Results of OKKAM Feature Based Entity Matching Algorithm for Instance Matching Contest of OAEI 2009

Heiko Stoermer, Nataliya Rassadko

name.surname-at-unitn.it
The University of Trento
via Sommarive, 14 Povo 38123 Italy

Abstract. To investigate the problem of entity recognition, we deal with the creation of the so-called Entity Name System (ENS) which is an open, public back-bone infrastructure for the (Semantic) Web that enables the creation and systematic re-use of unique identifiers for entities. The ENS can be seen as a very large, distributed "phonebook for everything", and ENS identifiers might be considered as a "phone number" of entities. Entity descriptions are based on free-form key/value "tagging" rather than on some precise formalism. However, such a genericity has its shortcomings: the ENS can never know what type of entity it is dealing with. We tackle this problem in a novel approach for entity matching that is called Feature Based Entity Matching (FBEM). In the current paper, we report an evaluation of FBEM on datasets provided by the OAEI committee for the instance matching contest.

1 Presentation of the system

With the growth and development of Semantic Web, the latter became like a collection of "information islands" which are poorly integrated to each other. The problem of information integration in Semantic Web is two-fold:

1. heterogeneity of vocabulary: the same concept can be referred via different URIs, and therefore may be considered to be as different concepts in different vocabularies;
2. entity recognition: the same real word object can be referred via different URIs in different repositories, and therefore may not be recognized as the same object.

While the first issue is widely recognized and investigated [4], the second one was largely neglected, although it received a lot of attention under the heading of record linkage, data deduplication, entity resolution, etc [1].

To investigate the problem of entity recognition, EU-funded OKKAM project [1] deals with the creation of the so-called Entity Name System (ENS) [3].

[1] http://www.okkam.org

1.1 State, purpose, general statement

In this section, we introduce the ENS and describe our interest in instance matching part of OAEI 2009.

Entity Name System (ENS) [3] is an open, public back-bone infrastructure for the (Semantic) Web that enables the creation and systematic re-use of unique identifiers for entities. It is implemented as a large-scale infrastructural component with a set of services needed for describing entities, and assigning identifiers to them.

Figuratively, the ENS can be seen as a very large, distributed "phonebook for everything", and ENS identifiers might be considered as a "phone number" of entities. This leads to a more efficient information integration, and thus a real global knowledge space, without the need for ex-post deduplication or entity consolidation.

In the ENS, we do not impose or enforce the usage of any kind of schema or strong typing for the description of different types of entities. Instead, entity descriptions are free-form and are based on key/value "tagging". In such a way, we support a complete genericity, without the need for any formalism or any abstract top-level categorizations. Taking into account the aforementioned peculiarities of the ENS, our restriction to the instance matching part of OAEI 2009 becomes evident.

Obviously, our model of such a generic entity description has its shortcomings: the ENS can never know what type of entity it is dealing with, and how the entity is described, due to an absence of a formal model. This becomes very relevant when searching for an entity, a process that we call entity matching. To address this problem, we rely on recent work [2] that has been performed with the goal to find out in an experimental setting how people actually describe (identify) entities. Based on these findings, we propose a novel approach for entity matching.

The approach takes into account not only the similarity of entity features (keys and values), but also the circumstance that certain features are more meaningful for identifying an entity than others. We call this approach as Feature Based Entity Model (FBEM) and we explain it in the next section.

1.2 Specific techniques used

We consider both a reference (matching) entity Q and candidate (matched) entity E as a set F of *features* f:

$$F = \{f\}; f =< n, v >;$$

where each feature f is a pair of name n and value v. We do not require neither name nor value to share a vocabulary or schema, or even a natural language, i.e., they are independent in content and size.

We enumerate all features of any particular entity with integer values and denote as f_i^Q, f_j^E the ith and jth features of entities Q and E respectively.
We define the following functions:

$n(f_i)$**:** returns the *name* part of a feature of an entity;
$v(f_i)$**:** returns the *value* part.

