Feature-Based Entity Matching: The FBEM Model, Implementation, Evaluation*

Heiko Stoermer, Nataliya Rassadko, and Nachiket Vaidya

University of Trento,
Dept. of Information and Communication Tech.,
Trento, Italy
{stoermer,rassadko,vaidya}@disi.unitn.it

Abstract. Entity matching or resolution is at the heart of many integration tasks in modern information systems. As with any core functionality, good quality of results is vital to ensure that upper-level tasks perform as desired. In this paper we introduce the FBEM algorithm and illustrate its usefulness for general-purpose use cases. We analyze its result quality with a range of experiments on heterogeneous data sources, and show that the approach provides good results for entities of different types, such as persons, organizations or publications, while posing minimal requirements to input data formats and requiring no training.

Keywords: Entity resolution, record linkage, information integration.

1 Introduction

The question whether two pieces of structured or semi-structured data refer to the same object (or entity) has been the objective of substantial work over several decades, and good solutions are the prerequisites for high-quality automatic integration or linkage of data, information and knowledge. This integration topic and the related quality issues are receiving increased attention by work performed in the context of the Semantic Web, or more generally, the linkage of entity-related information in more general-purpose web information systems.

The problem, which appears in literature under names such as record linkage, entity linkage, entity resolution, etc., poses substantial challenges to automated approaches, particularly when faced with a heterogeneous environment. In open, de-centralized environments that are emerging on the Web, we can find an increasing amount of (semi-) structured information. But especially this lack of centralization – which is one of the reasons for the dynamic development of the web and thus one of its strengths – leads to difficult situations where for example creators of structured information published on the web refer to the entities they describe with arbitrary, self-issued identifiers, making it impossible for an automated system to perform efficient, syntactical integration of information from different sources [4].

* The authors would like to thank Barbara Bazzanella for her excellent work on [2], and for the permission to reproduce Table 1. This work is partially supported by the FP7 EU Large-scale Integrating Project **OKKAM – Enabling a Web of Entities** (contract no. ICT-215032). For more details, please visit http://www.okkam.org

B. Pernici (Ed.): CAiSE 2010, LNCS 6051, pp. 180–193, 2010.

Nonetheless, the demand for high-quality integration comes naturally from the side of users and producers of progressive information systems alike. The cost-efficient creation of a news and article dossier about a certain person (which can then be sold against a micropayment), the linkage of contacts between social networks, the collection of opinions about a certain product from different sources, and many more examples all require good-quality, easy-to-use automated approaches for entity resolution.

In this paper we present a novel approach for entity resolution, named FBEM (feature-based entity matching). The approach is a generalization of earlier work[17], and combines probabilistic as well as ontological methods for deciding whether two records describe the same entity, taking into account intensional and extensional aspects of the entities at hand. The approach aims at general-purpose usefulness with special focus on web information systems, and bases on empirical findings about what are commonly used entity types, and how they are usually described.

The rest of this article is organized as follows: the following section makes an attempt at giving a concise account of the vast amount of related work from several fields. Section 3 explains the background knowledge underlying the FBEM approach and its representation in an ontology which serves as input to the algorithm. Sect. 4 then introduces the algorithm, proposing a formal model for entity similarity and describing its implementation. In Sect. 5 we perform a thorough investigation of the usefulness of the algorithm, presenting a series of experiments in entity resolution which cover entity types such as organizations, persons and scientific publications in datasets of different sizes. The article closes with a discussion of the results and a description of further work that is planned for improving the approach.

2 Related Work

In contrast to schema-level integration, entity-level integration deals with the actual individuals, not with integration of class structures or entity types. Entity level integration has to deal with deciding whether two entity descriptions refer to the same individual. Approaches to the problem of entity resolution can be broadly categorized into two main categories: one requires training data to adapt the matching procedure with machine learning techniques, the other depends on domain knowledge for matching. The approach presented in this paper falls in the latter category.

