Undefined 1 (2009) 1–5
IOS Press

EQUATER - An Unsupervised Data-driven Method to Discover Equivalent Relations in Large Linked Datasets

Editor(s): Name Surname, University, Country
Solicited review(s): Name Surname, University, Country
Open review(s): Name Surname, University, Country

Ziqi Zhang [a,*], Anna Lisa Gentile [a] and Eva Blomqvist [b] Isabelle Augenstein [a] Fabio Ciravegna [a]
[a] *Department of Computer Science, University of Sheffield, Regent Court, 211 Portobello, Sheffield, S1 4DP*
E-mail: {ziqi.zhang,a.gentile,i.augenstein,f.ciravegna}@sheffield.ac.uk
[b] *Department of Computer and Information Science, Linköping University, 581 83 Linköping, Sweden*
E-mail: eva.blomqvist@liu.se

Abstract. The Web of Data is currently undergoing an unprecedented level of growth thanks to the Linked Open Data effort. One escalated issue is the increasing level of heterogeneity in the published resources. This seriously hampers interoperability of Semantic Web applications. A decade of effort in the research of Ontology Alignment has contributed to a rich literature dedicated to such problems. However, existing methods can be still limited when applied to the domain of Linked Open Data, where the widely adopted assumption of 'well-formed' ontologies breaks due to the larger degree of incompleteness, noise and inconsistency both found in the schemata and in the data described by them. Such problems become even more noticeable in the problem of aligning relations, which is very important but insufficiently addressed. This article makes contribution to this particular problem by introducing EQUATER, a domain- and language-independent and completely unsupervised method to align equivalent relations across schemata based on their shared instances. Included by EQUATER are a novel similarity measure able to cope with unbalanced population of schema elements, an unsupervised technique to automatically decide similarity cutoff thresholds to assert equivalence, and an unsupervised clustering process to discover groups of equivalent relations across different schemata. The current version of EQUATER is particularly suited for a more specific yet realistic case: addressing alignment within a single large Linked Dataset, the problem that is becoming increasingly prominent as collaborative authoring is adopted by many large-scale knowledge bases. Using three datasets created based on DBpedia (the largest of which is based on a real problem currently concerning the DBpedia community), we show encouraging results from a thorough evaluation involving four baseline similarity measures and over 15 comparative models by replacing EQUATER components with their alternatives: the proposed EQUATER makes significant improvement over baseline models in terms of F1 measure (mostly between 7% and 40%). It always scores the highest precision and is also among the top performers in terms of recall. Together with the released dataset to encourage comparative studies, this work contributes valuable resources to the related area of research.

Keywords: ontology alignment, ontology mapping, Linked Data, DBpedia, similarity measure

1. Introduction

The Web of Data is currently seeing remarkable growth under the Linked Open Data (LOD) community effort. The LOD cloud currently contains over 870 datasets and more than 62 billion triples[1]. It is becoming a gigantic, constantly growing and extremely

*Corresponding author. E-mail: ziqi.zhang@sheffield.ac.uk.

[1] http://stats.lod2.eu/, visited on 01-11-2013

valuable knowledge source useful to many applications [30,15]. Following the rapid growth of the Web of Data is the increasingly pressing issue of heterogeneity, the phenomenon that multiple vocabularies exist to describe overlapped or even the same domains, and the same objects are labeled with different identifiers. The former is usually referred to schema-level heterogeneity and the latter as data or instance-level heterogeneity. It is widely recognized that currently LOD datasets are characterized by dense links at data-level but very sparse links at schema-level [40,24,14]. This may hamper the usability of data over large scale and decreases interoperability between Semantic Web applications built on LOD datasets. This work explores this issue and particularly studies linking relations across different schemata in the LOD domain, a problem that is currently under-represented in the literature.

Research in the area of Ontology Alignment [13,41] has contributed to a plethora of methods towards solving heterogeneity on the Semantic Web. The mainstream work [36,28,37,21,25,29,43,8,6,38] is archived under the Ontology Alignment Evaluation Initiative (OAEI) [16]. However, it has been criticized that these methods are tailored to cope with nicely structured and well defined ontologies [17], which are different from LOD ontologies characterized by noise and incompleteness [39,40,47,14,17,51]

Despite such rich literature, we notice that aligning heterogeneous relations is not yet well-addressed, especially in the LOD context. Recent research has found that this problem is considered to be harder than, e.g., aligning classes or concepts [18,14,5]. Relation names are more diverse than concept names [5], and the synonymy and polysemy problems are also more typical [14,5]. This makes aligning relations in the LOD domain more challenging. Structural information of relations is particularly lacking [14,51], and the inconsistency between intended meaning of schemata and their usage in data is more wide-spread [18,14,17].

Further, a common limitation to nearly all existing methods is the need for setting a cutoff threshold of computed similarity scores in order to assert correspondences. It is known that the performance of different methods are very sensitive to thresholds [37,29,46,19,5], while finding optimal thresholds requires expensive tuning and availability of annotations; unfortunately, the thresholds are often context-dependent and requires re-tuning for different tasks [22,42].

This work focuses specifically on linking heterogeneous relations in the LOD domain. We introduce EQUATER (EQUivalent relATion findER), a completely unsupervised method for discovering equivalent relations for specific concepts, using only data-level evidence without any schema-level information. EQUATER has three components: (1) a similarity measure that computes pair-wise similarity between relations, designed to cope with the unbalanced (and particularly sparse) population of schemata in LOD datasets; (2) an unsupervised method of detecting cut-off thresholds based on patterns discovered in the data; (3) and an unsupervised clustering process that groups equivalent relations, potentially discovering relation alignments among multiple schemata. The principle of EQUATER is studying the shared instances between two relations, a feature that makes it particularly suited for aligning relations found in a single, large Linked Dataset, such as DBpedia. Although Ontology Alignment is usually concerned about linking schemata and data instances across different datasets, usage of heterogeneous resources is also common in single dataset, and is becoming an increasingly prominent problem as the practice of collaborative authoring encourages integration with existing large LOD datasets by various parties, who often fail to conform to a universal schema. As a realistic scenario, the DBpedia mappings portal[2] is a community effort dedicated to such a problem. Nevertheless, we also discuss how EQUATER can be improved to address cross-dataset alignment.

To thoroughly evaluate EQUATER, we use a number of datasets collected in a controlled manner, including one based on the practical problem faced by the DBpedia mapping portal. We create a large number of comparative models to assess EQUATER along the following dimensions: its similarity measure, capability of coping with dataset featuring unbalanced usage of schemata, automatic threshold detection, and clustering. We report encouraging results from these experiments. EQUATER successfully discovers equivalent relations across multiple schemata, and the similarity measure of EQUATER is shown to significantly outperform all baselines in terms of F1 (maximum improvement of 0.47, or 47%). It also handles unbalanced populations of schema elements and shows stability against several alternative models. Meanwhile, the automatic threshold detection method is shown to be very

[2] http://mappings.dbpedia.org/index.php/Mapping_en, visited on 01 August 2014

competitive - it even outperforms the supervised models on one dataset in terms of F1. Overall we believe that EQUATER provides an effective solution to practical problems in the LOD domain.