Now, we define $f_{i,j}sim(f_Q, f_E)$, a function that computes the similarity of two features f_i^Q, f_j^E as follows:

$$f_{i,j}sim(f_Q, f_E) =_{def}$$
$$sim \quad f_i^Q, f_j^E \quad * \begin{cases} 2 * \lambda * \mu, & \text{for } name(n(f_i^Q)), name(n(f_j^E)), id(f_i^Q, f_j^E); \\ 2 * \mu, & \text{for } name(n(f_i^Q)), name(n(f_j^E)); \\ \lambda * \mu, & \text{for } name(n(f_j^E)), id(f_i^Q, f_j^E); \\ \mu, & \text{for } name(n(f_j^E)); \\ 1, & \text{otherwise .} \end{cases} \tag{1}$$

Equation 1 relies on the following functions and parameters:

$sim(x, y)$: a suitable string similarity measure between x and y.
$name(x)$: a boolean function indicating whether the feature x denotes one of the possible names of the entity;
$id(x, y)$: the identity function, returning true if value parts of x and y are identical;
μ : the factor to which a name feature is considered more important than a non-name feature;
λ : the extra-factor attributed to the the presence of the value identity $id(x, y)$.

In our implementation, we selected Levenstein metric as a similarity measure (sim-function), and both λ and μ equal to 2. The latter can be interpreted as "the occurrence of a fact is as twice as important than its absence".

We have also implemented a vocabulary, small enough to be maintained in a runtime memory, that is used to detect the cases where entity feature name is actually a "name" of the entity, e.g., "name", "label", "title", "denomination", "moniker".

At this point, we are able to establish the similarity between individual features. To compute the complete feature-based entity similarity, which finally expresses to which extend E is similar to Q, we proceed as follows.

Let $maxv(V)$ be a function that computes the maximum value in a vector[2]. We then span the matrix M of feature similarities between Q and E, defined as

$$M := (fsim\,(Q, E))_{|Q|\times|E|} \rightarrow \mathbb{Q} \geq 0$$

with $fsim$ as defined above, and $|Q|, |E|$ being the number of elements of the vectors Q and E, respectively.

The feature-based entity similarity score fs is defined as the sum of all the *maximum similar* feature combinations between Q and E:

$$fs\,(Q, E) = \sum_{i=1}^{|Q|} maxv(M_i) \tag{2}$$

[2] Trivially defined as $maxv\,(V) = max_{i=1}^{|V|}\,(V_i)$, with $|V|$ being the number of elements of V.

So far, we provided a method to calculate fs-similarity that may belong to a wide range of values from zero to infinity [5]. This complicates an evaluation of actual similarity of entities. For example, if $fs = 7$ it might stand for identical entities in one dataset and completely different entities in the other one.

To resolve this problem, we normalize fs values as follows. Taking into account that M_i is a weighted value, we use a dot-notation to denote its weight w as $M_i.w$. Then the final formula of *normalized* similarity has the following form:

$$esim(Q, E) = \frac{fs(Q, E)}{\sum_{i=1}^{|Q|} maxv(M_i).w} \quad (3)$$

In the last formula, we simply divided a sum of weighted values on a sum of corresponding weights. This allows us to normalize similarity score within the range of $[sim(x, y)_{min}, sim(x, y)_{max}]$, e.g., $[0, 1]$ if similarity metric return the values in this range, which is true for Levenstein similarity.

1.3 Adaptations made for the evaluation

We parsed all provided rdf-files into a Jena-model [3] stored as a persistent SDB [4] with an underlying MySQL database [5]. To adapt our FBEM-model to the required output in the alignment format [6], we wrote a simple iterator over SDB-instances related to reference entities Q and to candidate entities E, i.e., we matched each Q against each E, where both Q and E were preliminarily converted to the ENS entity format.

For the reason of a better time-performance, we implemented a "typed" matching, i.e., Q and E should have been of the same entity type (e.g., people were matched against people, documents against documents). The types were easy to extract from the attribute "type" available in most benchmarks. We also implemented a "brute-force" matching, i.e. any-to-any, which did not consider any type features, to match those benchmarks where typing was not provided or was difficult to reason.

For each Q, we maintained a vector of E ranked w.r.t. a similarity value $esim(Q, E)$. The length of vector was limited to 50 elements due to time- and memory- performance reasons.

In the alignment file, we output only those elements of vector of Es that had a similarity value greater than or equal to a certain threshold. The threshold was selected empirically for each particular benchmark. More precisely, we run experiments for thresholds from the set $\{0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95\}$ and then selected that thresholds that gave us the most acceptable values of precision/recall from the viewpoint of the ENS methodology. Namely, we were eager to maintain as high precision as possible with any non-zero recall.

