appears to have a different nature than aligning other types of entities. Empirical analysis of existing ontologies has shown that different naming conventions are used for different entity types. For instance, object properties generally begin with a verb (e.g. attends, employs) or end with a preposition (e.g. friendOf, componentFor) while datatype properties are usually nouns (e.g. value, id, etc.). Additionally, the names of inverse properties were found to commonly follow one of two patterns: active and passive forms of the same verb (e.g. wrote and writtenBy) or the same noun phrase packed in auxiliary terms (e.g. memberOf and hasMember) [5]. These different naming conventions may be one reason for the generally poor performance of syntactic similarity metrics on properties. Nearly all alignment systems utilize one or more string similarity metrics, so mistakes at this level can impact the overall performance of those systems. Additionally, labels for properties often contain more general (i.e. common) words than those of classes, which can make semantic comparisons less effective.

Another particularly problematic aspect for property matching is the variety of design decisions made when an ontology is created. For instance, some ontologies are class-centric while others are property-centric (e.g. SeasonTicketHolder versus holdsSeasonTicket) [6]. Intuitively we would like to say that if two ontologies had these entities, there should be some sort of mapping between them. Other ontology design decisions that impact property matching are how to handle part-whole relationships and when to reify properties [7]. Additionally, taxonomies of properties are much less common than those of classes [5, 7]. These issues cause problems for structural similarity metrics. There has been some discussion in the literature of handling differences in design philosophy through ontology transformation, in which design patterns are recognized and translated into an analogous form [8]. Ritze and her colleagues used this pattern-centric approach to find complex mappings between classes (and value restrictions) in one ontology and properties in another [9].

In 2002 Melnik and his colleagues developed a strategy called similarity flooding to improve the performance of alignment systems. The general idea is that an initial pass is made through the datasets to establish a set of high precision anchor mappings, such as exact string matches. Then similarity values are propagated to adjacent nodes. If the similarity value of two nodes reaches a threshold, they are considered equivalent. The algorithm iterates until a fixed point is reached [10]. This technique may improve the performance on property alignment by leveraging the increased accuracy of class and instance alignment. An ontology-centric version of the basic similarity flooding technique was first employed in RiMOM and subsequently adopted by many other ontology alignment systems. Rather than propagating similarity values to all neighbors in a graph structure, this approach considered sub-concepts, siblings, and properties for classes and sub-properties, range, and domain for properties [11]. Suchanek et. al. applied this ontology-oriented similarity flooding approach in their PARIS alignment system, which identifies both equivalence and subsumption relations for classes and properties [12]. They found that while class alignments didn't do much to facilitate alignment of properties or instances, there was significant interplay between the latter two. This was particularly true for functional or nearly functional properties, in which any domain value maps to only one range value.

There have been several attempts to modify the standard similarity flooding approach to further improve the performance on property matching. For example, comparison of instance data and datatype property range values can be improved by using different similarity metrics for strings, dates, integers, etc. [13]. Further, in deference to the difficulty of matching properties, it is possible to propagate a fraction of the normal similarity values when adjacent properties are compatible rather than definite matches. This is the approach taken in [4] where compatibility for properties is defined as those with do-

mains and ranges that are either the same or subtypes of one another.

Extensional property alignment techniques build upon the interplay between property and instance alignment. Extensional matchers consider two properties more similar if the instances related by these properties within a dataset are similar. Examples of this approach include [14] and [15]. Extensional property matchers often perform well in practice, but they can break down if there are few or no instances that involve a particular property. This is an issue because many ontologies have a large T-box but little or no A-box.

2.2. Empirical Analysis

The OAEI Conference track is the only established non-synthetic test set for alignment systems that has reference alignments containing matches between properties as well as classes. We therefore use this alignment task as the basis for our empirical analysis. We begin by investigating whether the differences in naming conventions for properties and classes identified within the previous subsection hold for this dataset. As Table 1 shows, the naming patterns do indeed differ, most notably through the prevalence of verbs for properties and nouns for classes. Additionally, classes are more likely to have single word labels than are properties in this case.

We next consider the performance of current alignment systems with respect to properties. Table 2 shows the results of the top 2016 OAEI competitors on the Conference track, broken down into classes and properties. Because some of the alignment systems participating that year were not focused on producing general alignments (i.e. they were focused specifically on aligning biomedical data), we only show the performance of alignment systems with an F-measure better than that of edna, an edit-distance string metric. The table shows that the average F-measure for classes is more than three times that for properties, and the recall of property matches of even the best-performing alignment systems is quite low. In fact, the last several systems do not generate any matches involving properties at all.

We then sought to further explore the performance of the alignment systems on properties by conducting an in-depth analysis of the true positives, false positives, and false negatives involving properties that were identified by most alignment systems. However, as Table 2 shows, there was an unusually low number of alignment systems performing better than the string benchmark on the Conference test set in 2016, and several of those generated no matches involving properties. We have therefore conducted our analysis using data from 2013. This allowed us to analyze data from 15 different systems rather than only five. Table 3 shows that the performance of most alignment systems on property matching has not changed significantly during this time span. (The exception is AML, which is the alignment system of one of this paper's authors.)

Appendix A contains tables showing the most common correct, false positive, and false negative property matches identified by the participants in the 2013 OAEI. The frequency column in the tables indicates the number of alignment systems out of the 15 qualifying[2] systems that produced (or failed to produce, in the case of false negatives) each match. This data shows that the equivalent properties that were most frequently correctly identified all have very high string similarity. This is not surprising, because many current alignment systems give a strong weight to syntactic label similarity.

Unfortunately, this reliance on high string similarity also leads current alignment systems to many false positives involving properties. Many matches with highly similar or identical labels are not valid. In some cases the domain or range of the matched properties indicate that they are not being used in the same way. For instance, the domain of cmt:name is the union of Person and Conference whereas the domain of sigkdd:Name is only Person and a separate property, Name_of_conference, is used to represent a conference's name. In other cases the match may make sense in isolation but would lead to logical inconsistency of the merged ontology. This leads us to potential insights into methods alignment systems could use to improve their performance on property matching.