In the remainder of this paper, Section 2 discusses related work; Section 3 introduces the method; Section 4 describes a series of designed experiments and 5 discusses results, followed by conclusion in Section 6.

2. Related Work

2.1. Scope and terminology

An alignment between a pair of ontologies is a set of correspondences between entities across the ontologies [13,41]. Ontology entities are usually: *classes* defining the concepts within the ontology; *individuals* denoting the instances of these classes; *literals* representing concrete data values; *datatypes* defining the types that these values can have; and *properties* comprising the definitions of possible associations between individuals, called object properties, or between one individual and a literal, called datatype properties [25]. Properties connect other entities to form statements, which are called *triples* each consisting of a *subject*, a *predicate* (i.e., a property) and an *object*[3]. A correspondence asserts certain relation holds between two ontological entities, and the most frequently studied relations are equivalence and subsumption. Ontology alignment is often discussed at 'schema' or 'instance' level, where the former usually addresses alignment for classes and properties, the latter addresses alignment for individuals. This work belongs to the domain of schema level alignment.

As we shall discuss, in the LOD domains, data are not necessarily described by formal ontologies, but sometimes vocabularies that are simple renderings of relational databases [40]. Therefore in the following, wherever possible, we will use the more general term *schema* or *vocabulary* instead of *ontology*, and *relation* and *concept* instead of *property* and *class*. When we use the terms *class* or *property* we mean strictly in the formal ontology terms unless otherwise stated.

A fundamental operation in discovering ontology alignment is matching pairs of individual entities. Such methods are often called 'matchers' and are usually divided into three categories depending on the type of data they work on [13,41]. *Terminological* matchers work on textual strings such as URIs, labels, comments and descriptions defined for different entities within an ontology. The family of string similarity metrics has been widely employed for this purpose [5]. *Structural* matchers make use of the hierarchy and relations defined between ontological entities. They are closely related to measures of semantic similarity, or relatedness in more general sense [50]. *Extensional* matchers exploit data that constitute the actual population of an ontology or schema in the general sense, and therefore, are often referred to as instance- or data-based methods. For a concept, 'instances' or 'populated data' are individuals in formal ontology terms; for a relation, these can depend on specific matchers, but are typically defined based on triples containing the relation. Matchers compute a degree of similarity between entities in certain numerical range, and use cutoff thresholds to assert correspondences.

We discuss state-of-the-art in three sub-sections in the following: the mainstream work led under the annual OAEI campaigns; work particularly addressing ontology alignment in the LOD domain; and work specifically looking at aligning relations across different schemata.

2.2. OAEI and state-of-the-art

The OAEI maintains a number of well-known public datasets for evaluating ontology alignment methods, and hosts annual campaigns to compare the performance of different systems under a uniform framework. Work under this paradigm has been well-summarized in [13,41]. A predominant pattern shared by these work [36,28,37,21,25,29,43,8,6,38,19] is the strong preference of terminological and structural matchers to extensional methods [41]. There is also a trend of using a combination of different matchers (either across or within categories), since it is argued that the suitability of a matcher is dependent on different scenarios and therefore combining several matchers could improve alignment quality [38]. However, an associated problem is finding an optimal 'configuration' in the combination such as tuning weights associated with different matchers [36,37,29,43,38]. Without these, multi-matcher methods can even underperform single matchers [29]. Several studies have been carried out in this direction, such as Hu et al. [21] and Li et al. [29] that build on the notions of linguistic and structural 'comparability' of two ontologies; Nagy et al. [36,37] that uses the Dempster-Shafer [44] theory to

[3] To be clear, we will always use 'object' in the context of triples; we will always use 'individual' to refer to object instances of classes.

combine different evidences given by different matchers to arrive at a coherent interpretation; and Ngo et al. [38] that uses supervised learning to learn an optimal setting from training data. Most studies showed that ontology alignment can benefit from the use of background knowledge sources such as WordNet and Wikipedia [36,28,37,25,8,19].

General limitations Despite their success at OAEI compaigns, these methods suffer from several limitations even without considering the complications due to the LOD domain or relation heterogeneity. First, it is well-known that terminological matchers easily fail when the linguistic features of two entities differ largely. While to certain degree structural matchers can solve this issue, the problem is that both categories are very ontology-specific. As shown by Jean-Marry et al. [25] and Gruetze et al. [17], two ontologies constructed for the same domain using the same data resources by different experts could be vastly dissimilar in terms of taxonomy and terminological features (e.g., both DBpedia[4] and YAGO[5] ontologies represent knowledge extracted from Wikipedia).

On the other hand, even strong similarity between entities measured at terminological or structural level does not always imply strict equivalence, since ontological schemata may be interpreted in different ways by data publishers thus creating inconsistency between their intended meanings and actual usage patterns in data [39,17,51] (e.g., *foaf*[6]*:Person* may represent researchers in a scientific publication dataset, but artists in a music dataset).

Many state-of-the-art methods use a combination of different matchers. This does not necessarily solve the problems but can significantly increase complexity. Similarity computation is typically quadratic; the more matchers are combined, the larger the search space grows and the more computation is required. Additional effort is also required to carefully combine output from different matchers in a coherent way. As we shall discuss in the next sections, tougher challenges arise in the LOD domain or in the problem of aligning heterogeneous relations, as some of those problems become even more typical.

[4]http://dbpedia.org/Ontology, visited on 01-11-2013.

[5]http://www.mpi-inf.mpg.de/yago-naga/yago/, visited on 01-11-2013

[6]foaf:http://xmlns.com/foaf/0.1/

2.3. Ontology alignment in the LOD domain

Some characteristics of Linked Data require particular attention when adapting ontology alignment methods from the classic OAEI scenario to the LOD domain. First and foremost, vocabulary definitions are often highly heterogeneous and incomplete [17]. Textual features such as labels and comments for concepts and relations that are used by almost every method documented by OAEI are non-existent in some large ontologies. In particular, many vocabularies generated from (semi-)automatically created large datasets are based on simple rendering of relational databases and are unlikely to contain such information. For instance, Fu et al. [14] showed that the DBpedia ontology contained little linguistic information about relations except their names. The problem of inconsistency between the intended definitions and the actual usage of concepts and relations discussed before, is particularly prominent in the LOD domain [39,47,17], making such kinds of information unreliable evidence even if they are available. Empirically, Jain et al. [23] and Cruz et al. [6] showed that the top-performing systems in 'classic' ontology alignment settings such as the OAEI do not have clear advantage over others in the LOD domain.

Another feature of the Linked Data environment is the presence of large volumes of data and the availability of many interconnected information sources [39,40,46,17]. Thus extensional matchers can be better suited for the problem of ontology alignment in the LOD domain as they provide valuable insights into the contents and meaning of schema entities from the way they are used in data [39,47].

The majority of state-of-the-art in the LOD domain employed extensional matchers. Nikolov et al. [39] proposed to recursively compute concept mappings and entity mappings based on each other. Suchanek et al. [46] built a holistic model starting with initializing probabilities of correspondences based on instance (for both concepts and relations) overlap, then iteratively re-compute probabilities until convergence. However, equivalence between relations are not addressed.