The reason for selecting precision of the ENS performance was the following: we assume that the ENS user, while querying the ENS repository, expects few answers in

[3] http://jena.sourceforge.net/

[4] http://jena.sourceforge.net/SDB/

[5] http://mysql.com

[6] http://alignapi.gforge.inria.fr/format.html

the result set. However, these answers should be the most relevant to the user query. In other words, for the ENS it's better to answer with some highly precise entities rather than with a lot of somehow likely similar entities.

Precise threshold values we used to run FBEM-matching over each particular benchmark will be indicated in Sec. 2.

1.4 Link to the system and parameters file

`http://www.dit.unitn.it/~rassadko/OAEI2009/okkamsystem.zip`

1.5 Link to the set of provided alignments (in align format)

`http://www.dit.unitn.it/~rassadko/OAEI2009/okkamalignment.zip`

2 Results

Due to peculiarities of the ENS described in Sec. 1.1, we have restricted ourselves only to instance matching benchmarks.

2.1 A-R-S

The benchmark contains includes three datasets describimg instances from the domain of scientific publications:

- eprints - this dataset contains papers produced within the AKT research project and extracted using an HTML-wrapper from the source web-site;
- rexa - this dataset was extracted from the search results of the search server;
- SWETO-DBLP - a version of the DBLP dataset.

For A-R-S benchmark we applied a "typed" version (see Sec. 1.3) of FBEM-matching because all three datasets contained information about authors (typed with *foaf* namespace [7]) and their scientific publication (typed with *opus* namespace [8]).

We run our experiment with threshold 0.80. The result of our experiments are presented in Table 1.

In Sec. 1.3, we explained that we are interested in high precision with any non-zero recall. As Table 1 shows, we gained our objective. With a less tight threshold, it is possible to slightly sacrifice a precision for a better recall.

2.2 T-S-D

For this dataset we do not have results. First of all, typing of each particular data source was different from the others. This required reasoning over ontologies which were provided with datasets. Since our system does not support any kind of ontology reasoning, one might have made an attempt to run a "brute-force" matching, i.e., any-to-any. Unfortunately, due to a large size of data, we were unable to finish the match run timely.

[7] `http://xmlns.com/foaf/0.1/`
[8] `http://lsdis.cs.uga.edu/projects/semdis/opus`

Table 1. A-R-S results

Test	Precision	Recall	F-measure	Fallout
eprints-rexa	0.94	0.10	0.18	0.06
eprints-dblp	0.98	0.16	0.28	0.02
rexa-dblp	1.00	0.12	0.22	0.00

2.3 IIMB

IIMB benchmark is generated from a dataset provided by OKKAM. We run our experiment with threshold 0.95. Our results are shown in Table 2.

Table 2. IIMB results

Test	001	002	003	004	005	006	007	008	009	010
Precision	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.95	0.92
Recall	1.00	1.00	1.00	1.00	0.99	0.98	0.98	0.96	0.85	0.52
F-measure	0.98	0.98	0.98	0.98	0.97	0.97	0.97	0.96	0.90	0.66
Test	011	012	013	014	015	016	017	018	019	
Precision	0.88	0.94	0.92	0.00	0.91	0.86	0.72	0.82	0.71	
Recall	0.43	0.98	0.71	0.00	0.96	0.74	0.30	0.38	0.15	
F-measure	0.58	0.96	0.80	NaN	0.93	0.79	0.43	0.52	0.25	
Test	020	021	022	023	024	025	026	027	028	029
Precision	0.78	0.47	0.15	0.08	0.09	0.05	0.09	0.05	0.89	0.00
Recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	NaN
F-measure	0.88	0.64	0.25	0.15	0.17	0.10	0.16	0.10	0.94	NaN
Test	030	031	032	033	034	035	036	037		
Precision	0.28	0.80	0.09	0.08	0.10	0.00	0.11	0.00		
Recall	0.04	0.25	1.00	0.99	1.00	0.00	0.98	0.00		
F-measure	0.06	0.38	0.16	0.15	0.19	NaN	0.19	NaN		

Below, we provide our comments to the results presented in Table 2:

001 Contains an identical copy of the original ABox with the instance IDs randomly changed. And for this test, we performed well with pretty high precision.

002-010 Value transformations (i.e., typographical errors simulation). ENS user is not assumed to enter extremely misspelled queries. Therefore, we may conclude that our performance is appropriate. Although the recall dropped down at experiment 010, ENS user would still received highly relevant result set.

011-019 Structural transformations (i.e., deletion of one or more values, transformation of datatype properties into object properties, separation of a single property into more properties). From ENS viewpoint it might be seen as if the user query contained permutated feature names and feature values. For these test cases, we

have medium performance: with the precision around 0.70-0.90, the recall varies from 0.15 to 0.98. We believe, that these results are still acceptable for the ENS user.