Approaches of entity-level integration have been proposed under several names, ranging from duplicate detection [9], entity resolution [3, 11], merge/purge [12], object identifcation [18], reference reconciliation [8]. Extensive surveys are available [9, 5, 19]. A related group of algorithms are the ones that aim at matching entity names by computing the distance between the string values of corresponding entity names. The algorithms included in this group suggest general-purpose methods for computing the similarity between strings [14]. These algorithms are considered important since they are currently used as the basic metric on which more sophisticated approaches are based on. [7] describes and provides an experimental comparison of various string distance metrics. Other approaches involve schema matching [16] in cases where the entities to be matched are described with a different schema, also employing domain knowledge from ontologies, where available [15].

3 Knowledge about Entities

3.1 Background Knowledge

In her work on the foundations of entity representation [2], Bazzanella establishes six top-level entity types that are both specialized and generic enough to serve as a basic model for entity representation and matching in general-purpose environments such as the (Semantic) Web. These are: PERSON, ORGANIZATION, EVENT, ARTIFACT, LOCATION, OTHER.

In a feature-listing experiment with more than 350 participants, [2] compiles background knowledge about how humans describe entities, which – for the sake of completeness – we cite in Table 1.

Table 1. Relevance of features for describing basic entity types. (cf. [2]).

Entity Type (e)	**Attribute type** (a)	$p(e\|a)$	**Entity Type** (e)	**Attribute type** (a)	$p(e\|a)$
Person	surname	0.97	*Artifact*	artifact type	0.97
	first name	0.96		artifact name	0.94
	full name	0.96		brand	0.93
	affiliation	0.85		model	0.91
	occupation	0.83		features	0.83
Organization	organization name	0.98	*Location*	location name	0.98
	activity	0.85		location type	0.89
	organization type	0.85		use	0.63
	part of	0.49		place:province	0.59
	place:country	0.15		attraction	0.55
Event	event type	0.97			
	event name	0.96			
	date:year	0.92			
	date:month	0.84			
	protagonist:surname	0.79			

From these data we develop the following working hypothesis: we have two types of features (i) generic ones ("name" and "type"), that say nothing about the entity type but have an average importance in the model that is still higher than the one of an "unknown" feature, and (ii) type-discriminative ones that help us infer an entity type from its description; these features and their relevance values can be used beneficially to perform general-purpose entity resolution, by attempting to map two given records describing the same entity to the above model, and using the relevance values to influence the calculation of record similarity between the two.

3.2 An Ontology of Entity Description

In order to make use of the results described in the previous section in an actual algorithm, the FBEM OWL-ontology has been created which captures the aspects of features and feature weights in entity descriptions. The ontology defines concepts for Entity and Feature, as well as the necessary relations between them, as depicted in Fig. 1.

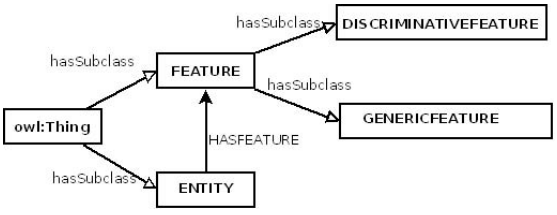

Fig. 1. Class structure of the background KB

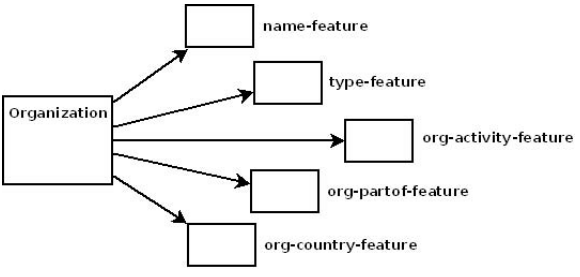

Fig. 2. Example individual of class Entity describing the Artifact type

The actual content of the ontology is in the individuals of the class Entity and its subclasses. One example is given in Fig. 2, which represents the entity type ORGANIZATION and its features such as *activity*, *type* or *partof*. Each of those feature individuals has a set of datatype properties containing the feature weights of type float, and – in order to support synonymity and multilinguality – also a set of so-called *label patterns* which contain the primary name of a feature as established in Sect. 3.1, as well as possible synonyms and natural language versions.

