String Similarity Metrics for Ontology Alignment

Michelle Cheatham and Pascal Hitzler

Kno.e.sis Center, Wright State University, Dayton OH 45435, USA

Abstract. Ontology alignment is an important part of enabling the semantic web to reach its full potential. The vast majority of ontology alignment systems use one or more string similarity metrics, but often the choice of which metrics to use is not given much attention. In this work we evaluate a wide range of such metrics, along with string pre-processing strategies such as removing stop words and considering synonyms, on different types of ontologies. We also present a set of guidelines on when to use which metric. We furthermore show that if optimal string similarity metrics are chosen, those alone can produce alignments that are competitive with the state of the art in ontology alignment systems. Finally, we examine the improvements possible to an existing ontology alignment system using an automated string metric selection strategy based upon the characteristics of the ontologies to be aligned.

1 Introduction

An ontology is a representation of the concepts in a domain and how they relate to one another. Engineering new ontologies is not a deterministic process – many design decisions must be made, and the designers' backgrounds and the target application will influence their decisions in different ways. The end result is that two ontologies that represent the same domain will not be the same. The goal of ontology alignment is to determine when an entity in one ontology is semantically related to an entity in another ontology (for a comprehensive discussion of ontology alignment, including a formal definition, see [9]).

Dozens of ontology alignment systems have been developed over the last decade, and nearly all of them use of a string similarity metric. But despite their ubiquity, there has been little systematic analysis on which metrics perform well when applied to ontology alignment. This paper fills that gap by analyzing such metrics in this domain, as well as the utility of string pre-processing approaches such as tokenization, translation, synonym lookup, and others. In particular, we seek to answer the following questions in this paper:

- What is the most effective string similarity metric for ontology alignment if the primary concern is precision? recall? f-measure?
- Does the best metric vary w.r.t. the nature of the ontologies being aligned?
- Does the performance of the metrics vary between classes and properties?

- Do string pre-processing strategies such as tokenization, synonym lookup, translations, normalization, etc improve ontology alignment results?
- What is the best we can do on the ontology alignment task using only string pre-processing and string similarity metrics?
- When faced with the task of aligning two ontologies, how can we automatically select which string similarity metrics and pre-processing strategies are best, without any training data available?
- How much does using optimized string similarity metrics improve an existing ontology alignment system?

Recent work by Ngo and his colleagues has analyzed the performance of some string metrics for ontology alignment and their interaction with structural and semantic metrics [15]. There has also been some prior analysis of string similarity metrics in the context of ontology alignment as part of the development of a new string similarity metric designed specifically for this domain done by Stoilos and his colleagues [16]. They compared the performance of a variety of string metrics on a subset of the Ontology Alignment Evaluation Initiative (OAEI)[1] benchmark test set and determined that the performance among metrics varied considerably. Another piece of work done in this area is a report produced by the Knowledge Web Consortium in 2004 that contained a description of a variety string (terminological) metrics and string normalization and stemming applied to the problem of ontology alignment [10].

When the area of interest is expanded to include string similarity metric studies for other domains, we find some more interesting surveys. For instance, Branting looked at string similarity metrics as applied to name matching in legal case files [4]. He evaluated the performance of various combinations of normalization, indexing (determining which names would be compared to one another) and similarity metrics. In addition, Cohen et al. did a very thorough analysis of string similarity metrics as applied to name matching tasks [7].

There has also been some (unsystematic) analysis of string similarity metric performance in the course of developing ontology alignment systems [3,8,11].

While string similarity metrics are certainly not a new area of research, it remains unclear which string metrics are best for ontology alignment. Since nearly all alignment algorithms use a string similarity metric, more clarity in this area would be of benefit to many researchers. The work presented here expands on the previous efforts discussed above by considering a wider variety of string metrics, string pre-processing strategies, and ontology types. It also takes the work further by placing the string metrics into a complete ontology alignment system and comparing the results of that system to the current state of the art.

The rest of the paper is structured as follows. In section 2, we give an overview of existing string similarity metrics and pre-processing strategies. In section 3 we describe our experimental setup, followed by the results of the experiments in section 4. We conclude in section 5. A more in-depth version of this paper that includes implementation details sufficient to reproduce these results can be found in a technical report [6].

[1] http://oaei.ontologymatching.org/

2 String Similarity Metrics and Pre-processing Strategies

The OAEI has become the primary venue for work in ontology alignment. Since 2006, participants in the OAEI competition have been required to submit a short paper describing their approach and results. We surveyed all of these papers to determine what lexical metrics were employed and what pre-processing steps were used (or proposed for use). In cases where the paper was not explicit about the string similarity metric used, the code for the alignment algorithm was downloaded and examined when possible.

We can group string metrics along three major axes: global versus local, set versus whole string, and perfect-sequence versus imperfect-sequence. Global versus local refers to the amount of information the metric needs in order to classify a pair of strings as a match or a non-match. Global metrics must compute some information over all of the strings in one or both ontologies before it can match any strings whereas for local metrics the pair of strings currently being considered is all the input that is required. Global metrics can be more tuned to the particular ontology pair being matched, but that comes at the price of increased time complexity. Perfect-sequence metrics require characters to occur in the same position in both strings in order to be considered a match. Imperfect-sequence metrics equate matching characters as long as their positions in the strings differ by less than some threshold. In some metrics, this threshold is the entire length of the string. Imperfect-sequence metrics are thought to perform better when the word ordering of labels might differ but may result in more false positives. A set-based string metric works by finding the degree of overlap between the words contained in two strings. The set-based metric must still use a basic string metric to establish if the individual tokens are equal. Word-based set metrics are generally thought to perform well on long strings.