Parundekar et al. [40] discussed aligning ontologies that are defined at different levels of granularity, which is common in the LOD domain. As a concrete example, they mapped the only class in the GeoNames[7] ontology - *geonames:Feature* - with a well defined one,

[7]http://www.geonames.org/

such as the DBpedia ontology, by using the notion of 'restriction class'. Slabbekoorn et al. [45] explored a similar problem: matching a domain-specific ontology to a general purpose ontology.

Jain et al. [24] proposed BLOOMS+, the idea of which is to build representations of concepts as subtrees from Wikipedia category hierarchy, then determine equivalence between concepts based on the overlap in their representations. Both structural and extensional matcher are used and combined. Cruz et al. [7] created a customization of the AgreementMaker system [8] to address ontology alignment in the LOD context, and achieved better average precision but worse recall than BLOOMS+. Gruetze et al. [17] and Duan et al. [12] also used extensional matchers in the LOD context but focusing on improving computation efficiency of the algorithms.

A review of these methods show that many have strong preference towards using extensional matchers for ontology alignment in the LOD domain, but they typically focus on aligning concepts not relations. As we shall discuss in the following, aligning relations involves new challenges.

2.4. Matching relations

Compared to concepts, aligning relations is generally considered to be harder [18,14,5]. The challenges concerning the LOD domain can become more noticeable when dealing with relations. In terms of linguistic features, relation names can be more diverse than concept names, this is because they frequently involve verbs that can appear in a wider variety of forms than nouns, and contain more functional words such as articles and prepositions [5]. The synonymy and polysemy problems are common. Verbs in relation names are more generic than nouns in concept names and therefore, they generally have more synonyms [14,5].

Same relation names are frequently found to bear different meanings in different contexts [18], e.g., in the DBpedia dataset 'before' is used to describe relationship between consecutive space missions, or consecutive Academy Award winners [14]. Such polysemy issue causes wide-spread inconsistency between definitions of relations and their actual usage in data. Indeed, Gruetze et al. [17] suggested definitions of relations should be ignored when they are studied in the LOD domain due to such issues.

In terms of structural features, Zhao et al. [51] showed that relations may not have domain or range defined in the LOD domain. Moreover, we carried out a test on the 'well-formed' ontologies released by the OAEI-2013 website, and found that among 21 downloadable[8] ontologies 7 defined relation hierarchy and the average depth is only 3. Fu et al. [14] also showed hierarchical relations between DBpedia properties were very rare.

For these reasons, terminological and structural matchers [53,25,29,38] can be seriously hampered if applied to matching relations, particularly in the LOD domain. Indeed, Cheatham et al. [5] compared a wide selection of string similarity metrics in several tasks and showed their performance on matching relations to be inferior to matching concepts. Thus in line with [39,12], we argue in favour of extensional matchers.

We notice only a few related work specifically focused on matching relations based on data-level evidence. Fu et al. [14] studied mapping relations in the DBpedia dataset. The method uses three types of features: data level, terminological, and structural. Similarity is computed using three types of matchers corresponding to the features. Zhao et al. [52,51] first created triple sets each corresponding to a specific subject that is an individual, such as *dbr:Berlin*. Then initial groups of equivalent relations are identified for each specific subject: if, within the triple set containing the subject, two lexically different relations have identical objects, they are considered equivalent. The initial groups are then pruned by a large collection of terminological and structural matchers, applied to relation names and objects to discover fuzzy matches.

Many extensional matchers used for matching concepts could be adapted to matching relations. One popular strategy is to compare the size of the overlap in the instances of two relations against the size of their total combined, such as the Jaccard and Dice metrics (or similar) used in [22,12,14,17]. However, we argue that in the LOD domain, usage of vocabularies can be extremely unbalanced due to the collaborative nature of LOD. Data publishers have limited knowledge about available vocabularies to describe their data, and in worst cases they simply do not bother [47]. As a result, concepts and relations defined from different vocabularies bearing the same meaning can have different population sizes. In such cases, the above strategy is unlikely to succeed, as suggested in [39].

Another potential issue is that current work assumes relation equivalence to be 'global', while it has been

[8] http://oaei.ontologymatching.org/2013/, visited on 01-11-2013. Some datasets were unavailable at the time.

suggested that, interpretation of relations should be context-dependent, and argued that equivalence should be studied at concept-specific context because essentially relations are defined specifically with respect to concepts [18,14]. Global equivalence cannot deal with the polysemy issue such as the previously illustrated example of 'before' bearing different meanings in different contexts. Further, to our knowledge, there is currently no public dataset specifically for aligning relations in the LOD domain, and current methods [14,52,51] have been evaluated on smaller datasets than those used in this study.

Additionally, it can be argued that aligning relations is closely related to the extensive literature on mapping database schemata. To name a few, Madhavan et al. [34] and Do and Rahm [10] studied methods of combining a number of different matchers to align database tables. Doan et al. [11] introduced a supervised model, using similar features used by extensional and terminological matchers. Kang and Naughton [27] suggested that attributes from two database tables can be aligned based on pair-wise attribute correlation detected within each individual table. This complements the classical extensional and terminological methods. Madhavan et al. [33] looked at using a corpus of schema as additional evidence (to the concerning schemata to be matched) for attribute mapping. Apart from the intrinsic limitations of the highly similar matching methods to those used in the ontology community, the LOD domain introduces new problems such as data sparsity and noise, which can invalidate such methods or require adaptation.

2.5. The cutoff thresholds in matchers

To date, nearly all existing matchers require a cutoff threshold to assert correspondence between entities. The performance of a matcher can be very sensitive to thresholds and finding an optimal point is often necessary to warrant the effectiveness of a matcher [37,29,46,19,5]. Such thresholds are typically decided based on some annotated data (e.g., [29,43,18]), or even arbitrarily in certain cases. In the first case, expensive effort must be spent on annotation and training. In both cases, the thresholds are often context-dependent and requires re-tuning for different tasks [22,43,42].

Another approach adopted in [12] and [14] is to sort the matching results in a descending order of the similarity score, and pick only the top-k results. This suffers from the same problem as cutoff thresholds since the value of k can be different in different contexts (e.g., in [12] this varied from 1 to 86 in the ground truth). To the best of our knowledge, our work is the first that studies the problem of automatically deciding thresholds based on data in ontology alignment.

2.6. Remark

To conclude this section of related work, we argue that aligning relations across schemata in the LOD domain is an important research problem that is currently under-represented in the literature of ontology alignment. The characteristics of relations found in the schemata from the LOD domain, i.e., incomplete (or lack of) definitions, inconsistency between intended meaning of schemata and their usage in data, and very large amount data instances, advocate for a renewed inspection of existing ontology alignment methods. We believe the solution rests in extensional methods that provide insights into the meaning of relations based on data, and unsupervised methods that alleviate the need for threshold tuning.

Towards these directions we developed a prototype [49] specifically to study aligning equivalent relations in the LOD domain. We proposed a different extensional matcher designed to reduce the impact of the unbalanced populations , and a rule-based clustering that employs a series of cutoff thresholds to assert equivalence between relation pairs and discover groups of equivalent relations specific to individual concepts. The method showed very promising results in terms of precision, and was later used in constructing knowledge patterns based on data [2,48]. The work described in this article is built on our prototype but largely extends it in several dimensions: (1) the matcher is largely revised and extended; (2) a method of automatic threshold detection based on data; (3) an unsupervised machine learning clustering approach to discover groups of equivalent relations; (4) augmented and re-annotated datasets that we make available to public; (5) extensive and thorough evaluation against a large set of comparative models, together with an in-depth analysis of the task of aligning relations in the LOD domain.