These label patterns are used by the FBEM implementation to attempt establishing a mapping of feature names of the given entities into the FBEM ontology, which is in fact a coarse, but very efficient way of schema matching, which we have implemented for different natural languages (currently German, Italian and English).

4 A Feature-Based Entity Similarity Model

4.1 The FBEM Similarity Score

To present a ranked list of candidate entities that match a reference entity (or a query), we require a score that serves as parameter for ranking, which expresses the closeness of a candidate entity to the reference entity, relative to all other candidates.

In our setting, the representation of a reference entity Q and candidate entity E is modeled as the set of features F plus an optional type t. The type information is not required to follow any form or (natural) language, but is free text:

$$Q \equiv E \equiv < F, t >;$$

The first part of the representation of Q and E is in the form of a set F of *features*, which are represented as <name, value> pairs that are independent in content and size (i.e. they don't necessarily share a vocabulary or schema, or even a natural language):

$$f =< n, v >;$$
$$F = \{f_1, f_2, ...\};$$

We define the following functions:

$n(f)$**:** returns the *name* part of a feature f of an E or Q;
$v(f)$**:** returns the *value* part of a feature f of an E or Q;
$typeof(E)$**:** returns the ontological type of an E or Q.

In order to establish similarity between two entities, we require additional operators and functions that are later used for the computation.

$$mapsto(f) =_{def} \exists x, f(\text{FEATURE}(x) \wedge f \in F \wedge \text{HASLABELPATTERN}(x, n(f)) \quad (1)$$

Axiom 1 provides the definition of a function that maps a feature of Q or E into the background KB. The function must return the feature into which f has been mapped, or NIL.

$$typeDesc(f) =_{def} \exists x(\text{DISCRIMINATIVEFEATURE}(x) \wedge mapsto(f) \equiv x) \quad (2)$$

Axiom 2 provides the definition of a boolean function that determines whether a feature of Q or E is type-descriminative. It does so by determining whether a feature can be mapped into the set of type-descriminative features of the background KB.

We furthermore define three boolean operators that describe whether Q and E are type-compatible(c_E), incompatible (ci_E), or of unknown compatibility (cu_E):

$$c_E(Q, E) =_{def} typeof(Q) \equiv typeof(E) \quad (3)$$

Axiom 3 states that Q and E are considered compatible if they have the same ontological type.

$$ci_E(Q, E) =_{def} typeof(Q) \neq typeof(E) \quad (4)$$

Axiom 4 defines that Q and E are considered incompatible if they have different ontological types in the background KB.

$$cu_E(Q, E) =_{def} \neg ci_E(Q, E) \wedge \neg c_E(Q, E) \quad (5)$$

Axiom 5 defines that Q and E are considered of unknown compatibility if they are neither compatible nor incompatible.

In order to involve background knowledge into the calculation of entity similarity, we also need to define a set of operators that indicate whether a pair of features f_Q and f_E are name-identical(cid_f), name-compatible(c_f), name-incompatible (ci_f) or of unknown compatibility (cu_f):

$$cid_f(f_Q, f_E) =_{def} \\ (c_E(Q,E) \vee cu_E(Q,E)) \wedge ((mapsto(n(Q)) \equiv mapsto(n(E)))) \quad (6)$$

Axiom 6 defines that two features are known to be name-identical with respect to the feature name, if their respective records Q and E are not incompatible, and if the features map into the same ontological feature of the background KB.