The list below contains all metrics found in the review of OAEI participants and categorizes them based on the classifications described above. For set-based metrics, the underlying base metric used is given in parentheses. A subset of these metrics (shown in bold) has been chosen for analysis related to various aspects of the ontology alignment problem. These metrics were chosen to reflect those most commonly used in existing alignment systems as well as to cover as fully as possible all combinations of the classification system provided.

- Set, Global, Perfect-sequence
  - Evidence Content (with exact) – Similar to Jaccard mentioned below, but words are weighted based on their evidence content (a function of their frequency in the ontology)
  - **TF-IDF (with exact match)** – Term Frequency / Inverse Document Frequency; the idea is that two entities are more similar if they share a word that is rare in the ontologies
- Set, Global, Imperfect-sequence
  - **Soft TF-IDF (with Jaro Winkler)** – A variant of TF-IDF that considers words equal based on Jaro Winkler (mentioned below) rather than exact match

- Set, Local, Perfect-sequence
  - **Jaccard (with exact match)** – The number of words two strings have in common divided by the total number of unique words
  - Overlap Coefficient (with exact) – The number of words two strings have in common divided by the number of words in the smaller string
- Set, Local, Imperfect-sequence
  - RWSA – Redundant, Word-by-word, Symmetrical, Approximate; strings are indexed by their Soundex representation and are a match if each word in the smaller string has a weighted edit distance less than a threshold for a word in the longer string [4]
  - **Soft Jaccard (with Levenstein)** – Levenstein is run on all pairs of words in both strings and the number less than the threshold is counted and divided by the number of words in the longer string
- Non-set, Global, Perfect-sequence
  - None
- Non-set, Global, Imperfect-sequence
  - COCLU – Compression-based Clustering; a Huffman tree is used to cluster the strings based on a metric called the Cluster Code Difference, and strings in the same cluster are considered equivalent [17]
- Non-set, Local, Perfect-sequence
  - **Exact Match** – Checks for string equality
  - **Longest Common Substring** – The length of the largest substring common to both strings, normalized by the length of the strings
  - Prefix – Checks if the first string is a prefix of the second
  - Substring Inclusion – Whether the first string is contained in the second
  - Suffix – Whether the first string is a suffix of the second
- Non-set, Local, Imperfect-sequence
  - Jaro – Based on the number of matching and transposed characters, where characters match if they are within a window based on the lengths of the strings and are transposed if they match but are in reverse order
  - **Jaro Winkler** – Variation of the Jaro metric that gives a preference to strings that share a common prefix
  - **Levenstein** – The number of insertions, deletions, and substitutions required to transform one string to another; also called edit distance
  - Lin – Based on the idea that similarity between two strings can be determined by taking a measure of what they have in common and dividing by a measure of the information it takes to describe them [12]
  - **Monge Elkan** – A variant of Smith Waterman (see below) with non-linear gap penalties, approximate character matching, and particular parameter values [14]
  - **N-gram** – Converts strings into sets of n-grams (we use n=3); the resulting sets are compared using a set similarity metric such as cosine similarity or Dice's coefficient
  - Normalized Hamming Distance – The number of substitutions required to transform one string into another, divided by the length of the string

- Smith Waterman – Uses dynamic programming over a matrix describing the matches, insertions, and deletions between two strings
- Smith Waterman Gotoh – A variant of the Smith-Waterman metric that has affine gap penalties
- **Stoilos** – Specifically developed for ontology alignment, this metric explicitly considers both the commonalities and the differences of the strings being compared [16]
- String Matching (SM) – A variant of Levenstein in which the difference between the length of the shorter string and the edit distance is divided by the length of the shorter string [13]

Often alignment algorithms modify the strings before computing their similarity. All of the pre-processing approaches that were either tried or proposed by OAEI participants are listed here. The approaches mentioned by more than two participants are shown in bold – these will be examined in detail.

These approaches can be divided into two major categories: syntactic and semantic. Syntactic pre-processing methods are based on the characters in the strings or the rules of the language in which the strings are written. They can generally be applied quickly and without reference to an outside data store. Semantic methods relate to the meanings of the strings. These methods generally involve using a dictionary, thesaurus, or translation service to retrieve more information about a word or phrase.

- Syntactic
  - **tokenization** – Splitting strings into their component words based on delimiters and camelCase
  - split compound words
  - **normalization** – Elimination of stylistic differences due to capitalization, punctuation, word order, and characters not in the Latin alphabet
  - **stemming/lemmatization** – Elimination of grammatical differences due to verb tense, plurals, etc. We use the Porter stemming algorithm
  - **stop word removal** – Removal of very common words. The Glasgow stop word list is used in this work[2]
  - consider part-of-speech – "Functional" words such as articles, conjunctions, and prepositions are weighted less (or removed completely)
- Semantic
  - **synonyms** – Strings are supplemented with their synonyms
  - antonyms – Used with metrics considering differences and commonalities
  - categorization – An external source containing a category hierarchy is used. Strings falling into the same category are considered more similar.
  - language tag – Leverage language tags contained in some ontologies to avoid comparing labels in different languages or as an aid for translation
  - **translations** – Strings are translated before they are compared. We have used Google Translate.
  - expand abbreviations and acronyms – There have been several attempts to do such expansions into long form, by either looking them up in external knowledge sources or using language production rules

[2] http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words

3 Experimental Setup

In this section we describe the experimental framework. More detail about the implementation can be found in the technical report at [6]. In addition, the source code for these experiments can be downloaded from http://www.pascal-hitzler.de/pub/StringMetricTester.jar.