Finally, we see that the properties involved in the most common false negatives generally have a much lower string similarity, such as cmt:hasBeenAssigned and ekaw:ReviewerOfPaper. Again, this is not surprising since many existing alignment systems rely heavily on string similarity. This strategy tends to perform reasonably well on classes and instances, but it is less effective on properties due to the prevalence of verbs in property labels and the variety of tenses and plurality that come into play (e.g. "I am" versus "we are"). In many of these cases, the domain and range of the properties *do* have strong syntactic similarity however, e.g. Reviewer and Paper for hasBeenAssigned and Possible_Reviewer and Paper for reviewerOfPaper. Further, there were some quite frequently missed equivalent properties that have strong clues in the labels themselves, such as cmt:writePaper and confOf:writes. Of the 31 common false negatives, 13 have noticeable string similarity. This raises the possibility that alignment systems could potentially improve their performance on property matching by utilizing their current approaches after first normalizing the property labels in some way. We take up this challenge in the next section.

3. A Baseline Approach

One of our goals with this work is to develop a basic approach to property matching that can serve as a reasonable baseline for matchers that attempt to align properties. Our approach relies primarily on string similarity and

[2]Those performing better than the basic edit-distance string metric

Naming Characteristic	Object Properties	Data Properties	Classes
Begins with any verb	0.74	0.59	0.09
Begins with an auxiliary verb	0.47	0.58	0.00
Ends with a preposition	0.40	0.09	0.00
Starts with a noun	0.22	0.38	0.74
More than one word	0.88	0.86	0.67

Table 1: Naming characteristics of classes and properties in the OAEI Conference track

System	Class Prec	Class Rec	Class Fms	Prop Prec	Prop Rec	Prop Fms
AML	0.83	0.7	0.76	1	0.41	0.58
CroMatch	0.78	0.74	0.76	0.94	0.35	0.51
LogMap	0.84	0.64	0.73	0.62	0.28	0.39
XMap	0.86	0.63	0.73	0.75	0.2	0.32
LogMapBio	0.8	0.71	0.64	0.27	0.11	0.07
DKPAOM	0.82	0.59	0.69	0	0	0
NAISC	0.85	0.55	0.67	0	0	0
FCAMap	0.75	0.61	0.67	0	0	0
Average	0.82	0.65	0.71	0.45	0.17	0.23

Table 2: Performance of the top 2016 OAEI competitors on classes versus properties

domain and range information. The focus on string similarity is in keeping with our belief that the label given to an entity is a very good indicator of its intended meaning and use. The use of domain and range information takes advantage of the basic structural information that is most commonly available for properties.

This concept of using a string-based approach as a baseline against which to compare full-featured alignment systems is standard for the field. For example, the OAEI compares all competing alignment systems against a string equality and an edit distance baseline. In addition, string metrics such as the one described here are often directly incorporated into a full-featured alignment system. We have shown in previous work that over half of the correct correspondences in the OAEI Conference track and 85 percent of those in the OAEI Anatomy track can be found using a string-based approach [2]. Identifying the remaining correspondences without generating an inordinate number of false positives is quite difficult and requires the capabilities of more advanced techniques, but the string metrics provide a valuable starting point.

3.1. Design

We have arrived at the following approach, called PropString[3], which we will explain with a running example in which cmt:writtenBy is compared with ekaw:reviewWrittenBy.

In the first step, four strings are extracted for each property: the label, the core concept, the domain, and the range. All strings are tokenized and put into lower case. The label is simply the entity's label. The core concept is either the first verb in the label that is greater than four characters long or, if there is no such verb, the first noun in the label, together with any adjectives that modify that noun. For example, the label "wrote paper" has the core concept "wrote" and the label "has corresponding author" has the core concept "corresponding author." We arrived at this technique through an analysis of common naming patterns for properties. We used the Stanford log-linear part of speech tagger to compute the core concept [16]. The domain/range string is a concatenation of the labels of any classes in the domain/range of a property.[4] This consideration of the lexical similarity of the domain and range of properties is somewhat similar to the work by Vizenor and his colleagues in [17]. Their approach, which is focused on the biomedical domain, used domain and range similarity as a sanity check on the alignment of properties. One question that might be asked is "why not concatenate the domain and range information onto the property's label and consider the entire thing as one string?" The reason this is not done is because it frequently confuses inverse properties, in which the domain of one property is the same as the range of the other and vice versa, as equivalent. The first two rows of Table 4 show this part of the algorithm for the running example.

The similarity of each of these four pairs of strings is then computed using a TF-IDF metric, which was the string metric shown in [2] to have the best performance on properties. TF-IDF is a bag-of-words approach that, in this case, considers two labels more similar if they share relatively uncommon words, as measured by their frequency across all property labels. For the entity label and core concept, the soft TF-IDF metric, trained on the properties from both ontologies, is used. Soft TF-IDF differs from

[3] http://michellecheatham.com/files/PropString.zip

[4] In the case of datatype properties, the range is set to "literal." This was done rather than using the actual datatype because many times information that is inherently numeric is encoded as a string.