$$c_f(f_Q, f_E) =_{def} \\ (c_E(Q,E) \vee cu_E(Q,E)) \wedge (n(Q) \equiv n(E)) \quad (7)$$

Axiom 7 defines that two features are name-compatible if their respective records Q and E are not incompatible, and the respective feature names are identical strings (but unknown with respect to our background KB).

$$ci_f(f_Q, f_E) =_{def} \\ (ci_E(Q,E)) \vee (mapsto(n(Q)) \neq mapsto(n(E))) \quad (8)$$

Axiom 8 defines that two features are name-incompatible if their respective records Q and E are incompatible, or if they map into different ontological features of the background KB.

$$cu_f(f_Q, f_E) =_{def} \\ \neg cid_f(f_Q, f_E) \wedge \neg c_f(f_Q, f_E) \wedge \neg ci_f(f_Q, f_E) \quad (9)$$

Axiom 9 defines that two features are of unknown compatibility if they are neither name-identical, name-compatible or name-incompatible.

Now, we define $fsim(f_Q, f_E)$, a function that computes the similarity of two features f_Q, f_E, taking into account the similarity of the value parts of f_x, as well as our background knowledge base:

$$fsim(f_Q, f_E) =_{def} \\ sim\left(v(f_Q), v(f_E)\right) * \begin{cases} w_c + p(E|f_E), & \text{for } cid_f(f_Q, f_E); \\ w_c, & \text{for } c_f(f_Q, f_E); \\ w_u, & \text{for } cu_f(f_Q, f_E); \\ 0, & \text{otherwise .} \end{cases} \quad (10)$$

Equation 10 relies on the following functions and parameters:

$sim(x, y)$: a suitable string similarity measure between x and y.

$p(E|f_E)$: the relevance of the feature to describe a given entity type, as defined in the background KB (see Sect. 3.1); please note that $p(Q|f_Q) = p(E|f_E)$ because the model requires that the features map into the same element of the background KB;

w_c : the importance which is given to the fact that a pair of features is compatible;

w_u : the importance which is given to the fact that a pair of features is of unknown compatibility.;

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[1]. 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, $|Q|, |E|$ being the number of elements of the vectors Q and E, respectively, and $\mathbb{Q}$ is set of rational numbers.

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

$$fs\,(Q,E) = \sum_{i=1}^{|Q|} maxv(M_i) \tag{11}$$

Here M_i is i^{th} row of the matrix M. Please note that when type of entity E and that of query Q are given or can be infered and found to be not matching, E and Q will be incompatible. Our approach is modeled in such a way that when Q and E is incompatible, the similarity score for them will be zero.

Taking into account that M_i is a weighted value, we use a dot-notation denote its weight w as $M_i.w$. Using a mathematical normalization of the similarity score, the final entity similarity measure $esim(Q,E)$ can be defined as follows:

$$esim(Q,E) = \frac{fs(Q,E)}{\sum_{i=1}^{|Q|} maxv(M_i).w} \tag{12}$$

In the last formula, we 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., $[0,1]$.

4.2 Implementation

For implementing an algorithm that can provide the entity similarity defined in Eq. 12, there are several variables to set and operators to implement which can not be trivially derived from the definitions in the previous section. In the following we give details about their implementation.

$typeof(E)$**:** An implementation of the the $typeof$ operator is required to return the ontological type of the entity E or the intended ontological type of the desired entity that is described by Q. Compatibility of types will later influence the similarity that is established between a Q and an E (especially type-incompatibility will lead to a significantly lower score). The ontological type of Q or E can be inferred from a successful mapping of a given feature into an individual of class Discriminative-Feature in the FBEM ontology, or by an explicit specification of the type. Please

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

note that due to the flexibility of this model, the involvement of a sophisticated schema-matching component from such a freely specified type to one of the types in our knowledge model may be of benefit. For the results presented in this article, the implementations of various functions are given below.