The OAEI conference track consists of finding equivalence relations among 16 relatively small real-world ontologies describing the same domain – conference organization. The multifarm track consists of the ontologies from the conference track translated by native speakers into eight different languages. The goal is to align all combinations of languages. Finally, the anatomy track consists of two ontologies from the biomedical domain: one describing the anatomy of a mouse and the other the anatomy of a human. As is common for biomedical ontologies, these are significantly larger than those found in the conference track, with each containing around 3000 classes.

In order to get a sense of whether the results on the OAEI test sets generalize to similar cases, we have also run our tests on other ontology pairs of the same type. As an analog to the conference test set, we have used two BizTalk files representing the domain of purchase orders: CIDX and Excel.[3] In addition, native speakers have assisted us in translating these schemas into German, Portuguese, Finish, and Norwegian so that we also have an analog for the OAEI multifarm track. Finally, we have attempted to match the Gene Ontology[4] to the multifun schema,[5] both of which cover topics from biomedicine (the Gene Ontology covers the general domain of genetics, while the multifun schema is a description of cell function). The reference alignment for this test set was generated by domain experts.[6] The GO ontology and associated schema mappings are made possible by the work of the Gene Ontology Consortium [2].

Our test framework takes the two ontologies to be aligned and compares the label of every entity in the first ontology to every entity in the second. For each pair of labels, the "metric" being tested is computed for both permutations of the pair. (The measurements are asymmetric, so are technically not metrics.) These results are put into two separate two dimensional arrays. Then the stable marriage algorithm is run on these two arrays to determine the best matches. Finally, any mappings for which the minimum of metric.compute(labelA, labelB) and metric.compute(labelB, labelA) is less than one threshold or the maximum of those two values is less than a second threshold are thrown out. Due to the nature of the test framework, each metric requires at least two parameters: the thresholds for the similarities between the two strings (in both directions). In addition, the soft set metrics (soft Jaccard and soft TF-IDF) require an additional parameter – the threshold for the base similarity metric. These parameters were systematically modified in increments of 0.1 to reasonably thoroughly cover the search space while optimizing f-measure.

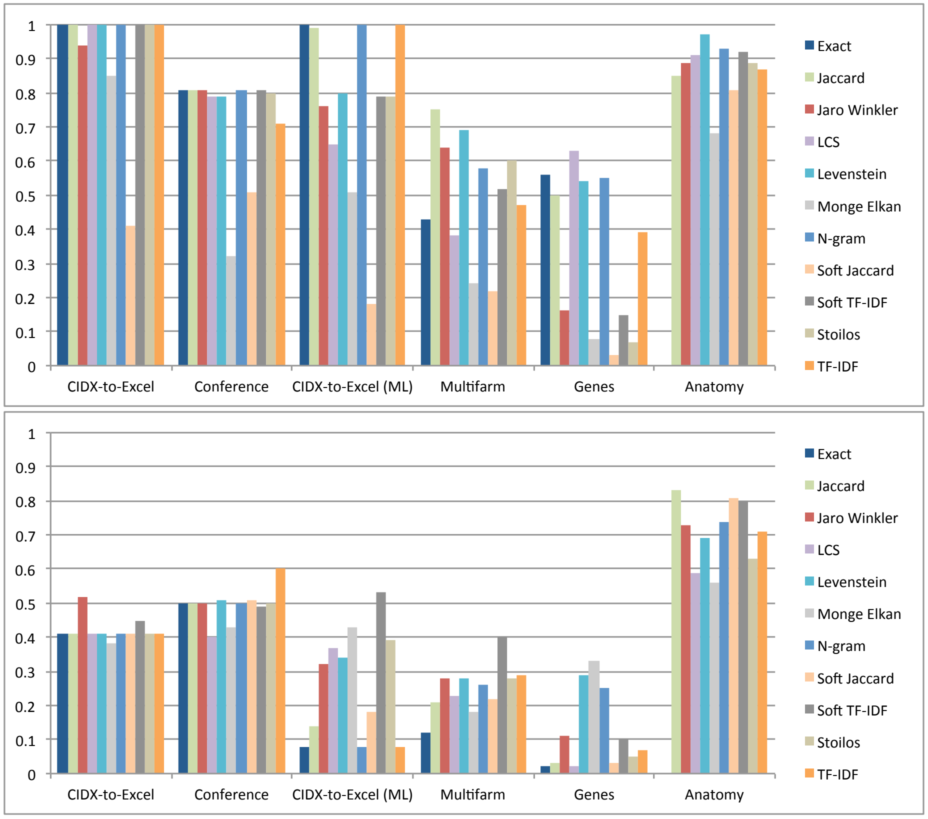
[3] http://disi.unitn.it/~accord/Experimentaldesign.html

[4] http://www.geneontology.org/GO.database.shtml

[5] http://genprotec.mbl.edu/files/MultiFun.html

[6] http://www.geneontology.org/GO.indices.shtml

Fig. 1. String metric precision (top) and recall (bottom) using the best-performing pre-processing strategies



4 Results

String Metrics and Pre-processing Strategies Figure 1 shows the performance of each string similarity metric with respect to both precision and recall. The best-performing string pre-processing strategy was used for each test set when collecting this data.