We focus on equivalence only because firstly, it is considered the major issue in ontology alignment as it is the focus by the majority of related work; secondly, hierarchical structures for relations are very rare, especially in the LOD domain.

<x, r, y>	*<dbr[9]:Sydney_Opera_House, dbo[10]:openningDate, '1973'>, <dbr:Royal_Opera_House, dbo:openningDate, '1732'>, <dbr:Sydney_Opera_House, dbpp[11]:yearsactive, '1973'>*
r_1	*dbo:openningDate*
r_2	*dbpp:yearsactive*
$arg(r_1)$	*(dbr:Sydney_Opera_House, '1973'), (dbr:Royal_Opera_House, '1732')*
$arg_s(r_1)$	*dbr:Sydney_Opera_House, dbr:Royal_Opera_House*
$arg_o(r_1)$	*'1973','1732'*

Table 1
Notations used in this paper and their meaning

3. The EQUATER Method

3.1. Task formalization

In this section we describe EQUATER - our domain- and language-independent method for finding equivalent relations from LOD datasets. EQUATER belongs to the category of extensional matchers according to [13,41], and only uses instances of relations as its evidence to predict equivalence. In the following, we write *<x, r, y>* to represent triples, where x, y and r are variables representing subject, object and relation respectively. We will call x, y the arguments of r, or let $arg(r) = (x, y)$ return pairs of x and y between which r holds true. We call such argument pairs as instances of r. We will also call x the subject of r, or let $arg_s(r) = x$ return the subjects of any triples that contain r. Likewise we call y the object of r or let $arg_o(r) = y$ return the objects of any triples that contain r. Table 1 shows examples using these notations.

EQUATER takes as input a URI representing a specific concept C and a set of triples *<x, r, y>* whose subjects are individuals of C, or formally $type(x) = C$. In other words, we study the relations that link C with everything else. The intuition is that such relations may carry meanings that are specific to the concept (e.g., the example of the DBpedia relation 'before' in the context of different concepts).

Our task can be formalized as: given the set of triples *<x, r, y>* such that x are instances of a particular concept, i.e., $type(x) = C$, determine 1) for any pair of (r_1, r_2) derived from *<x, r, y>* if $r_1 \equiv r_2$; and 2) create clusters of relations that are mutually equivalent. To approach the first goal, we firstly introduce a data-driven similarity measure (Section 3.2). For a specific concept we hypothesize there exists only a handful of truly equivalent relation pairs (true positives) with high similarity scores, however, there can be a large number of pairs of relations with low similarity scores (false positives) due to noise in the data caused by, e.g., misuse of schemata or purely coincidence. Therefore, we propose to automatically detect concept-specific thresholds based on patterns in the similarity scores of relation pairs of the concept. Then pairs with scores beyond the threshold are considered to be equivalent (Section 3.3). For the second goal, we apply unsupervised clustering to the set of equivalent pairs and create clusters of mutually equivalent relations (Section 3.4). Clustering effectively discovers equivalence transitivity or invalidates pair-wise equivalence. This may also discover alignments among multiple schemata at the same time, while state-of-the-art alignment models usually align pairs of schemata.

[9] dbr:http://dbpedia.org/resource/
[10] dbo:http://dbpedia.org/ontolgy/
[11] dbpp:http://dbpedia.org/property/

3.2. Measure of similarity

The goal of the measure is to assess the degree of similarity between a pair of relations within a concept-specific context, as illustrated in Figure 1. The measure consists of three components, the first two of which are previously introduced in our prototype [49].

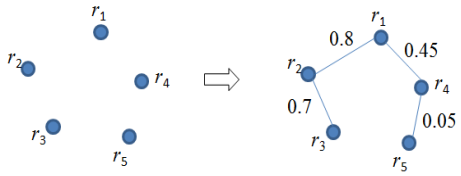

Fig. 1. The similarity measure computes a numerical score for pairs of relations. r_3 and r_5 has a score of 0.

3.2.1. Triple agreement

Triple agreement evaluates the degree of shared argument pairs of two relations in triples. Equation 1 firstly computes the overlap (intersection) of argument pairs between two relations.

$$arg_\cap(r_1, r_2) = arg(r_1) \cap arg(r_2) \quad (1)$$

Then the triple agreement is a function that returns a value between 0 and 1.0:

$$ta(r_1, r_2) = max\{\frac{|arg_{\cap}(r_1, r_2)|}{|arg(r_1)|}, \frac{|arg_{\cap}(r_1, r_2)|}{|arg(r_2)|} \quad (2)$$

The intuition of triple agreement is that if two relations r_1 and r_2 have a large overlap of argument pairs with respect to the size of either relation, they are likely to have an identical meaning. We choose the max of the two values in equation 2 rather than balancing the two as this copes with the unbalanced usage of different schemata in LOD datasets, the problem which we discussed in Section 2.4. As an example, consider Figure 2. The size of argument pair overlap between r_1 and r_2 is 4 and it is relatively large to r_1 but rather insignificant to r_2. ta chooses the maximum between the two giving a strong indication of equivalence between the relations. We note that similar forms have been used in [39] for discovering similar concepts and in [40,46] for studying subsumption relations between concepts. However we believe that this could be used to find equivalent relations due to the largely unbalanced population for different vocabularies, as well as the lack of hierarchical structures for relations as discussed before in Section 2.4. We confirm this empirically in experiments later in Section 5.

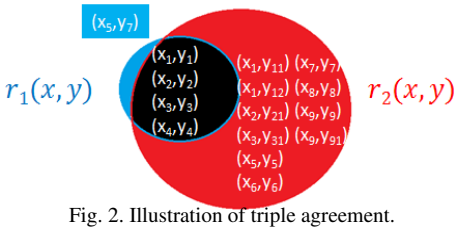

Fig. 2. Illustration of triple agreement.

3.2.2. *Subject agreement*

Subject agreement provides a complementary view by looking at the degree to which two relations share the same subjects. The motivation of having sa in addition to ta can be illustrated by Figure 3. The example produces a low ta score due to the small overlap in the argument pairs of r_1 and r_2. A closer look reveals that although r_1 and r_2 have 7 and 11 argument pairs, they have only 3 and 4 different subjects respectively and two are shared in common. This indicates that both r_1 and r_2 are *1-to-many* relations. Again due to publisher preferences or lack of knowledge, triples may describe the same subject (e.g., *dbr:London*) using heterogeneous relations (e.g., *dbo:birthPlace Of*, *dbpp:place OfOriginOf*) with different sets of objects (e.g., {*dbr:Marc_Quinn*, *dbr:David_Haye*, *dbr:Alan_Keith*} for *dbo:birthPlaceOf* and {*dbr:Alan_Keith*, *dbr:Adele_Dixon*} for *dbpp:placeOfOriginOf*). ta does not discriminate such cases.