$mapsto(f)$**:** The $mapsto$ function establishes a mapping between a feature f and an instance of the class FEATURE of our background ontology. It does so by checking whether it is possible to establish a string match between $n(f)$ and one of the values of the HASLABELPATTERN property of the instance.

$sim(x, y)$**:** This operator returns a similarity measure between the strings x and y, in the range of [0,1].

$p(E|f_E))$**:** The relevance of a feature for a certain entity is retrieved from the background KB, using the $mapsto$ operator. If the $mapsto$ function fails, this value will be unknown.

w_c**:** This weight can be empirically optimized, default is 0.5.

w_u**:** This weight can be empirically optimized, default is 0.25.

The algorithm implements certain optimizations, e.g. the reasoning that is required to derive the necessary facts from the background ontology is being performend at loading time of the algorithm, materialized and then cached in very efficient memory structures because direct interaction with the OWL model has proven to be unsustainable in terms of runtime performance.

5 Evaluation

5.1 Evaluating String Similarity Measures

As explained in the previous sections, the FBEM algorithm performs string similarity matching between a selection of values of the entities that are to be compared. To understand what is the impact on the selection of a string similarity measure on the overall performance of the algorithm, an experiment has been performed that runs FBEM on the Eprints and Rexa datasets (see Sect. 5.5) using four different metrics: the well-known Levenshtein [13] and Soundex algorithms, as well as Monge-Elkan [14] and TagLink [6]. The results reported in Table 2 illustrate quite drastically that a poor choice of string similarity measure has a negative impact on the quality of matching results. While Levenshtein delivers good results, it does so in very few cases. Other approaches such as Soundex or Monge-Elkan fail our requirements completely. Only the TagLink measure provides acceptable results, and a possible reason may be that the algorithm establishes similarities between tokens of strings which makes it less vulnerable to difference in token sequence (as is the case for names, such as “Barack Obama” vs. “Obama, Barack. T.”).

5.2 Record Identity Tests

To perform a baseline evaluation of the FBEM algorithm, a test of record identity has been performed to measure the quality of the algorithm when no difference between records exist. The dataset that was used is a collection of 50,000 *entity profiles* taken

Table 2. FBEM performance using different string similarity measures

	Levenshtein	Soundex	Monge-Elkan	TagLink
Precision	0.95	0.00	0.01	0.72
Recall	0.05	0.16	0.48	0.77
F-Measure	0.10	0.01	0.02	0.75

Table 3. Experimental results analyzing the matching of identical records (full duplicates)

Experiment 1		Experiment 2	
Dataset A size	100	Dataset A size	100
Dataset B size	100	Dataset B size	50,000
Overlap A ∩ B	100	Overlap A ∩ B	100
Precision	1	Precision	1
Recall	1	Recall	1

from the Entity Name System[2], which contains a majority of geographic entities, as well as some organizations and persons. The records are represented in a flat list of free-form name/value pairs that follow no particular schema [1].

Starting from this basic dataset, we performed two experiments. One is a random selection of 100 samples that were matched against each other (carthesian product). A second experiment evaluates result quality of 100 random samples when matched against the complete dataset of 50,000 records. The results are reported in Table 3.

5.3 Aligning Restaurant Records

The restaurant dataset[3] is composed of about 864 restaurant records from two different data sources (Fodor's and Zagat's restaurant guides), which are described by name, street, city, phone and restaurant category. Among these, 112 record pairs refer to the same entity, but usually display certain differences. An example is given in Table 4.

The objective of this experiment was to use real-world data that do not cover the "usual suspects" such as scientific articles or authors. In detail, the data were organized in a way that the 112 records from Fodor's which had a counterpart in Zagat's were added to a dataset A, and all the others were merged into a dataset B. Then we performed an evaluation using the FBEM algorithm without any modifications, to measure how many entries in dataset A could be successfully found in dataset B. The quality of results is very promising, both recall and precision are above 0.95; more details are reported in Table 5.