The conference and CIDX-to-Excel datasets do not reveal much disparity among the string similarity metrics. Also, the optimal thresholds indicate that the best approach is to look for matches that are as exact as possible.

The multifarm test set is much more challenging than the conference domain in terms of both precision and recall. The multilingual version of the CIDX-to-Excel dataset is less troublesome, most likely because the English-only version of that test set is also straightforward. Both multilingual test sets reveal a much wider disparity among string similarity metrics than the English-only versions, making the selection of an appropriate string similarity metric more important.

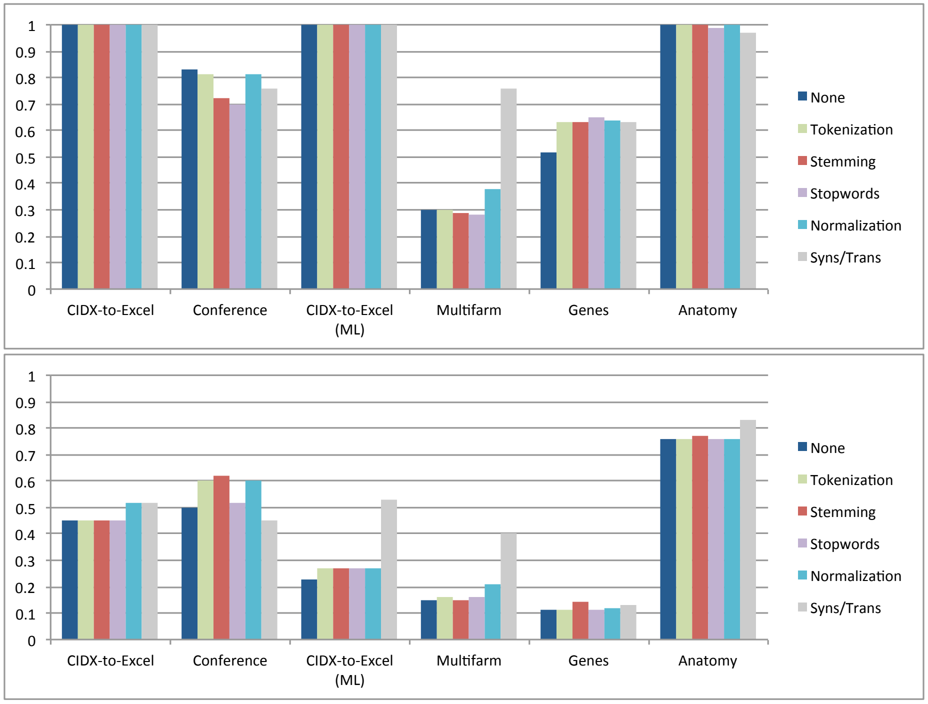
The difference between the best- and worst-performing metrics is even larger for the biomedical test sets. Recall is also significantly higher for the anatomy test set than any of the others. This is expected because biomedical datasets usually deal with a smaller, more regular vocabulary. There is often a small set of nouns with associated modifiers. The effect is not as pronounced for the Genes dataset. These test sets are also interesting in that there is a more clear choice to be made between metrics that have good precision verus good recall.

Another area of interest is the correlation of the performance on the analogous datasets. Variations in the absolute heights of the bars between analogous data sets are to be expected because the overall difficulty of matching a particular ontology pair may vary considerably – what we are looking for is the same general shape of the bars for the adjacent sets. A word of caution is in order here: the OAEI datasets have become the de facto standard for ontology alignment evaluation for a reason. It is very difficult to find quality datasets and reference alignments of the same scale elsewhere. In particular, the CIDX-to-Excel dataset is a single pair of schemas, while the conference set is made up of 16 ontologies. Also, the Genes test set is matching two biomedical schemas that have much less conceptual overlap than the Anatomy ontologies, so the number of correct matches is much smaller. The result in both cases is that smaller variations in performance are magnified for these analogous cases. Despite these limitations, we can see from the results that choosing a string similarity metric is less important for "standard" ontologies (i.e. English ontologies covering non-technical domains) because performance varies little among metrics. This is not the case for the multilingual and biomedical ontologies. In addition, we see that choosing a string similarity metric based on its performance on the OAEI test sets leads to good relative performance on analogous ontology matching problems.

Next we consider the effect of the different string pre-processing strategies on precision and recall for all of the test sets when the best-performing metric is used. Figure 2 shows this information. For the synonyms/translations set, synonyms were used for the test sets in which both ontologies were in English, while translations were used for the multilingual test sets. For a more detailed description of the results of each pre-processing strategy on every metric, please see the technical report [6].