020-029 Logical transformations (i.e., instantiation of identical individuals into different subclasses of the same class, instantiation of identical individuals into disjoint classes, instantiation of identical individuals into different classes of an explicitly declared class hierarchy). These cases are impossible for ENS because ENS does not have any schema or ontology. Yet having conducted a "brute-force" (non-typed) matching of each entity Q against each entity E, we could still provide the ENS user with some information.

030-037 Several combinations of the previous transformations. For these test cases, we have an uneven performance which is expected.

3 General comments

3.1 Comments on the results

We mainly commented our results in Sec. 2. In general, we believe that FBEM performs well for the purposes of the ENS. Namely, we are able to answer user queries with a high precision. And this is a strength of our approach. As the weakness, we have to admit that recall values are not so much satisfactory. And in the next section, we will discuss the ways to deal with this problem.

3.2 Discussions on the way to improve the proposed system

We need to experiment with other similarity metrics $sim(x, y)$ since Levenstein metrics deals badly with the permutated words, e.g., "Stephen Potter" and "Potter, Stephen". This can lead to a low recall as in our results for A-R-S benchmark.

Basic structural analysis is also planned to be introduced. For example, one entity Q may have attributes "first name" and "given name" while entity E can contain only "name" (i.e. both first and give name together). We believe that elements of structural analysis will help us improve both precision and recall for the cases like in tests 20-29 for IIMB benchmark.

We are currently working on a more extended version of FBEM-model which concentrates not only on names of entities, but also on other features that might identify entity. For example, a feature "isbn" uniquely identifies book, "e-mail" likely identifies a person etc. We will rely on the empirical study [2] which we mentioned above.

Finally, we did not expect the datasets larger than 1Gb. However, this forced us to include in our future research also a loaded bulk-matching, e.g., 1Gb dataset against 1Gb dataset.

3.3 Comments on the OAEI 2009 procedure

We are satisfied with the OAEI 2009 procedure.

3.4 Comments on the OAEI 2009 test cases

As we said above, the test cases turned to be unfeasible for our matching procedure.

3.5 Comments on the OAEI 2009 measures

We are satisfied with the OAEI 2009 measures.

3.6 Proposed new measures

No proposals.

4 Conclusion

In the current paper, we proposed an evaluation of a novel approach for entity matching that is called Feature Based Entity Matching (FBEM) over datasets provided by the OAEI committee for the instance matching contest.

Since FBEM could be a candidate to a set of matching modules of the ENS, we were eager to maintain as high precision as possible with any non-zero recall. In general, we gained our objective. Namely, we perform well in the cases where there is no need in ontology reasoning or structural analysis.

We are satisfied with our results. However, there are several directions (see Sec. 3.2) to improve the performance of FBEM from the viewpoint of both precision and recall values.

Acknowledgments. This paper has been supported by the FP7 EU Large-scale Integrating Project OKKAM "Enabling a Web of Entities" (contract no. ICT-215032). For more details, visit `http://fp7.okkam.org`.

References

1. Duplicate record detection: A survey. *IEEE Trans. on Knowl. and Data Eng.*, 19(1):1–16, 2007. Senior Member-Elmagarmid, Ahmed K. and Member-Ipeirotis, Panagiotis G. and Member-Verykios, Vassilios S.
2. B. Bazzanella, P. Bouquet, and H. Stoermer. A Cognitive Contribution to Entity Representation and Matching. Technical Report DISI-09-004, Ingegneria e Scienza dell'Informazione, University of Trento., 2009. `http://eprints.biblio.unitn.it/archive/00001540/`.
3. P. Bouquet, H. Stoermer, C. Niederee, and A. Mana. Entity Name System: The Backbone of an Open and Scalable Web of Data. In *Proceedings of the IEEE International Conference on Semantic Computing, ICSC 2008*, CSS-ICSC, pages 554–561. IEEE Computer Society, 2008.
4. J. Euzenat and P. Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.
5. H. Stoermer and P. Bouquet. A Novel Approach for Entity Linkage. In *Proceedings of IRI 2009, the 10th IEEE Internationational Conference on Information Reuse and Integration, August 10-12, 2009, Las Vegas, USA*, volume 10 of *IRI*, pages 151–156. IEEE Systems, Man and Cybernetics Society, August 2009.