System	Class Prec	Class Rec	Class Fms	Prop Prec	Prop Rec	Prop Fms
AML	0.86	0.62	0.72	1.00	0.20	0.33
AMLback	0.86	0.64	0.73	1.00	0.24	0.39
CIDER_CL	0.46	0.59	0.52	0.07	0.22	0.11
HerTUDA	0.84	0.56	0.67	0.26	0.20	0.23
HotMatch	0.81	0.57	0.67	0.24	0.20	0.22
IAMA	0.87	0.55	0.67	0.14	0.07	0.09
LogMap	0.82	0.65	0.73	0.62	0.28	0.39
MapSSS	0.74	0.59	0.66	0.00	0.00	0.00
ODGOMS	0.87	0.55	0.67	0.32	0.26	0.29
ODGOMS1_2	0.81	0.66	0.73	0.32	0.26	0.29
ServOMap_v104	0.74	0.65	0.69	0.00	0.00	0.00
StringsAuto	0.71	0.63	0.67	0.00	0.00	0.00
WeSeEMatch	0.85	0.54	0.66	0.50	0.02	0.04
WikiMatch	0.84	0.54	0.66	0.26	0.22	0.24
YAM++	0.82	0.71	0.76	0.68	0.57	0.62
Average	0.79	0.60	0.68	0.36	0.18	0.21

Table 3: Performance of the top 2013 OAEI competitors on classes versus properties

Property	Label	Core	Domain	Range
cmt:writtenBy	written by	written	Review	Reviewer
ekaw:reviewWrittenBy	review written by	written	Review	Possible_Reviewer
Similarities	Soft TF-IDF: 0.82	Soft TF-IDF: 1.0	TF-IDF: 1.0	TF-IDF: 0.45

Table 4: Example of the baseline property matching approach

TF-IDF in that it considers labels to have words in common if they have a syntactic similarity above some threshold, rather than only if they are exact matches. Here the internal threshold for that metric is set at 0.9. The similarity of the domain and range is computed using a standard TF-IDF metric trained on all entities from both ontologies, which was shown in [2] to have reasonable performance on classes in terms of both precision and recall. These values are shown in the third row of Table 4.

While the vast majority of alignment systems use a string similarity metric, they use them in different ways. One approach is to find highly precise "anchor" matches which serve as the seed that the rest of the alignment grows out from. Another approach is to use a string metric to filter out any obviously incorrect matches in order to reduce computational complexity. This requires a string metric with high recall. To address both of these use cases, our approach can be run in two configurations: precision-oriented and recall-oriented. In the precision-oriented mode, a pair of entities is considered a match if the similarity values for their core concepts, domains, **and** ranges are all greater than the threshold. In the recall-oriented mode, the pair is considered a match if the similarity values for their core concepts **or** their domains and ranges are greater than the threshold (0.9 by default). In the running example, cmt:writtenBy would not be considered a match in precision-oriented mode because the range similarity is less than the threshold, but it would be considered a match in the recall-oriented mode since the core similarity exceeds the threshold.

Allowing matches based solely on high similarity of domain and range in the recall-oriented configuration results in very low precision unless further steps are taken. We use a confidence value to reduce the number of false positives. The confidence value is calculated by averaging the similarity values for the labels, their domains, and their ranges. For example, the confidence value for the entity pair in Table 4 is the average of 0.82, 1.0 and 0.45, which is 0.76. We keep a list of each entity that is considered part of a match so far, along with the entity it maps to and the confidence value. Every time a new potential match between properties is identified, its confidence value is checked against any existing current matches involving those properties. If the new match has a greater confidence value, the old match is removed in favor of the new one, otherwise the new match is ignored. Using the label similarity when computing the confidence values rather than the core concept eliminates the loss of precision associated with extracting the core concept, effectively breaking any ties in favor of the closer lexical match. The overall effect is that any properties with the same domain and range act as a filter, with the specific match from that set chosen based on the actual property label. This is shown below, where the YAGO property "influences," with a domain and range of "Person," is being matched:

```
yago:influences = dbpedia:relative: 0.67
yago:influences = dbpedia:father: 0.67
yago:influences = dbpedia:mother: 0.67
yago:influences = dbpedia:spouse: 0.67
```

System	Precision	Recall	F-measure
PropString (prec)	1.0	0.26	0.41
PropString (rec)	0.34	0.5	0.4
Soft TF-IDF	0.2	0.24	0.22

Table 5: Results on the OAEI Conference track

```
yago:influences = dbpedia:influencedBy: 0.93
yago:influences = dbpedia:influenced: 0.99
```

3.2. Evaluation

Table 5 shows the results of PropString on the OAEI Conference track. The system was configured with a threshold of 0.9 and to only include matches in which both entities were in the namespace of the ontologies to be matched (in accordance with the OAEI guidelines). The results are compared with those of Soft TF-IDF with a threshold of 0.8. This was shown in [2] to be the best-performing string metric for property alignment. It is evident that PropString greatly outperforms Soft TF-IDF on this test set. The precision-oriented configuration of PropString quintuples the precision of Soft TF-IDF (to a perfect 1.0) while maintaining roughly the same recall. Analogously, the recall-oriented version doubles the recall of Soft TF-IDF while still achieving noticeably better precision. The F-measures for both the precision- and recall-oriented configurations are double that of Soft TF-IDF.

Comparing Tables 3 and 5, we also see that this relatively simplistic approach outperforms most full-featured alignment systems on this task in terms of F-measure. Recall that all of the matchers shown in Table 3 performed better than the string edit distance metric edna. However, the PropString approach uses only basic strategies beyond this. This is a potentially better baseline against which to measure the contributions of a full-featured alignment system on this task.

Our next goal was to evaluate the reasonableness of the confidence value PropString assigned to each match. To do this, we compared this value to the degree of concurrence on the match among a group of people familiar with ontology alignment. The 13 experts were given a link to download a Java program and accompanying data files. The entity labels from each match were stripped of the URL, tokenized, and put into lower case. Additionally, in order to provide the experts with some context for the labels, all of the axioms in the ontologies were translated to English using Open University's SWAT Natural Language Tools.[5] Any axioms related to either of the entities in the match were displayed to the users. Users were then asked whether each pair of entities were equivalent.

Appendix B contains a table that again shows the most commonly missed matches by the top-performing alignment systems from 2013, but this time with the percentage of experts who agreed with each match and the confidence value that PropString (running in the recall configuration with a threshold of 0.9) assigned to each of these matches. PropString was able to correctly identify 9 of these 31 matches, including 8 of the 22 on which more than half of the experts agreed (shown in bold in the table). This is quite encouraging considering that these were the most difficult matches for current alignment systems to identify. Several matches with limited or no label similarity were correctly found, such as edas:endDate = sigkdd:End_of_conference and conference:contributes = ekaw:authorOf.