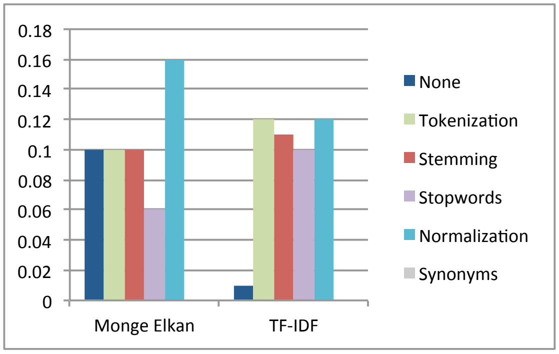
In general pre-processing strategies do not have a strong impact on performance. The most notable exception is translation, which drastically improves both precision and recall on test sets involving different languages. Normalization helps in these cases as well, albeit to a lesser extent. This is primarily due to transliteration of languages involving a non-Latin alphabet, such as Russian. In addition, tokenization is somewhat valuable, particularly if the ontologies to be matched use different naming conventions, such as underscores versus camelCase to delineate words. Finally, considering synonyms aids recall by a small but noticeable amount for biomedical ontologies. For the most part the pre-processing strategies exhibit similar behavior on the analogous data sets.

Fig. 2. Impact of pre-processing strategies on precision (top) and recall (bottom) using the best-performing metric



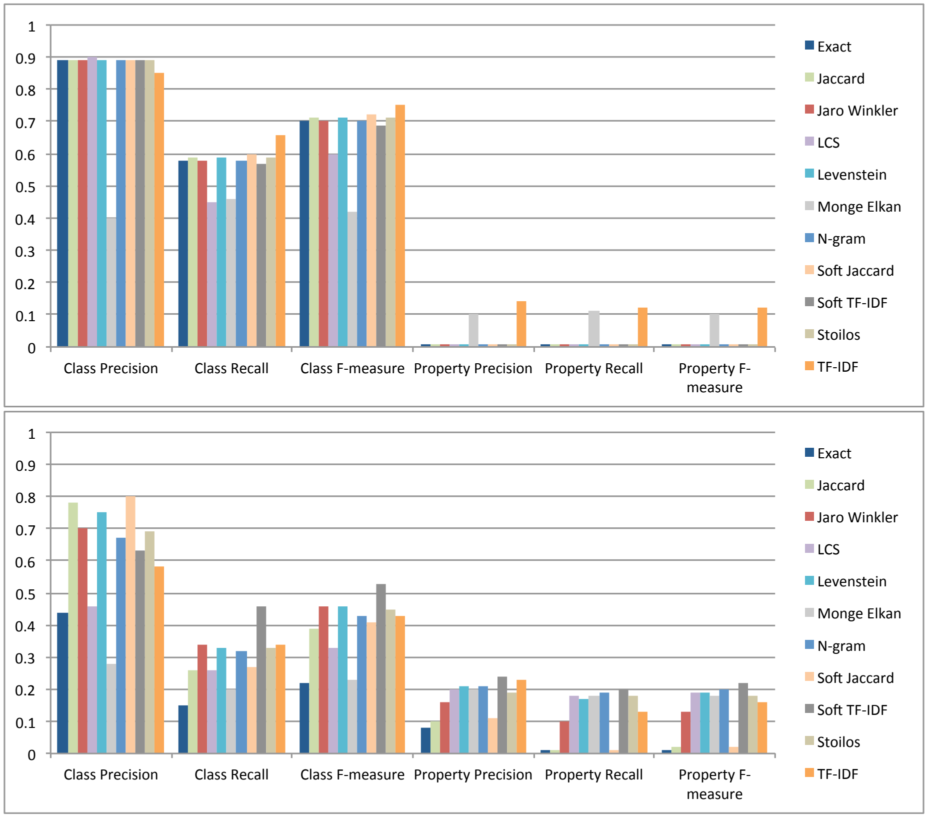
Classes vs Properties Others have found that human experts have a more difficult time agreeing on when properties match than on classes [13]. We seek here to determine if string similarity metrics also have particular difficulty with properties by looking at the performance of the metrics on classes versus properties for the Conference and Multifarm data sets. There are no matching properties in the other test sets.

Fig. 4. F-measures of Monge Elkan and TF-IDF on properties in the conference dataset for all of the string pre-processing strategies.



While this experiment should be performed on a wider variety of data sets for confirmation, the results shown in figure 3 support the theory that string similarity metrics perform much worse on properties than on classes. This suggests that more work should be done in this area in the future. It ap-

Fig. 3. F-measures of all metrics on the classes and properties in the conference dataset using string tokenization (top) and in the multifarm dataset using string tokenization and translation (bottom).