5.4 Aligning Person Records

The *people2* dataset[4] contains two files, A with original records of people and B another with modified records from the first file. B contains maximum 9 modified entries for

[2] `http://www.okkam.org`

[3] Originally provided by Sheila Tejada, downloaded from `http://www.cs.utexas.edu/users/ml/riddle/data.html`

[4] Febr project: `http://sourceforge.net/projects/febrl/`

Table 4. Example records of the restaurants data set

Fodor's		Zagat's	
name	carmine's	name	carmine's
street	2450 broadway between 90th and 91st sts.	street	2450 broadway
city	new york	city	new york city
phone	212/362-2200	phone	212-362-2200

Table 5. Evaluation results of FBEM on the restaurant dataset

Dataset details		Results	
Dataset A size	112	False positives	2
Dataset B size	744	False negatives	4
Overlap A ∩ B	112	Recall	95%
		Precision	98%

an original records in A, with maximum 3 modifications per attribute, and maximum 10 modifications per record. The original file contains 600 records, while B contains 400 records which are modifications of 95 original records from A. The attributes of the records are record id, given name, surname, street number, address, suburb, postcode, state, date of birth, age, phone number, social security number. An example is given in Table 6. It is evident from the example that there substantial modifications per record. Still, the results show precision and recall both above 0.77; more details are reported in Table 7.

Table 6. Example records of the people2 data set

	A		B
rec_id	2280	rec_id	2285
given_name	kate	given_name	katrh
surname	peat	surname	peat
street_number	**111**	street_number	**1**
...			
address_1	duffy street	address_1	street duffy
suburb	robina	suburb	robivna
date_of_birth	19450303	date_of_birth	19450033
phone_number	02 90449592	phone_number	04 03014449
...			

As evident in table 8 there are 505 entries which are not present in the alignment. This dataset was used to test for true negatives, i.e. the ability of the algorithm to decide that a searched entity is *not* present in the target data. This experiment is important for scenarios where concrete decisions have to be taken, instead of delivering only a list potential matches without giving further information about their quality. The results for the experiment are promising, with accuracy at 95% for the given dataset. Table 8 gives the details of the experiment.

Table 7. Evaluation results of FBEM on the people2 dataset

Dataset details		Results	
Dataset A size	600	False positives	71
Dataset B size	400	False negatives	93
Overlap A ∩ B	400	Recall	77%
		Precision	81%
		F-Measure	79%
		Fallout	19%

Table 8. Results characterizing the ability to decide about true negatives

Total number of entities in dataset A size	600
Number of entities which have no correspondence in B	505
Total number of entities which have no duplicate in our result	488
Number of missing entities	26
Number of falsely detected entities	9
Accuracy	95%

5.5 Aligning Bibliographic Databases

For the experiments described in this section we used benchmarks provided by the instance matching contest of the Ontology Alignment Evaluation Initiative 2009 (OAEI)[5]. Each benchmark is accompanied with a gold standard in alignment format [10]. The gold-standard alignment is the set of pairs of entities that are known to match, which is then used to evaluate precision and recall of the alignment produced by the experiment.

The benchmark consists of three datasets containing instances from the domain of scientific publications, i.e. information about authors and their publications in proceedings, journals and books.

Eprints contains data about papers produced for the AKT research project. The dataset contains around 1000 entities.

Rexa is generated from the search results of the search server Rexa. The dataset contains over 20000 entities.

SWETO-DBLP is a version of DBLP modeled against the SWETO ontology. The dataset contains over 1000000 entities.

For this experiment, the data are homogenized into the flat name/value pair format that the FBEM algorithm accepts as input. Note that the homogenization is not an integral part of the FBEM algorithm, and thus can be customized. However, a simple, generic homogenizer for RDF data was used for the described experiments in order to produce comparable results. This homogenizer performs no particular reasoning or encodes any kind of knowledge about the underlying data, it only converts the long WWW-style URIs into short names, and removes relations that point to other classes or individuals, thus maintaining only datatype properties directly associated to the entity at hand.