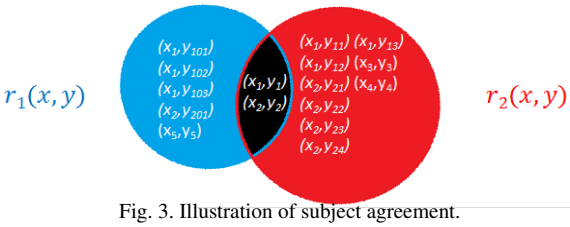

Fig. 3. Illustration of subject agreement.

Subject agreement captures this situation by hypothesizing that two relations are likely to be equivalent if (α) a large number of subjects are shared between them and (β) a large number of such subjects also have shared objects.

$$sub_{\cap}(r_1, r_2) = arg_s(r_1) \cap arg_s(r_2) \quad (3)$$

$$sub_{\cup}(r_1, r_2) = arg_s(r_1) \cup arg_s(r_2) \quad (4)$$

$$\alpha(r_1, r_2) = \frac{|sub_{\cap}(r_1, r_2)|}{|sub_{\cup}(r_1, r_2)|} \quad (5)$$

$$\beta(r_1, r_2) = \frac{\sum_{x \in sub_{\cap}(r_1, r_2)} \begin{cases} 1 & \text{if } \exists y : (x, y) \in arg_{\cap}(r_1, r_2) \\ 0 & \text{otherwise} \end{cases}}{|sub_{\cap}(r_1, r_2)|} \quad (6)$$

, both α and β return a value between 0 and 1.0, and subject agreement combines both to also return a value in the same range as

$$sa(r_1, r_1) = \alpha(r_1, r_2) \cdot \beta(r_1, r_2) \quad (7)$$

Equation 5 evaluates the degree to which two relations share subjects based on the intersection and the union of the subjects of two relations. Equation 6 counts the number of shared subjects that have at least one overlapping object. The higher the β, the more the two relations 'agree' in terms of their shared subjects $sub_{\cap}$. For each subject shared between r_1 and r_2 we count 1 if they have at least 1 object in common and 0

otherwise. Since both r_1 and r_2 can be 1-to-many relations, few overlapping objects could mean that one is densely populated while the other is not, which does not mean they 'disagree'. The agreement $sa(r_1, r_2)$ balances the two factors by taking the product. As a result, relations that have high sa will share many subjects (α), a large proportion of which will also share at least one object (β). Following the example in Figure 3 it is easy to calculate $\alpha = 0.4$, $\beta = 1.0$ and $sa = 0.4$.

3.2.3. *Knowledge confidence modifier*

Although ta and sa computes scores of similarity from different dimensions, as argued by [22], in practice, datasets often have imperfections due to incorrectly annotated instances, data spareness and ambiguity, so that basic statistical measures of co-occurrence might be inappropriate if interpreted in a naive way. Specifically in our case, the divisional equations of ta and sa components can be considered as comparison between two items - sets of elements in this case. From the cognitive point of view, to make a meaningful comparison of two items we must possess adequate knowledge about each such that we 'know' what we are comparing and can confidently identify their 'difference'. Thus we hypothesize that our confidence about the outcome of comparison directly depends on the amount of knowledge we possess about the compared items. We can then solve the problem by solving two subtasks: (1) quantifying knowledge and (2) defining 'adequacy'.

The *quantification of knowledge* can be built on the principle of human *inductive learning* - learning by examples. The intuition is that given a task (e.g., learning to recognize horses) of which no a-priori knowledge is given, humans are capable of generalizing examples and inducing knowledge about the task. Exhausting examples is unnecessary and typically our knowledge converges after seeing certain amount of examples and we learn little from additional examples - a situation that indicates the notion of 'adequacy'. Such a learning process can be modeled by 'learning curves', which are designed to capture the relation between how much we experience (examples) and how much we learn (knowledge). Therefore, we propose to approximate the modeling of confidence by models of learning curves. In the context of EQUATER, the items we need knowledge of are pairs of relations to be compared. Practically, each is represented as a *set* of instances, i.e., *examples*. Thus our knowledge about the relations can be modeled by learning curves corresponding to the number of examples (i.e., argument pairs) in each set.

We propose to model this problem based on the theory by Dewey [9], who suggests human learning follows an 'S-shaped' curve as shown in Figure 4. As we begin to observe examples, our knowledge grows slowly as we may not be able to generalize over limited cases. This is followed by a steep ascending phase where, with enough experience and new re-assuring evidence, we start 'putting things together' and gaining knowledge at a faster phase. This rapid progress continues until we reach convergence, an indication of 'adequacy' and beyond which the addition of examples adds little to our knowledge.

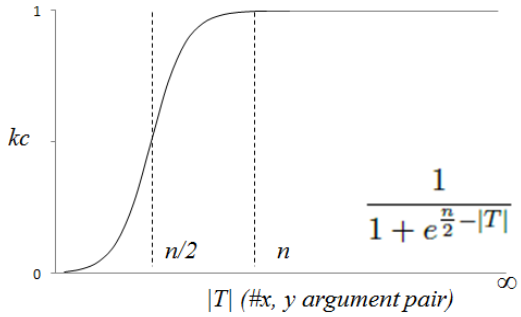

Fig. 4. The logistic function modelling knowledge confidence

Empirically, we model such a curve using a logistic function shown in Equation 8, where T denotes a set of argument pairs of relations we want to understand, kc is the shorthand for *knowledge confidence* (between 0.0 and 1.0) representing the amount of knowledge or level of confidence corresponding to different amounts of examples, and n denotes the number of examples by which one gains adequate knowledge about the set and becomes fully confident about comparisons involving the set (hence the corresponding relation it represents) .

$$kc(|T|) = lgt(|T|) = \frac{1}{1 + e^{\frac{n}{2} - |T|}} \tag{8}$$

It could be argued that other learning curves (e.g., exponential) could be used as alternative; or we could use simple heuristics instead (e.g., discard any relations that have fewer than n argument pairs). However, we believe that the logistic model better fits the problem since the exponential model usually implies rapid convergence, which is hardly the case in many real learning situations; while the simplistic thresh-

old based model may harm recall. We show empirical comparison in Section 5.

Next we revise ta and sa as ta^{kc} and sa^{kc} respectively by integrating the kc measure in the equation:

$$ta^{kc}(r_1, r_2) = max\{\frac{|arg_{\cap}(r_1, r_2)|}{|arg(r_1)|} \cdot kc(|arg(r_1)|), \frac{|arg_{\cap}(r_1, r_2)|}{|arg(r_2)|} \cdot kc(|arg(r_1)|)\} \tag{9}$$

$$sa^{kc}(r_1, r_1) = \alpha(r_1, r_2) \cdot \beta(r_1, r_2) \cdot kc(|arg_{\cap}(r_1, r_2)|) \tag{10}$$

, where the choice of the kc function can be either lgt, or some alternative models to be detailed in Section 4. The choice of the kc function does not break the mathematical consistency of the formula. In Equation 9, our confidence about a ta score depends on our knowledge of either $arg(r_1)$ or $arg(r_2)$ (i.e., the denominators). Note that the denominator is always a superset of the numerator, the knowledge of which we do not need to quantify separately since intuitively, if we know the denominator we should also know its elements and its subsets. Likewise in Equation 10, our confidence about an sa score depends on the knowledge of the shared argument pairs between r_1 and r_2 as any other components in the equation are essentially subsets of this set. Both Equations return a value between 0 and 1.0.