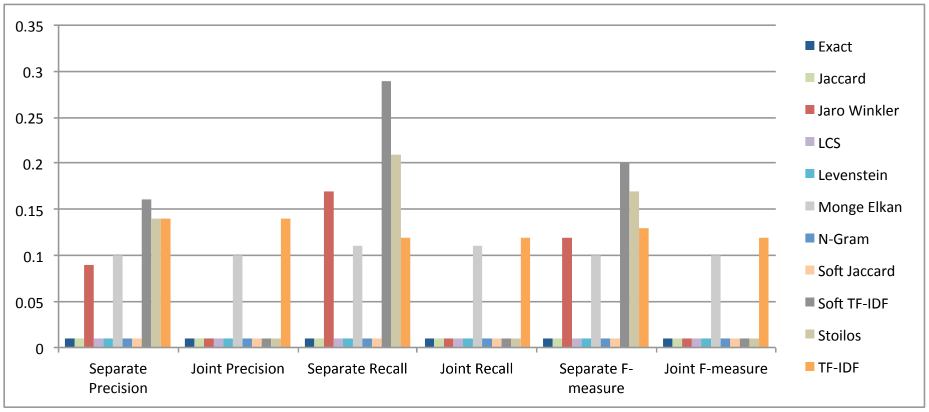
pears from an empirical analysis of our results that properties are particularly challenging for ontology alignment systems for several reasons. Properties frequently involve verbs, which can appear in a wider variety of forms than nouns (i.e. in addition to plurality/conjugation, verbs vary by tense). There are also often more functional words, such as articles and prepositions, in property names. Also, there are generally more common synonyms available for the (often very generic) verbs in property names than the (often more specific) nouns in class names. We therefore thought that stemming, stop word removal, or synonym lookup might be effective when matching properties. However, that turned out not to be the case. Figure 4 shows the effect of various pre-processing strategies in combination with the two metrics that performed the best on properties for the conference test set: Monge Elkan and TF-IDF. Tokenization is required for the TF-IDF metric to work because it is a global set metric. Normalization improved the performance of Monge Elkan somewhat. It seems that putting the words into alphabetical order reduces the number of gap penalties for matching properties in Monge Elkan. This had no effect for TF-IDF because set metrics are not sensitive to word order.

It is curious that properties are more easily matched on the mulitfarm data set. This data set consists of exactly the same ontologies as the conference set, just translated into a variety of languages. More analysis, possibly with the help of native speakers, is needed to determine the cause.

The above results were collected using the best thresholds found by optimizing the f-measure on the overall alignment problem (both classes and properties). In addition, we wanted to determine whether it was helpful to choose different thresholds for classes and properties. Figure 5 shows the best results achieved for property matching on the conference dataset when the thresholds were optimized based solely on the f-measure for properties. The precision, recall, and f-measure when the thresholds were optimized for overall f-measure are reproduced here for ease of comparison. The results for Monge Elkan and TF-IDF remain the same with either approach, but the results for several other metrics are improved significantly, indicating that for this dataset there is value in selecting different similarity metric thresholds for class and property comparisons.

Recommendations The results of the different metrics on the test sets reveal a potential trap for developers of ontology alignment systems. Results on the conference test set, which is representative of many real-world ontologies, do not show much difference in the performance of the metrics in terms of f-measure. However, the other test sets reveal that all string metrics are not created equal – performance of different metrics on the multi-lingual and biomedical test sets varied considerably. Choosing a string metric for use on these alignment tasks involves a significant impact on precision and recall. The moral of the story is that when choosing a string metric for use in an ontology alignment algorithm, one

Fig. 5. F-measures of all metrics using tokenization on the conference dataset when the thresholds were optimized once for classes and properties together (joint) versus separately for classes and properties (separate).



should consider the characteristics of the ontologies being aligned and whether precision or recall is more important for the algorithm. Below are some general guidelines which are suggested by our expriments:

- Standard ontology (English, non-technical)
    - Precision: All but Monge Elkan
    - Recall: TF-IDF
    - F-measure: All but Monge Elkan and LCS
- Multilingual
    - Precision: Soft Jaccard, Jaccard
    - Recall: Soft TF-IDF
    - F-measure: Soft TF-IDF
- Biomedical
    - Precision: Levenstein
    - Recall: Jaccard, Soft Jaccard, Soft TF-IDF
    - F-measure: Soft TF-IDF, Jaccard, Soft Jaccard

Of the pre-processing strategies analyzed, few were beneficial. Tokenization is useful if the naming conventions differ between the ontologies (camelCase versus underscores to separate words, for example). Translation is very helpful when ontologies involve multiple languages. If translation is not available, normalization can be useful for multilingual ontology pairs, particularly if one of the languages uses a non-Latin alphabet and can be transliterated. Synonyms can be useful (particularly with respect to recall) for biomedical ontologies, where the synonyms are often embedded in the ontologies themselves.

Class labels are significantly easier for string metrics to match than are property labels. Performance can be improved by using different thresholds for classes and properties. It would be helpful to look into this further by examining what enables some metrics to do better than others and potentially develop a new metric that emphasizes these strengths further regarding property labels.

String-centric Ontology Alignment

We now turn to the question of how much we can accomplish using only string metrics. To answer this we first align the ontologies using the optimal string metrics and pre-processing strategies for each test set. The algorithm works in the same way as our test framework – comparing every label in the first ontology to every label in the second and using the stable marriage algorithm to find the best mappings. The difference is that here we run the algorithm repeatedly: first with a high-precision metric and then with a high-recall metric. Because string metrics were found to perform extremely poorly on properties, this approach does not attempt to match those (i.e. any property matches in the reference alignment are automatically false negatives). For this proof-of-concept, the algorithm is hardcoded with the optimal metrics and thresholds for the particular test set under consideration.