[5] All data are available from `http://oaei.ontologymatching.org/2009/`

Table 9. Matching results for the Eprints-Rexa-DBLP benchmark

	Precision	Recall	F-Measure	Fallout
Eprints – Rexa	0.72	0.78	0.75	0.28
Eprints – DBLP	0.91	0.78	0.84	0.09
Rexa – DBLP	0.83	0.94	0.88	0.17

Table 10. Type-specific matching results between the Eprints and Rexa datasets

	Precision	Recall	F-Measure	Fallout
Person	0.70	0.81	0.73	0.30
Article	0.77	0.71	0.74	0.23
Inproceedings	0.96	0.69	0.80	0.04

In the experiment, the datasets Eprints, Rexa and DBLP were were aligned pairwise. In order to achieve a result within a reasonable timeframe, the large amount of data contained in DBLP was reduced to the entities that are actually contained in DBLP-related gold standard, and then aligned w.r.t. eprints and rexa datasets. Finally, the resulting alignments were evaluated against the corresponding golden standard provided with the datasets.

The matching results are reported in Table 9, and give a positive overall picture. Important to note are the values for Fallout, which is defined as (1-Precision) and reflects the amount of wrong mappings.

In order to gain a better understanding of the detailed strengths and weaknesses of FBEM w.r.t. entity types that are being matched, an additional experiment was performed that does typed matching between the Eprints and Rexa data. The three types that were selected from the datasets are “person”, “article” and “inproceedings”. The results in Table 10 does however not exhibit a very clear trend that could lead to a solid identification of entity types for which FBEM is particularly (un)usable. In terms of precision it behaves better on the larger records that describe publications, and less well on Person data, whereas recall behaviour is the opposite.

6 Discussion and Future Work

One important result presented in this paper is the strong influence that a string similarity metric has on the quality of the matching results. In Sect. 5 we have experimented with a selection of metrics which showed quite drastic differences in outcome. For this reason, one of the next steps will be to work on a component for FBEM that implements heuristics which will allow to select a suitable similarity measure for a given feature value. Such heuristics can base on aspects such as single-token vs. multi-token strings, well-known patterns, and the detection of specific names, e.g. for persons or companies, for which highly specialized algorithms exist.

A second goal is to further broaden the scope of the evaluation of the approach. While this aspect has been substantially expanded since the publication of the predecessor approach, the analyses we have performed have not yet shown results diverse enough

to understand which kind of data and/or entity types FBEM is particularly suitable for. For this reason, more heterogeneous benchmarks will be performed.

A further, mid-term goal is to address more in detail the aspects of large-scale, high-performance entity resolution. Several performance aspects have already been kept in mind during the development of the FBEM implementation. Nonetheless, several additional aspects have to be adressed. First, a good selection of stopping techniques needs to be compiled which will allow the algorithm to cease comparing features when a highly relevant match has been found (e.g. an identical social security number). Finally, blocking techniques will be analyzed to limit the amount of entity-to-entity comparisons. For example, the Entity Name System [4] implements a high-performance, index-based preselection of candidates, which are then further compared with more sophisticated, but also more costly methods. This approach will be one of the first to be evaluated.

7 Conclusion

In this paper we have presented a probabilistic, general-purpose approach for entity resolution which bases on background knowledge about entities and their representation. We have illustrated the important role that entity resolution plays in the engineering of applications that require good quality data integration, and have shown in a series of experiments that for common entity types such as people, organizations or locations, the FBEM approach delivers satisfying results, both in recall and in precision. While already performing at a good level of quality, several areas of improvement have been identified and discussed. These will be addressed in future evolutions of the approach.