Finally, the similarity of r_1 and r_2 is:

$$e(r_1, r_2) = \frac{ta^{kc}(r_1, r_2) + sa^{kc}(r_1, r_2)}{2} \tag{11}$$

3.3. Determining thresholds

After computing similarity scores for relation pairs of a specific concept, we need to interpret the scores and be able to determine the minimum score that justifies equivalence between two relations (Figure 5). This is also known as the mapping selection problem. As discussed before, one typically derives a threshold from training data or makes an arbitrary decision. The solutions are non-generalizable and the supervised method also requires expensive annotations.

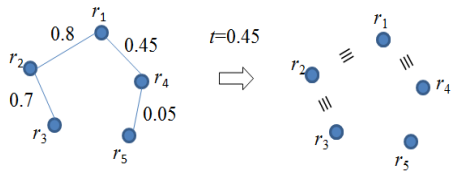

Fig. 5. Deciding a threshold beyond which pairs of relations are considered to be truely equivalent.

We use an unsupervised method that determines thresholds automatically based on observed patterns in data. We hypothesize that a concept may have only a handful of equivalent relation pairs whose similarity scores should be significantly higher than the non-equivalent noisy pairs that may happen to have non-zero similarity scores due to imperfections in data such as spelling errors, schema misuse, or merely coincidence. For example, Figure 6 shows the scores (e) of 101 pairs of relations of the DBpedia concept $Book$ ranked by $e(> 0)$ appear to form a long-tailed pattern consisting of a small population with high similarity and a very large population with low similarity.

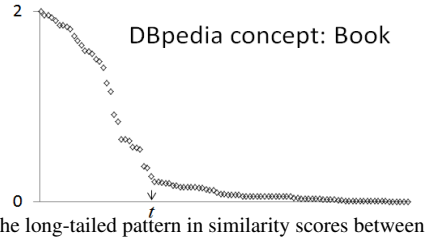

Fig. 6. The long-tailed pattern in similarity scores between relations computed using e. t could be the boundary threshold.

On this basis, we propose to separate the non-zero scored relation pairs into two groups based on the principle of maximizing the difference of similarity scores between the groups. While a wide range of data classification and clustering methods can be applied for this purpose, here we use an unsupervised method - Jenks natural breaks [26].

Jenks natural breaks aims to minimize within-class variance while maximizing between-class variance. Given i the expected number of groups in the data, the algorithm starts by dividing the data into arbitrary i groups, followed by an iterative process aimed at optimizing the 'goodness-of-variance-fit' based on two figures: the sum of squared deviations between classes, and the sum of squared deviations from the array mean. The resulting optimal classification is called Jenks natural breaks.

Empirically, given a continuous variable (i.e., e) and an array of data values (i.e., similarity scores of relation pairs for a concept C), we apply Jenks natural

breaks to the values with $i = 2$ to break them into two sets. The boundary value t is the threshold used to separate relation pairs that we consider as equivalent from those that we consider non-equivalent.

3.4. Clustering

So far Sections 3.2 and 3.3 described our method to answer the first question set out in the beginning of this section, i.e., predicting relation equivalence. The proposed method studies each relation pair independently from other pairs. This may not be sufficient for discovering equivalent relations due to two reasons. First, two relations may be equivalent even though no supporting data are present. For example, in Figure 7 we can assume $r_1 \equiv r_3$ based on transitivity although no data directly supports a positive similarity score between them. Second, a relation may be equivalent to multiple relations (e.g., $r_2 \equiv r_1$ and $r_2 \equiv r_3$) from different schemata, thus forming a cluster; and furthermore some equivalence links may appear too weak to hold when compared to the cluster context (e.g., $e(r_1, r_4)$ appears to be much lower compared to other links in the cluster of r_1, r_2, and r_3).

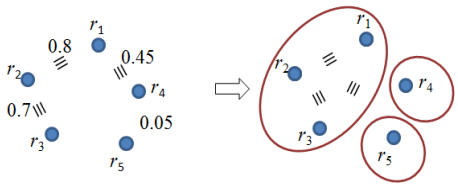

Fig. 7. Clustering discovers transitive equivalence and invalidates weak links.

The second goal of EQUATER is to address such issues by clustering mutually equivalent relations for a concept. Essentially clustering brings in additional context to decide pair-wise equivalence, which may lead to discover transitive equivalence and invalidate weak links. Potentially, this also allows creating alignments between multiple schemata at the same time. Given $\{r_i, r_j : e(r_i, r_j) \geq t\}$ the set of equivalent relation pairs discovered before, we identify the number of distinct relations h and create an $h \times h$ *distance* matrix M. The value of each cell $m_{i,j}, (0 \leq i, j < h)$ is defined as:

$$m_{i,j} = maxE - e(r_i, r_j) \tag{12}$$

where $maxE$ is the maximum possible similarity given by a measure (e.g., in Equation 11 $maxE = 1.0$). Then we use the group-average agglomerative clustering algorithm [35] that takes M as input and creates clusters of equivalent relations. To automatically decide the optimal number of clusters, we use the well-known Calinski and Harabasz [3] stopping rule.

4. Experiment Settings

Following the two goals described at the beginning of Section 3, we design a series of experiments to thoroughly evaluate EQUATER in terms of its capability of predicting equivalence of two relations of a concept (*pair equivalence*) and grouping equivalent relations (*clustering*). Different settings are created along three dimensions by selecting from several choices of (1) similarity measure, (2) threshold detection methods and (3) different models of kc.

4.1. Measures of similarity

We compare the proposed measure of similarity against four baselines. Our criteria for the baseline measures are: 1) to cover different types of matchers; 2) to focus on methods that have been practically shown effective in the LOD context, and where possible, particularly for aligning relations; 3) to include some best performing methods for this particular task.

The first is a string similarity measure, the Levenshtein distance metric (***lev***) that proves to be one of the best performing terminological matcher for aligning both relations and classes [5]. Specifically, we measure the string similarity (or distance) between the URIs of two relations, but we remove namespaces from relation URIs before applying the metric. As a result, *dbpp:name* and *foaf:name* will be both normalized to *name* and thus receiving the maximum similarity score.

The second is a semantic similarity measure by Lin (***lin***) [31], which uses both WordNet's hierarchy and word distributional statistics as features to assess similarity of two words. Thus two lexically different words (e.g., 'cat' and 'dog' can also be similar). Since URIs often contain strings that are concatenation of multiple words (e.g., 'birthPlace'), we use simple heuristics to split them into multiple words when necessary (e.g., 'birth place'). Semantic similarity measures are also popular techniques in ontology alignment.