Table 1. Results of strings only approaches and the competitors from the OAEI 2012 competition on the conference data set (left) and the anatomy data set (right)

Metric	Prec.	Recall	F-meas.
YAM++	0.81	0.69	0.75
LogMap	0.82	0.58	0.68
StringsOpt	**0.85**	**0.55**	**0.67**
StringsAuto	**0.79**	**0.57**	**0.66**
Optima	0.62	0.68	0.65
CODI	0.74	0.57	0.64
GOMMA	0.85	0.47	0.61
Wmatch	0.74	0.50	0.60
WeSeE	0.76	0.49	0.60
Hertuda	0.74	0.50	0.60
MaasMatch	0.63	0.57	0.60
LogMapLt	0.73	0.50	0.59
HotMatch	0.71	0.51	0.59
Baseline 2	0.79	0.47	0.59
ServOMap	0.73	0.46	0.56
Baseline 1	0.80	0.43	0.56
ServOMapLt	0.88	0.40	0.55
MEDLEY	0.54	0.50	0.52
ASE	0.63	0.43	0.51
MapSSS	0.50	0.51	0.50
AUTOMSv2	0.67	0.36	0.47
AROMA	0.33	0.48	0.39

Metric	Prec.	Recall	F-meas.
GOMMA-bk	0.92	0.93	0.92
YAM++	0.94	0.86	0.90
CODI	0.97	0.83	0.89
StringsOpt	**0.88**	**0.87**	**0.88**
LogMap	0.92	0.85	0.88
GOMMA	0.96	0.80	0.87
StringsAuto	**0.86**	**0.84**	**0.85**
MapSSS	0.94	0.75	0.83
WeSeE	0.91	0.76	0.83
LogMapLt	0.96	0.73	0.83
TOAST*	0.85	0.76	0.80
ServOMap	1.00	0.64	0.78
ServOMapLt	0.99	0.64	0.78
HotMatch	0.98	0.64	0.77
AROMA	0.87	0.69	0.77
StringEquiv	1.00	0.62	0.77
Wmatch	0.86	0.68	0.76
Optima	0.85	0.58	0.69
Hertuda	0.69	0.67	0.68
MaasMatch++	0.43	0.78	0.56

The results are shown in tables 1 and 2 under the heading StringsOpt, along with the results of the OAEI 2012 competitors [1]. Of particular note are Baseline 1 and Baseline 2, which are unrefined string similarity approaches. Baseline 1 uses string equality and Baseline 2 is the same but with dashes, underscores and the word “has” removed from strings prior to comparison.

It is evident that StringsOpt compares very well with state-of-the-art ontology alignment systems, but *this is not an apples-to-apples comparison* because it is not generic (due to the hard-coded metrics based on the test set). The next step is to add some means of selecting the appropriate string metrics and thresholds at runtime, with the goal of developing a method that is fully autonomous and does not rely on any training data. As a first attempt, we have developed an analysis module that runs before our main alignment algorithm to select the string metrics. This analysis module examines an ontology to find the answers to three simple questions: Is the ontology in English? What is the average number of words per entity label (after tokenization)? Does the ontology contain embedded synonyms?

Based on the results of the analysis module and whether precision or recall is currently of interest in the alignment process, a string metric and thresholds are chosen. This is currently done using the hard-coded set of rules shown below, but more research remains to be done in this area. Note that these rules do not break cleanly among the different test sets – they are based on underlying features of the ontologies to be matched.

- Precision
  - Less than two words per label: **Jaro-Winkler 1, 1**
  - Two or more words per label
    - Synonyms: **Soft Jaccard .2, .5 with Levenstein .9 base metric**
    - No synonyms: **Soft Jaccard 1, 1 with Levenstein .8 base metric**
- Recall
  - Less than two words per label: **TF-IDF .8, .8**
  - Two or more words per label
    - Synonyms: **Soft TF-IDF .5, .8 with Jaro-Winkler .8 base metric**
    - Different Languages: **Soft TF-IDF 0, .7 with Jaro-Winkler .9 base metric**
    - Other: **Soft TF-IDF .8, .8 with Jaro-Winkler .8 base metric**

We have added this automatic metric selection step to our approach. The results for this are shown in tables 1 and 2 under StringsAuto. We have also added it to MapSSS, an existing ontology alignment system [5]. The results for the version of MapSSS using these optimized metrics and string pre-processing strategies are compared with the results of the original system in Table 3. The deeper insight into string labels has significantly improved the performance of MapSSS on the conference test set and marginally improved it on the anatomy test set. The extremely large gains on the multifarm test set are due to the inclusion of translation as a string pre-processing strategy.

5 Conclusions and Future Work

For some types of ontologies, the performance of different string similarity metrics varies greatly in terms of both precision and recall. It is important to be

Table 2. Results of the strings only approaches together with the competitors from the OAEI 2012 competition on the multifarm data set ("same" are alignments of the same ontologies in different languages and "diff" are alignments of different ontologies in different languages)

Metric	Prec (diff)	F-ms (diff)	Rec (diff)	Prec (same)	F-ms (same)	Rec (same)
AUTOMSv2	0.49	0.17	0.10	0.69	0.11	0.06
GOMMA	0.29	0.32	0.36	0.63	0.40	0.29
MEDLEY	0.16	0.10	0.07	0.34	0.14	0.09
WeSeE	0.61	0.42	0.32	0.90	0.42	0.27
Wmatch	0.22	0.22	0.22	0.43	0.18	0.11
YAM++	0.50	0.42	0.36	0.91	0.64	0.49
StringsOpt	**0.58**	**0.40**	**0.31**	**0.90**	**0.38**	**0.24**
StringsAuto	**0.64**	**0.39**	**0.28**	**0.93**	**0.26**	**0.15**

Table 3. Results of the original and improved MapSSS alignment algorithm on the OAEI 2012 data sets