It is also important for an alignment system (or similarity metric) to assign meaningful confidence values, so that users can select a similarity threshold that is appropriate for their application. In 2015 the OAEI began including an evaluation of how closely an alignment system's confidence value for each match reflects human opinion [18]. The table in Appendix B also shows that the confidence values assigned to the matches found by PropString have quite reasonable correlation to the percentage of experts who agreed with the matches. Among the nine matches found by PropString there is one case in which PropString correctly found the match while the experts did not. With that outlier omitted, the Pearson correlation coefficient is 0.73.

We now turn from a holistic evaluation of the performance of this approach to analyzing the effect of each aspect of the method on overall performance. This is somewhat difficult to do because the aspects do not stand alone – they influence one another. We take the approach of considering the impact of each design aspect when removed from the complete PropString approach (Table 6), as well as when added to the basic Soft TF-IDF metric (Table 7). We evaluate these impacts with respect to the OAEI Conference track benchmark.

Table 6 shows that there are not any superfluous aspects to the PropString metric – removing any element reduces performance. In particular, removing the idea of extracting the core concept from property labels has such a large effect on recall that the precision-oriented configuration becomes nearly useless. Similarly, using simple label similarity for the confidence value rather than averaging label, domain, and range similarity nearly cuts precision in half in the recall-oriented configuration. Consideration of domain and range in the similarity computation is shown to be the key to the PropString approach.

Table 7 shows that no approach in isolation can achieve the overall precision, recall, or F-measure of the complete PropString metric. Also, the table shows that extracting the core concept from the property labels and considering domain and range information independent of the property label both significantly improve recall, as designed. Further, we see that considering domain and range in addition to the property label has a very large impact on precision. Finally, training the soft TF-IDF metric on only properties rather than all entities did not improve results, but it also does not significantly negatively impact preci-

[5] http://swat.open.ac.uk/tools/

	Precision-Oriented			Recall-Oriented		
Configuration	Precision	Recall	F-measure	Precision	Recall	F-measure
PropString	1.00	0.26	0.41	0.34	0.50	0.40
Property-trained	1.00	0.26	0.41	0.35	0.50	0.41
Core concept	0.88	0.15	0.26	0.39	0.48	0.43
Confidence calculation	0.86	0.13	0.23	0.26	0.37	0.31
Domain/range	0.20	0.24	0.22	0.20	0.24	0.22
Soft TF-IDF	0.20	0.24	0.22	0.20	0.24	0.22

Table 6: Impact of individual components removed from PropString on performance on the OAEI Conference track

sion or recall, which is useful in the sense that it is more scalable and time-efficient. (Note that the precision and recall orientations are based on whether or not domain and range are required to be similar, so there is only one row in Table 7 that differs between the precision-oriented and recall-oriented configurations.)

4. A Proposed Benchmark

In the seven ontologies within the OAEI conference track for which reference alignments are available, the number of properties is on par with the number of classes (355 versus 558); however, the reference alignments contain significantly fewer matches between properties than between classes (46 versus 259). It would be helpful to have a second real-world alignment task with which to assess the performance of alignment systems when it comes to matching properties. In this section we propose such a benchmark and evaluate the performance of several alignment systems on it.

4.1. Proposed Property Matching Benchmark

We have chosen DBPedia 3.9[6] and YAGO[7] as the basis for this benchmark. Both DBPedia and YAGO contain millions of instances and thousands of schema-level entities. This scale is too large for many current alignment systems. We are specifically interested in aligning the properties of these two datasets, so we have extracted a cohesive subset of each one that will allow us to do this without requiring an inordinately long runtime. This was done using the following procedure:

1. For each property in YAGO, randomly choose five facts that involve the property. For properties with less than five facts, use all that are available.
2. Add the classes of every instance mentioned in the facts from step 1.
3. Randomly add up to five other facts related to the instances from step 1.
4. Repeat step 2 for any additional instances added during step 3.
5. Compute the "closure" of this set of entities by recursively retrieving all schema-related axioms related to any entity within our sample.

The procedure for creating the DBPedia sample was analogous, except that instead of randomly choosing the facts in step 1, we selected facts with the same instances as our YAGO sample when available. This is possible because, since DBPedia and YAGO both represent information from Wikipedia, there is an error-free mapping of instances that point to the same Wikipedia page. The characteristics of these dataset samples are shown in Table 8. We have published this dataset so that other researhers can make use of it.[8]

There is currently no curated alignment of the properties in the DBPedia and YAGO datasets. In [1], we began the process of creating one by using Amazon's Mechanical Turk crowdsourcing platform. A set of potential mappings is needed to bootstrap the crowdsourcing effort. We used the matches produced by the alignment systems PARIS [19], LogMap [20], and AgreementMakerLight (AML) [21], PropString and a basic string similarity metric for this purpose. We chose PARIS because it has already been used on a matching task involving DBPedia and YAGO. LogMap and AML were chosen due to their strong performance in the OAEI over the past many years.

In our previous work, we focused on creating a nuanced reference alignment that would be capable of evaluating the performance of alignment systems that identified subsumption as well as equivalence relationships between two ontologies. Unfortunately, preliminary testing showed that the crowdsourcing results on identifying subsumption (rather than equivalence relations) were not very reliable. Others have indicated problems with scammers for these tasks as well [22]. We therefore invited only Turkers who had previously demonstrated good performance on alignment verification tasks to participate in that one, which meant we only received input from six or seven individuals for each match.