In preparing this work it has proven quite difficult to collect suitable datasets which include evaluation standards, even though related work has been performed for many years. Publication not only of evaluation results, but also of benchmarks, seems vital in this context. The authors thus plan to compile the data used in this and future experiments in a coherent way, so that benchmarking of related approaches is rendered considerably more easy.

References

[1] Bazzanella, B., Chaudhry, J.A., Palpanas, T., Stoermer, H.: Towards a General Entity Representation Model. In: Proceedings of the 5th Workshop on Semantic Web Applications and Perspectives (SWAP 2008), Rome, Italy (December 2008)

[2] Bazzanella, B., Stoermer, H., Bouquet, P.: Top Level Categories and Attributes for Entity Representation. Technical Report 1, University of Trento, Scienze della Cognizione e della Formazione (September 2008)

[3] Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S.E., Widomr, J., Jonas, J.: Swoosh: A Generic Approach to Entity Resolution. Technical report, Stanford InfoLab (2006)

[4] Bouquet, P., Stoermer, H., Niederee, C., Mana, A.: 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, August 2008, pp. 554–561. IEEE Computer Society, Los Alamitos (2008), CSS-ICSC 2008-4-28-25

[5] Brizan, D.G., Tansel, A.U.: A Survey of Entity Resolution and Record Linkage Methodologies. Communications of the IIMA 6(3), 41–50 (2006)
[6] Camacho, H., Salhi, A.: A string metric based on a one to one greedy matching algorithm. In: Research in Computer Science number, pp. 171–182 (2006)
[7] Cohen, W.W., Ravikumar, P., Fienberg, S.E.: A Comparison of String Distance Metrics for Name-Matching Tasks. In: Proceedings of the IJCAI 2003 Workshop IIWeb, Acapulco, México, August 9-10, pp. 73–78 (2003)
[8] Dong, X., Halevy, A., Madhavan, J.: Reference Reconciliation in Complex Information Spaces. In: SIGMOD 2005: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, pp. 85–96. ACM Press, New York (2005)
[9] Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate Record Detection: A Survey. IEEE Transactions on Knowledge and Data Engineering 19(1), 1–16 (2007)
[10] Euzenat, J.: An api for ontology alignment. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 698–712. Springer, Heidelberg (2004)
[11] Garcia-Molina, H.: Pair-wise entity resolution: overview and challenges. In: Yu, P.S., Tsotras, V.J., Fox, E.A., Liu, B. (eds.) Proceedings CIKM 2006, Arlington, Virginia, USA, November 6-11, p. 1. ACM, New York (2006)
[12] Hernández, M.A., Stolfo, S.J.: The merge/purge problem for large databases. SIGMOD Rec. 24(2), 127–138 (1995)
[13] Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10, 707–710 (1966)
[14] Monge, A.E., Elkan, C.: An Efficient Domain-Independent Algorithm for Detecting Approximately Duplicate Database Records. In: DMKD (1997)
[15] Noy, N.F.: Semantic Integration: a Survey of Ontology-based Approaches. SIGMOD Rec. 33(4), 65–70 (2004)
[16] Rahm, E., Bernstein, P.A.: A Survey of Approaches to Automatic Schema Matching. VLDB Journal: Very Large Data Bases 10(4), 334–350 (2001)
[17] Stoermer, H., Bouquet, P.: A Novel Approach for Entity Linkage. In: Zhang, K., Alhajj, R. (eds.) Proceedings of IRI 2009, the 10th IEEE Internationational Conference on Information Reuse and Integration, Las Vegas, USA, August 10-12. IRI, vol. 10, pp. 151–156. IEEE Systems, Man and Cybernetics Society (2009)
[18] Tejada, S., Knoblock, C.A., Minton, S.: Learning object identification rules for information integration. Inf. Syst. 26(8), 607–633 (2001)
[19] Winkler, W.E.: The State of Record Linkage and Current Research Problems. Technical report, Statistical Research Division, U.S. Census Bureau, Washington, DC (1999)