The third is the extensional matcher proposed by [14] (***fu***) to address particularly the problem of aligning relations in DBpedia:

$$fu(r_1, r_2) = \frac{|arg_s(r_1) \cap arg_s(r_2)|}{|arg_s(r_1) \cup arg_s(r_2)|} \cdot \frac{|arg_o(r_1) \cap arg_o(r_2)|}{|arg_o(r_1) \cup arg_o(r_2)|} \quad (13)$$

The fourth baseline is the 'corrected' Jaccard function proposed by Isaac et al. [22]. The original Jaccard function has been used in a number of studies concerning mapping concepts across ontologies [22,12,42]. Isaac et al. [22] showed that it is one of the best performing measures in their experiment, however, they also pointed out that one of the issue with Jaccard is its inability to consider the absolute sizes of two compared sets. As an example, Jaccard does not distinguish the cases of $\frac{100}{100}$ and $\frac{1}{1}$. In the latter case, there is little evidence to support the score (both 1.0). To address this, they introduced a 'corrected' Jaccard measure (***jc***) as below:

$$jc(r_1, r_2) = \frac{\sqrt{|arg(r_1) \cap arg(r_2)| \cdot (|arg(r_1) \cap arg(r_2)| - 0.8)}}{|arg(r_1) \cup arg(r_2)|} \quad (14)$$

4.2. Methods of detecting thresholds

We compare three different methods of threshold detection. The first is Jenks Natural Breaks ***jk*** that comprises part of EQUATER, discussed in Section 3.3. For the second method we use the k-means clustering [32] algorithm (***km***) for unsupervised threshold detection. K-means takes the same input of *jk* and creates two clusters such that each data value belongs to the cluster with the nearest mean. The boundary value that separates the two clusters are used as threshold. Since both methods find boundaries based on data in an unsupervised manner, we are able to define concept-specific threshold that may fit better than an arbitrarily determined global threshold.

Next, we also use a supervised method (denoted by ***s***) to derive a uniform threshold for all concepts based on annotated data. To do so, suppose we have a set of m concepts and for each concept, we create pairs of relations found in data and ask humans to annotate each pair (to be detailed in Section 4.5). This becomes the training data that we use to derive a uniform threshold. Then we choose a similarity measure to be evaluated, and use it to score each pair and rank results by scores. Using the annotations, we can evaluate accuracy at each rank and at certain rank the accuracy should be maximized. We record the similarity score at this rank, and use it as the optimal threshold for that concept. Due to the difference in concept-specific data, we expect to obtain different optimal thresholds for each of the m concepts in the training data. However, in reality, the thresholds for new data will be unknown a-priori. Therefore we use the average of all thresholds derived from the training data concepts as an approximation and use it for testing.

4.3. Models of kc

We compare EQUATER's logistic model (***lgt***) of kc against two alternative models. The first is a naive threshold based model that discards any relations that have fewer than n argument pairs. Intuitively, n can be considered the minimum number of examples to ensure that a relation has 'sufficient' data evidence to 'explain' itself. Following this model, if either r_1 or r_2 in a pair has fewer than n triples their ta and sa scores will be 0, because there is insufficient evidence in the data and hence we 'know' too little about them to evaluate similarity. Such strategy is adopted in [22]. To denote this alternative method we use $-\boldsymbol{n}$.

The second is an exponential model, denoted by ***exp*** and shown in Figure 8. We model such a curve using an exponential function shown in Equation 15, where k is a scalar that controls the speed of convergence and $|T|$ returns the number of observed examples in terms of argument pairs.

$$kc(|T|) = exp(|T|) = 1 - e^{-|T| \cdot k} \quad (15)$$

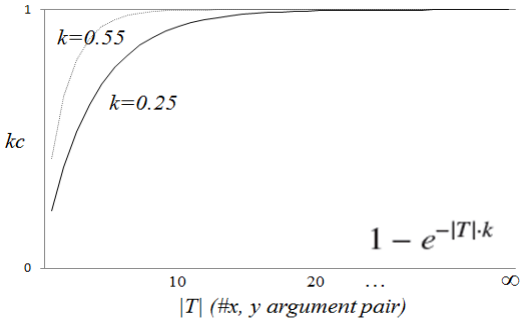

Fig. 8. The exponential function modelling knowledge confidence

For each model we need to define a parameter. For lgt, we need to define n, the number of examples

above which we obtain adequate knowledge and therefore maximum confidence. Our decision is inspired by the empirical experiences of bootstrapping learning, in which machine learns a task starting from a handful of examples. Carlson et al. [4] suggest that typically 10 to 15 examples are sufficient to bootstrap learning of relations from free form Natural Language texts. In other words, we consider 10 to 15 examples are required to 'adequately' explain the meaning of a relation. Based on this intuition, we experiment with $n = 10, 15$, and 20. Likewise this also applies to the model $-n$, for which we experiment with 10, 15 and 20 as thresholds.

We apply the same principle to the exp model. However, the scalar k is only indirectly related to the number of examples. As described before, it affects the speed of convergence, thus by setting appropriate values the knowledge confidence score returned by the function reaches its maximum at different numbers of examples. We choose $k = 0.55, 0.35$ and 0.25 that are equivalent to reaching the maximum kc of 1.0 at 10, 15 and 20 examples.

Additionally, we also compare against a version of EQUATER's similarity measure without kc, denoted by ~~kc~~, which simply combines ta and sa in their original forms. Note that this can be considered as the prototype similarity measure[12] we developed in [49].

4.4. Creation of settings

By taking different choices from the three dimensions above, we create different models for experimentation. We will denote each setting in the form of msr^{kc}_{thd}, where msr, kc, thd are variables each representing one dimension (similarity measure, kc and threshold detection respectively). Note that the variable kc only applies to EQUATER's similarity measure. Thus j_s means scoring relation pairs using the Jaccard function, then find threshold based on training data; while e^{lgt}_{jk} is EQUATER in its original form, i.e., using EQUATER's similarity measure (with the logistic model of knowledge confidence), and Jenks Natural Breaks for automatic threshold detection. Figure 9 shows a contingency chart along msr and thd dimensions, with the third dimension included as a variable kc. The output from each setting is then clustered using the same algorithm.

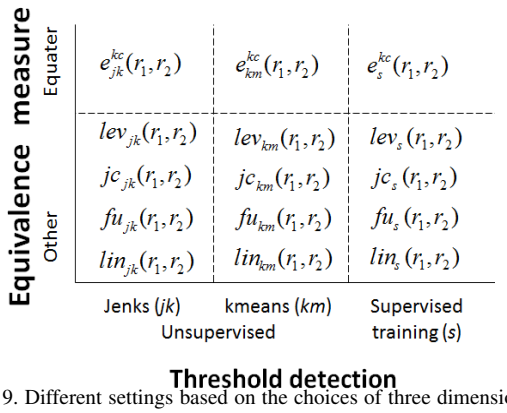

Fig. 9. Different settings based on the choices of three dimensions. *kc* is a variable whose value could be *lgt* (Equation 8), *exp* (Equation 15), or $-n$.

The Metrics we use for evaluating pair accuracy are the standard Precision, Recall and F1; and the metrics for evaluating clustering are the standard purity, inverse-purity and F1 [1].