In the new work presented here, we advance the DBPedia-YAGO reference alignment beyond what was presented in [1] in two ways. First, we fall back to only validating equivalence relations, for which crowdsourcing techniques are

[6] http://wiki.dbpedia.org/Downloads39

[7] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/

[8] http://www.michellecheatham.com/files/dbpedia-yago.zip

Configuration	Precision-Oriented			Recall-Oriented		
	Precision	Recall	F-measure	Precision	Recall	F-measure
Soft TF-IDF	0.2	0.24	0.22	0.2	0.24	0.22
Property-trained	0.19	0.24	0.21	0.19	0.24	0.21
Core concept	0.17	0.35	0.23	0.17	0.35	0.23
Confidence calculation	0.38	0.41	0.39	0.38	0.41	0.39
Domain/range	1.00	0.26	0.41	0.34	0.50	0.40
PropString	1.00	0.26	0.41	0.34	0.50	0.40

Table 7: Impact of individual components added to Soft TF-IDF on performance on the OAEI Conference track

Dataset	DBPedia	YAGO
Classes	617	10962
Object Properties	1046	85
Data Properties	1407	37
Named Individuals	8685	1680
Datatypes	23	23
Annotations	77	125
Total Entities	11855	12912

Table 8: Characteristics of the DBPedia and YAGO samples

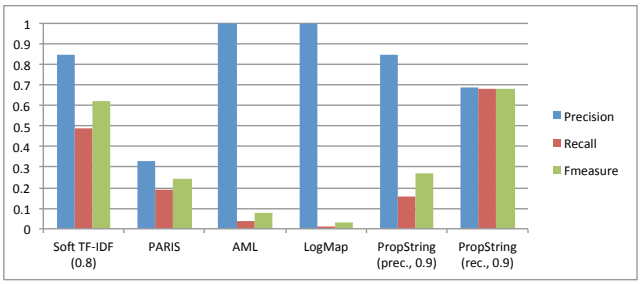

Figure 2: Results of the YAGO-DBPedia alignment task

more reliable. While this is obviously a compromise, the vast majority of current alignment systems focus solely on identifying equivalence relations, and this approach has allowed us to increase the number of respondents per match from six to 40 and achieve good consensus on the validity of each match. Secondly, we expand the group of alignment systems used to provide the candidate matches by including LogMap and AML.

Combining the matches generated by each matcher resulted in a set of 147 potential matches. We asked all 40 workers to opine on the validity of each of these. In order to provide the workers with some context on which to base their answer, we provided information about the domain and range of each property and up to five examples of instances with values for each property. An example of one of these questions is shown below:

```
A "person" has a property called "directed"
that involves a "thing." Examples are:

dario argento -> four flies on grey velvet
eldor urazbayev -> tailcoat for shalopaya
masahiro shinoda -> gonza the spearman
jon monday -> the last straw film
d. w. griffith -> the fight for freedom


A "thing" has a property called "director" that
involves a "person." Examples are:

la rabbia -> pier paolo pasolini
two living, one dead -> anthony asquith
sasneham -> sathyan anthikad
la rabbia -> giovannino guareschi

smart blonde -> frank mc donald (director)

Does "directed" mean the same thing as "director"?
```

We consider a match to be valid if at least half of the 40 Mechanical Turk participants agreed with it. This yielded a set of 68 valid matches. The percentage of agreement with the consensus matcher across all prospective matches was 0.81, which is essentially the same as the degree of consensus shown among ontology alignment experts [23].

4.2. Initial Benchmark Evaluation

We then evaluated the performance of each of the matching systems in isolation on this alignment task. Figure 2 compares the results. PropString was run in both its precision and recall configurations. The threshold for Soft TF-IDF was set to 0.8 and the threshold for PropString was set to 0.9, based upon the best-performing thresholds for these approaches on the Conference track. AML was run with a threshold of 0.7. We tried several different values for the LogMap threshold, going as low as 0.2, but that system only produced one equivalent property relation.

The basic string metric Soft TF-IDF produces the highest precision. Further, that precision is 0.85, which is on par with the degree of agreement among the Turkers on these matches. So we see that a straightforward string metric can in some ways outperform more sophisticated alignment strategies. In fact, PARIS, AML, LogMap and the precision-oriented configuration of PropString have such low recall that they may not be of much utility for many application scenarios. This is surprising considering the strong performance of these matchers on the properties

within the Conference track. We feel that this wide variation in performance is further indication of the need for more benchmarks involving property alignment.

Another thing to note from these results is the strong performance of the recall-based configuration of PropString, both relative to the other approaches and in an absolute sense. We argue that the basic property matching technique described here, which also considers the lexical similarity of domain and range, is the more appropriate string-based baseline against which to compare alignment systems on property matching tasks: a pure string metric baseline will likely make the results of many matching systems appear quite good, when this performance is achieved largely though basic domain and range restrictions and greedy culling of redundant matches rather than the more complex (and computationally expensive) strategies the matcher may employ.

Of course, the preliminary nature of the YAGO-DBPedia reference alignment must be kept in mind. More work, involving the incorporation of results produced by many other alignment system on this pair of ontologies, is still needed to confirm these results. Since a limited number of alignment systems have been made public, this will take a community effort on the part of researchers in the field.

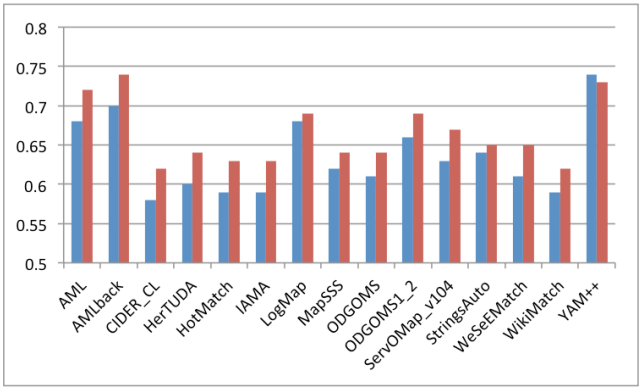

Figure 3: Performance of alignment systems on the OAEI Conference track without (left/blue) and with (right/red) PropString included.