Test Set	Measure	Original	Improved	Improvement	OAEI 2012 Placement
Conference	Precision	0.50	0.73	46%	Tied 11th
	Recall	0.51	0.57	12%	Tied 4th
	F-measure	0.50	0.64	28%	Tied 4th
Anatomy	Precision	0.94	0.86	-8%	Tied 14th
	Recall	0.75	0.84	12%	4th
	F-measure	0.83	0.85	2%	6th
Multifarm	Precision diff	0.08	0.45	463%	4th
	Recall diff	0.04	0.28	600%	4th
	F-measure diff	0.05	0.35	547%	3rd
	Precision same	0.97	0.96	-1%	1st
	Recall same	0.50	0.25	-50%	4th
	F-measure same	0.66	0.40	-40%	Tied 3rd

cognizant of this when selecting a string metric for a particular use. This paper has established guidelines to assist researchers in making this selection. In addition, we have found that many string pre-processing strategies commonly used, such as stop word removal and word stemming, are in many cases unhelpful and in some cases counter-productive. We have presented data on which pre-processing strategies are useful in particular situations. In addition, we have developed a basic system to automatically select an appropriate string similarity metric for a given pair of ontologies at runtime. Finally, we have applied this technique to an existing ontology alignment algorithm and quantified the improvement in performance.

There are several paths for future work based on the idea of pushing string similarity metrics as far as they can go in terms of ontology alignment. A first step is to develop a string similarity metric that performs better on properties. Another possibility is to create a string-based structural metric by considering the similarity between the labels of all of an entity's neighbors with those of another entity. This idea is similar to current "semantic flooding" approaches in that it takes advantage of the principle of locality, but is entirely lexically based. For biomedical ontologies, it might also be possible to use string metrics to find subsumption relations in addition to equivalences since many labels in these ontologies are of the form noun and modifiers+noun (e.g. vein and pulmonary vein). In terms of the work presented in this paper, the results should be validated for more pairs of ontologies. Also, the analysis module for metric selection can be made more flexible.

Acknowledgements. This work was supported by the National Science Foundation under award 1017225 "III: Small: TROn—Tractable Reasoning with Ontologies."

References

1. Aguirre, J.L., Grau, B.C., Eckert, K., Euzenat, J., Ferrara, A., van Hague, R.W., Hollink, L., Jimenez-Ruiz, E., Meilicke, C., Nikolov, A., et al.: Results of the ontology alignment evaluation initiative 2012. In: Proc. 7th ISWC Workshop on Ontology Matching (OM). pp. 73–115 (2012)
2. Ashburner, M., et al.: Gene Ontology: tool for the unification of biology. Nature Genetics 25(1), 25–29 (2000)
3. Bethea, W.L., Fink, C.R., Beecher-Deighan, J.S.: JHU/APL Onto-Mapology results for OAEI 2006. In: Proc. ISWC Workshop on Ontology Matching. pp. 144–152 (2006)
4. Branting, L.K.: A comparative evaluation of name-matching algorithms. In: Proceedings of the 9th International Conference on Artificial Intelligence and Law. pp. 224–232. ACM (2003)
5. Cheatham, M.: MapSSS results for OAEI 2011. In: Proceedings of the ISWC 2011 Workshop on Ontology Matching. pp. 184–190 (2011)
6. Cheatham, M., Hitzler, P.: The role of string similarity metrics in ontology alignment. Tech. rep., Kno.e.sis Center, Wright State University (2013), available from http://www.pascal-hitzler.de/pub/smet13.pdf
7. Cohen, W.W., Ravikumar, P., Fienberg, S.E., et al.: A comparison of string distance metrics for name-matching tasks. In: Proceedings of the IJCAI-2003 Workshop on Information Integration on the Web (IIWeb-03). vol. 47 (2003)
8. Curino, C., Orsi, G., Tanca, L.: X-SOM results for OAEI 2007. In: Proceedings of the Second International Workshop on Ontology Matching. pp. 276–285 (2007)
9. Euzenat, J., Shvaiko, P.: Ontology matching, vol. 18. Springer Heidelberg (2007)
10. Euzenat, J., et al.: State of the art on ontology alignment. Knowledge Web Deliverable D 2, 2–3 (2004)
11. Lambrix, P., Tan, H., Liu, Q.: SAMBO and SAMBOdtf results for the ontology alignment evaluation initiative 2008. In: Proceedings of the Third International Workshop on Ontology Matching. pp. 190–198 (2008)
12. Lin, D.: An information-theoretic definition of similarity. In: Proceedings of the 15th International Conference on Machine Learning. vol. 1, pp. 296–304. San Francisco (1998)
13. Maedche, A., Staab, S.: Measuring similarity between ontologies. In: Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, pp. 251–263. Springer (2002)
14. Monge, A.E., Elkan, C.: The field matching problem: Algorithms and applications. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. pp. 267–270 (1996)
15. Ngo, D., Bellahsene, Z., Todorov, K.: Opening the black box of ontology matching. In: The Semantic Web: Semantics and Big Data, pp. 16–30. Springer (2013)
16. Stoilos, G., Stamou, G., Kollias, S.: A string metric for ontology alignment. In: The Semantic Web–ISWC 2005, pp. 624–637. Springer (2005)
17. Valarakos, A.G., Paliouras, G., Karkaletsis, V., Vouros, G.: A name-matching algorithm for supporting ontology enrichment. In: Methods and Applications of Artificial Intelligence, pp. 381–389. Springer (2004)