5. Application to an Existing Matcher

Because our baseline outperforms the full-featured alignment systems it was compared against on the conference and DBPedia-YAGO test sets, it is natural to wonder whether the performance of these systems could be improved by leveraging techniques from the baseline approach. We therefore explore this possibility in this section. We begin with a naive inclusion of our lexical property matching techniques to the full-featured systems by simply replacing all property matches identified by each system with those suggested by the precision-oriented version of our techniques. This integration has the benefit of not requiring the source code for the matcher. This approach improves the overall F-measure of the full-featured alignment systems by an average of 5 percent on the OAEI conference test set (see Figure 3).

We also explored a deeper integration of our baseline property alignment approach with an existing alignment system. We chose AgreementMakerLight for this because AML was designed from the beginning to facilitate the inclusion of essentially any matching algorithm.

AML considers a matcher to be any algorithm that takes two ontologies and an optional preliminary alignment between them and returns an alignment relating entities from each ontology. When given two ontologies, the system profiles the ontologies and according to the identified profile runs a set of different matchers. The outputs of the matchers are combined simply by joining the alignments, and then different selection strategies can be employed to achieve the desired cardinality. Furthermore, the logical validity of the alignment is checked and inconsistent mappings are removed. In its default configuration, AML employs three matchers: lexical (literal name matches), word matcher (weighted Jaccard index between words in the entity names), and parametric string (a configurable string similarity metric with pre-processing such as stop word removal, word stemming, etc.). These can be used as primary or secondary matchers according to the ontology profile. A greedy selection approach then removes all mappings that have confidence values below a threshold or that conflict with mappings which have a higher confidence value. For this evaluation we added the lexical property matching approach described in Section 3 as a fourth matcher.

The results of using AML in this configuration are presented in Figures 4 (Conference track) and 5 (YAGO-DBPedia). These results indicate that integrating lexical property matching in this manner is not very effective: the modified version of AML performs slightly better on the conference track when run in the precision-oriented configuration, but significantly worse on that dataset when recall is favored. Performance of the combined system in both the precision- and recall-oriented modes is better than the original AML but remains unacceptably low. The underlying issue is that the other three matchers within AML often found incorrect property matches that the property-centric lexical matcher disagreed with, while frequently voting down matches from the property-centric module that were actually correct.

The results of the two experiments described in this section show that improving performance of existing alignment systems is indeed possible, but more work will need to be done to leverage lexical property matching metrics to the fullest extent possible. In particular, it is unlikely that a one-size-fits-all, or even an ensemble, approach is going to lead to strong results on matching tasks that involve both properties and classes. It would be interesting to explore the performance of alignment techniques that exploit the interrelation of classes and properties within an ontology. This could be done by modifying alignment systems

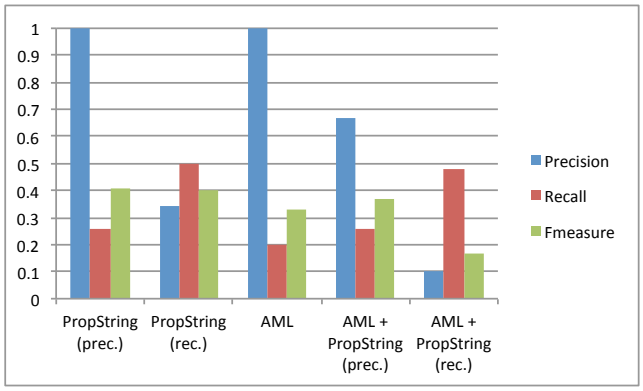

Figure 4: Performance on the OAEI Conference track

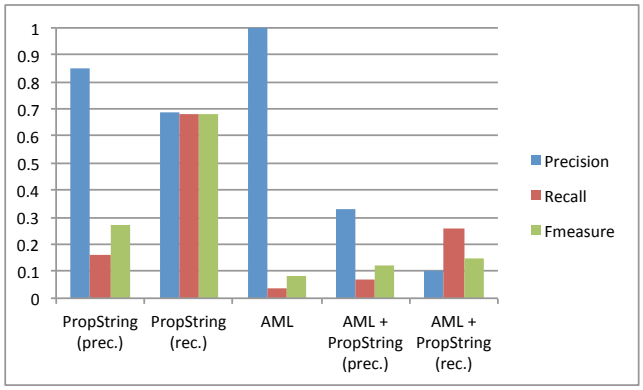

Figure 5: Performance on YAGO-DBPedia

to recognize, for example, that two properties with matching domains and ranges are more likely to themselves be matches[9] and two classes that involve the same properties in the same way are more likely to be related, using the well-known similarity flooding technique [10]. Unfortunately, seeing the impact of this type of cross-fertilization requires an alignment test case (and associated reference alignment) that contains large numbers of both properties and classes, and such a test case has not yet been established. This remains an important area of future work. It is possible that such reference alignments could be created and validated using hybrid automated and interactive techniques such as those described in [24].

6. Conclusions

This work explored the performance of current ontology alignment systems on property alignment using the OAEI Conference track as a benchmark. The paper also introduced PropString, a basic approach to aligning properties intended to serve as a baseline for more complex alignment systems on property matching tasks. The PropString approach was shown to perform better than the best-performing string metric by a wide margin. It also compared favorably to many full-featured alignment systems. Additionally, the confidence values generated by this approach were shown to correlate well with the degree of expert agreement on property matches. Furthermore, a second benchmark involving property matches was suggested, with a reference alignment generated using a crowdsourcing approach via Mechanical Turk. The performance of several matchers was evaluated in order to provide baseline results for this alignment task. Finally, the baseline property matching approach was integrated into the AML alignment system, though the results are difficult to evaluate given the lack of benchmarks that have a significant number of both classes and properties.

This work addressed several limitations of our previous work on this topic [1]. However, more work remains to be done. In particular, research on property matching requires more benchmarks highlighting this aspect of ontology alignment. Our work on the YAGO-DBPedia reference alignment is a good start, but it needs to be refined by including potential matches generated by more alignment systems. These matches can then be manually verified, either through Mechanical Turk or by experts. Correspondingly, additional experimentation regarding crowdsourcing reference alignments using Mechanical Turk needs to be done to verify the potential uses of the approach. For instance, our preliminary results showed that general users can often give good input on "yes or no" alignment verification tasks but that more complex questions regarding the type of relationship between two entities (e.g. equivalence, subsumption, inverse properties) is more difficult [25]. It would be useful to develop guidelines for when and how to qualify users for different types of alignment tasks.

Funding This work was supported by the National Science Foundation award ICER-1440202 "EarthCube Building Blocks: Collaborative Proposal: GeoLink - Leveraging Semantics and Linked Data for Data Sharing and Discovery in the Geosciences." and the Portuguese Funda cao para a Ciecia e Tecnologia UID/CEC/00408/2013 and PTDC/EEI-ESS/4633/2014.

[9] While the baseline property matching approach described in Section 3 takes this into consideration, it is limited by only making lexical similarity comparisons rather than the richer comparisons possible through a full-featured alignment system.

References

[1] M. Cheatham, P. Hitzler, The properties of property alignment, in: Proceedings of the 9th International Conference on Ontology Matching-Volume 1317, CEUR-WS. org, 2014, pp. 13–24.

[2] M. Cheatham, P. Hitzler, String similarity metrics for ontology alignment, in: International Semantic Web Conference, Springer, 2013, pp. 294–309.

[3] A. Maedche, S. Staab, Measuring similarity between ontologies, in: International Conference on Knowledge Engineering and Knowledge Management, Springer, 2002, pp. 251–263.

[4] N. Pernelle, F. Saïs, B. Safar, M. Koutraki, T. Ghosh, N2r-part: identity link discovery using partially aligned ontologies, in: Proceedings of the 2nd International Workshop on Open Data, ACM, 2013, p. 6.

[5] V. Svátek, O. Šváb-Zamazal, V. Presutti, Ontology naming pattern sauce for (human and computer) gourmets, in: Proceedings of the 2009 International Conference on Ontology Patterns-Volume 516, CEUR-WS. org, 2009, pp. 171–178.
[6] O. Šváb, Exploiting patterns in ontology mapping, in: The Semantic Web, Springer, 2007, pp. 956–960.
[7] N. F. Noy, C. D. Hafner, The state of the art in ontology design: A survey and comparative review, AI magazine 18 (3) (1997) 53.
[8] O. Sváb-Zamazal, V. Svátek, F. Scharffe, et al., Pattern-based ontology transformation service, in: Proc. 1st IK3C international conference on knowledge engineering and ontology development (KEOD), 2009, pp. 210–223.
[9] D. Ritze, C. Meilicke, O. Šváb-Zamazal, H. Stuckenschmidt, A pattern-based ontology matching approach for detecting complex correspondences, in: Proceedings of the 4th International Conference on Ontology Matching-Volume 551, CEUR-WS. org, 2009, pp. 25–36.
[10] S. Melnik, H. Garcia-Molina, E. Rahm, Similarity flooding: A versatile graph matching algorithm and its application to schema matching, in: Data Engineering, 2002. Proceedings. 18th International Conference on, IEEE, 2002, pp. 117–128.
[11] J. Li, J. Tang, Y. Li, Q. Luo, Rimom: A dynamic multistrategy ontology alignment framework, IEEE Transactions on Knowledge and Data Engineering 21 (8) (2009) 1218–1232.
[12] F. Suchanek, S. Abiteboul, P. Senellart, Ontology alignment at the instance and schema level, arXiv preprint arXiv:1105.5516.
[13] L. Zhao, R. Ichise, Instance-based ontological knowledge acquisition, in: Extended Semantic Web Conference, Springer, 2013, pp. 155–169.
[14] K. Gunaratna, K. Thirunarayan, P. Jain, A. Sheth, S. Wijeratne, A statistical and schema independent approach to identify equivalent properties on linked data, in: Proceedings of the 9th International Conference on Semantic Systems, ACM, 2013, pp. 33–40.
[15] Y. Liu, S.-H. Chen, J.-G. Gu, Property alignment of linked data based on similarity between functions, order 8 (4).
[16] K. Toutanova, D. Klein, C. D. Manning, Y. Singer, Feature-rich part-of-speech tagging with a cyclic dependency network, in: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, Association for Computational Linguistics, 2003, pp. 173–180.
[17] L. T. Vizenor, O. Bodenreider, A. T. McCray, Auditing associative relations across two knowledge sources, Journal of biomedical informatics 42 (3) (2009) 426–439.
[18] M. Cheatham, Z. Dragisic, J. Euzenat, D. Faria, A. Ferrara, G. Flouris, I. Fundulaki, R. Granada, V. Ivanova, E. Jiménez-Ruiz, et al., Results of the ontology alignment evaluation initiative 2015, in: 10th ISWC workshop on ontology matching (OM), No commercial editor., 2015, pp. 60–115.
[19] F. M. Suchanek, S. Abiteboul, P. Senellart, Paris: Probabilistic alignment of relations, instances, and schema, Proceedings of the VLDB Endowment 5 (3) (2011) 157–168.
[20] E. Jiménez-Ruiz, B. C. Grau, Logmap: Logic-based and scalable ontology matching, in: International Semantic Web Conference, Springer, 2011, pp. 273–288.
[21] D. Faria, C. Pesquita, E. Santos, M. Palmonari, I. F. Cruz, F. M. Couto, The agreementmakerlight ontology matching system, in: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems", Springer, 2013, pp. 527–541.
[22] J. Mortensen, M. A. Musen, N. F. Noy, Crowdsourcing the verification of relationships in biomedical ontologies., in: AMIA, 2013.
[23] M. Cheatham, P. Hitzler, Conference v2. 0: An uncertain version of the oaei conference benchmark, in: International Semantic Web Conference, Springer, 2014, pp. 33–48.
[24] Z. Dragisic, V. Ivanova, P. Lambrix, D. Faria, E. Jiménez-Ruiz, C. Pesquita, User validation in ontology alignment, in: International Semantic Web Conference, Springer, 2016, pp. 200–217.
[25] R. Amini, Towards best practices for crowdsourcing ontology alignment benchmarks, Ph.D. thesis, Wright State University (2016).

Appendix A

Property 1	Property 2	Freq.
cmt:email	confOf:hasEmail	11
confOf:hasFirstName	edas:hasFirstName	11
conference:has_an_email	confOf:hasEmail	9
cmt:email	conference:has_an_email	9
conference:has_the_last_name	edas:hasLastName	9
conference:has_a_review	ekaw:hasReview	9
conference:has_the_first_name	edas:hasFirstName	9
conference:has_the_first_name	confOf:hasFirstName	9

Table 9: Most common correct property matches identified by alignment systems in the 2013 OAEI

Property 1	Property 2	Freq.
iasted:pay	sigkdd:pay	9
confOf:hasEmail	edas:hasEmail	9
cmt:email	edas:hasEmail	8
cmt:name	sigkdd:Name	8
confOf:hasPhone	edas:hasPhone	8
confOf:hasStreet	edas:hasStreet	7
confOf:hasPostalCode	edas:hasPostalCode	7
iasted:obtain	sigkdd:obtain	7
confOf:hasTopic	edas:hasTopic	7
conference:has_an_email	edas:hasEmail	7
cmt:writtenBy	confOf:writtenBy	7

Table 10: Most common false positive property matches identified by alignment systems in the 2013 OAEI

Property 1	Property 2	Freq.
cmt:hasBeenAssigned	ekaw:reviewerOfPaper	15
cmt:assignExternalReviewer	conference:invites_co-reviewers	15
cmt:assignedByReviewer	conference:invited_by	15
edas:endDate	sigkdd:End_of_conference	15
conference:is_given_by	sigkdd:presentationed_by	15
conference:has_a...tutorial_topic	confOf:hasTopic	15
conference:contributes	iasted:write	15
cmt:hasBeenAssigned	confOf:reviewes	15
conference:gives_presentations	sigkdd:presentation	15
conference:has_the_last_name	confOf:hasSurname	15
cmt:assignedTo	ekaw:hasReviewer	15
confOf:reviewes	edas:isReviewing	15
confOf:hasSurname	edas:hasLastName	15
conference:has_a_review_expertise	edas:hasRating	15
cmt:writtenBy	ekaw:reviewWrittenBy	15
cmt:hasSubjectArea	confOf:dealsWith	14
cmt:writePaper	confOf:writes	14
edas:isReviewedBy	ekaw:hasReviewer	14
cmt:hasAuthor	confOf:writtenBy	14
confOf:writes	edas:hasRelatedPaper	14
edas:hasCostAmount	sigkdd:Price	14
cmt:assignedTo	edas:isReviewedBy	14
edas:startDate	sigkdd:Start_of_conference	14
cmt:hasConferenceMember	edas:hasMember	14
cmt:hasBeenAssigned	edas:isReviewing	14
edas:hasLocation	ekaw:heldIn	14
edas:hasName	sigkdd:Name_of_conference	14
edas:isReviewing	ekaw:reviewerOfPaper	14
confOf:hasEmail	sigkdd:E-mail	13
conference:has_an_email	sigkdd:E-mail	13
conference:contributes	ekaw:authorOf	13

Table 11: Most common correct false negative property matches by alignment systems in the 2013 OAEI

Appendix B

Property 1	Property 2	Expt.	Sys.
cmt:hasBeenAssigned	ekaw:reviewerOfPaper	0.46	0.0
cmt:assignExternalReviewer	conference:invites_co-reviewers	0.54	0.0
cmt:assignedByReviewer	conference:invited_by	0.31	0.0
edas:endDate	**sigkdd:End_of_conference**	**0.85**	**0.84**
conference:is_given_by	sigkdd:presentationed_by	0.85	0.0
conference:has_a_track-workshop-tutorial_topic	confOf:hasTopic	0.31	0.0
conference:contributes	iasted:write	0.31	0.0
cmt:hasBeenAssigned	confOf:reviewes	0.62	0.0
conference:gives_presentations	sigkdd:presentation	0.46	0.0
conference:has_the_last_name	confOf:hasSurname	0.92	0.0
cmt:assignedTo	ekaw:hasReviewer	0.69	0.0
confOf:reviewes	edas:isReviewing	0.77	0.0
confOf:hasSurname	edas:hasLastName	1.0	0.0
conference:has_a_review_expertise	edas:hasRating	0.23	0.0
cmt:writtenBy	**ekaw:reviewWrittenBy**	**0.69**	**0.75**
cmt:hasSubjectArea	confOf:dealsWith	0.46	0.0
cmt:writePaper	**confOf:writes**	**0.62**	**0.61**
edas:isReviewedBy	**ekaw:hasReviewer**	**0.92**	**0.67**
cmt:hasAuthor	confOf:writtenBy	1.0	0.0
confOf:writes	edas:hasRelatedPaper	0.23	0.0
edas:hasCostAmount	sigkdd:Price	0.85	0.0
cmt:assignedTo	edas:isReviewedBy	0.92	0.0
edas:startDate	**sigkdd:Start_of_conference**	**0.92**	**0.84**
cmt:hasConferenceMember	edas:hasMember	0.54	0.0
cmt:hasBeenAssigned	edas:isReviewing	0.62	0.0
edas:hasLocation	ekaw:heldIn	0.92	0.0
edas:hasName	**sigkdd:Name_of_conference**	**0.08**	**0.85**
edas:isReviewing	ekaw:reviewerOfPaper	1.0	0.0
confOf:hasEmail	**sigkdd:E-mail**	**0.92**	**0.87**
conference:has_an_email	**sigkdd:E-mail**	**0.92**	**0.86**
conference:contributes	**ekaw:authorOf**	**0.69**	**0.63**

Table 12: PropString performance on the most common correct property matches omitted by alignment systems in the 2013 OAEI