4.5. Dataset preparation

The OAEI archives a fair number of datasets for evaluating ontology alignment systems. However, we do not use these because, as discussed before, they do not represent the particular characteristics in the LOD domain, also the number of aligned relations is very small - less than 2‰(56) of mappings found in their gold standard datasets are equivalent relations[13]. Instead, we study the problem of heterogeneous relations on DBpedia. Although DBpedia is a single dataset, we believe it as an adequate testbed of the problem for several reasons. First and foremost, multiple vocabularies are used in the dataset, including RDFS, Dublin Core[14], WGS84 Geo[15], FOAF, SKOS[16], the DBpedia ontology, original Wikipedia templates and so on. In particular, the DBpedia ontology is extremely rich in relations: the current DBpedia ontology version 3.9 covers 529 concepts and 2,333 different relations[17]. Previous researchers have already noted the prevailing issue of relation heterogeneity in the DBpedia dataset [18,14]. The majority is found between the DBpedia ontology and other vocabularies, especially the original Wikipedia templates, due to the enormous amount of relations in both vocabularies. A Wikipedia tem-

[12] Readers may notice that we dropped the 'cardinality ratio' component from the prototype, since we discovered that component may negatively affect performance.

[13] Based on the downloadable datasets as by 01-11-2013.

[14] dc=http://purl.org/dc/elements/1.1/

[15] geo=http://www.w3.org/2003/01/geo/wgs84_pos#

[16] skos=http://www.w3.org/2004/02/skos/core#

[17] http://dbpedia.org/Ontology

plate usually defines a concept and its properties[18]. When populated, they become infoboxes, which are processed to extract triples that form the backbone of the DBpedia dataset. Currently, data described by relations in the DBpedia ontology and the original Wikipedia template properties co-exist and account for a very large population in the DBpedia dataset.

The disparity between the different vocabularies in DBpedia is such a pressing issue that the team has dedicated particular effort to address it, which is known as the DBpedia mappings portal. The DBpedia mappings portal is a website that invites collaborative effort to create mappings between certain structured content on Wikipedia to the manually curated DBpedia ontology. One task is mapping Wikipedia templates to concepts in the DBpedia ontology, and then mapping properties in the templates to relations of mapped concepts. Such mappings are useful for both tidying up existing DBpedia dataset and future data publication on DBpedia. On the one hand, they can improve information retrieval from DBpedia, as we already showed in [2,48]. On the other hand, the DBpedia Extraction Framework can use such mappings in future to homogenize information extracted from Wikipedia before generating structured information in RDF. It is known that manually creating such mappings requires significant work, and as a result, as by November 2013, less than 55% of mappings between Wikipedia template properties and relations in the DBpedia ontology are complete[19].

Further, DBpedia is the most representative LOD dataset as it is predominantly used in research concerning Linked Data. It is also currently the largest hub connecting multiple datasets in the LOD domain, thus a large majority of LOD datasets can benefit from reducing heterogeneity on DBpedia. All these facts make DBpedia an interesting and reasonable testbed of the problem.

We collected three datasets for experiments. The first dataset is created based on the mappings published on the DBpedia mappings portal. We processed the DBpedia mappings Webpages as by 30 Sep 2013 and created a dataset containing 203 DBpedia concepts. Each concept has a page that defines the mapping from a Wikipedia template to a DBpedia concept, and lists a number of mapping pairs from template properties to the relations of the corresponding concept in the DBpedia ontology. We extracted a total of 5388 mappings and use them as gold standard (*dbpm*). However, there are three issues with this dataset. First, the community portal focuses on mapping the DBpedia ontology with the original Wikipedia templates. Therefore, mappings between the DBpedia ontology and other vocabularies are rare. Second, due to the ongoing nature of the mapping task, the dataset is largely incomplete. Therefore, we only use this dataset for evaluating recall. Third, it has been noticed that the mappings created are not always strictly 'equivalence'. Some infrequent mappings such as 'broader-than' have also been included. Overall the *dbpm* dataset should not be considered a perfect gold standard for the task.

For this reason, we manually created a dataset based on 40 DBpedia (DBpedia ontology version 3.8) and YAGO[20] concepts. The choices of such concepts are based on the QALD1 question answering dataset[21] for Linked Data. For each concept, we query the DBpedia SPARQL endpoint using the following query template to retrieve all triples related to the concept[22].

```
SELECT * WHERE {
?s a <[Concept_URI]> .
?s ?p ?o .
}
```

Next, we build a set P containing unordered pairs of predicates from these triples and consider them as candidate relation pairs for the concept. We also use a *stop list* of relation URIs to filter meaningless relations that usually describes Wikipedia meta-level information, e.g., *dbpp:wikiPageID, dbpp:wikiPageUsesTemplate*. Each of the measures listed in Section 4.1 is then applied to compute similarity of the pairs in this set and may produce either a zero or non-zero score. We then create a set cP concatenating the pairs with non-zero scores by any of the measures, and ask human annotators to annotate cP. Note that $cP \subset P$ and may not exclusively contain all true positives of the concept since there can be equivalent pairs of relations obtaining zero similarity score by all the measures. However, we believe it is a reasonable approximation. Moreover

[18] Not in formal ontology terms, but rather a Wikipedia terminology.

[19] http://mappings.dbpedia.org/server/statistics/en/, visited on 01-11-2013

[20] http://www.mpi-inf.mpg.de/yago-naga/yago/

[21] http://greententacle.techfak.uni-bielefeld.de/~cunger/qald1/evaluation/dbpedia-test.xml

[22] Note that DBpedia by default returns a maximum of 50,000 triples per query. We did not incrementally build the exhaustive result set for each concept since we believe the data size is sufficient for experiment purposes.

it would be extremely expensive to annotate the set of all pairs completely.

The data is annotated by four computer scientists and the annotation took three weeks, where one week was spent on creating guidelines. Annotators can also query DBpedia for triples containing the relation to assist their interpretation. However, a notable number of relations are still incomprehensible. These often have peculiar names and are rarely used (e.g., *dbpp:n, dbpp:wikt*). Pairs containing such relations cannot be annotated and are ignored in evaluation. On average, it takes 0.5 to 1 hour to annotate one concept. We measured inter-annotator-agreement using a sample dataset based on the method by [20], and the IAA is 0.8. The dataset is then randomly split into a development set (*dev*) containing 10 concepts for developing our measure and a test set (*test*) containing 30 concepts for evaluation.

To encourage comparative studies in the future, we publish all datasets and associated resources used in this study[23]. The statistics of the three datasets are shown in Table 2. Figure 10 shows the ranges of the percentage of true positives in the *dev* and *test* datasets. To our knowledge, this is by far the largest annotated dataset for evaluating relation alignment in the LOD domain.

4.6. General process

Given a concept from any of the three datasets, we query the DBpedia SPARQL endpoint to obtain a triple dataset and create candidate set of relation pairs following the same procedure described above. For each setting created according to Section 4.4, we apply the methods to the triple dataset to (1) compute similarity score for each relation pair and determine if they should be considered truly equivalent based on the score, and (2) create clusters of equivalent relations for the concept.

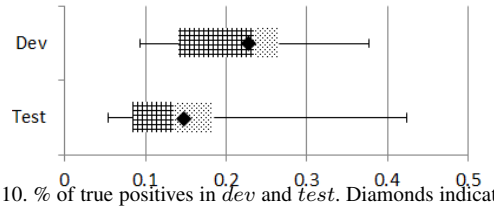

Fig. 10. % of true positives in dev and $test$. Diamonds indicate the mean.

[23]http://staffwww.dcs.shef.ac.uk/people/Z.Zhang/resources/jws2014/data_release.zip. The cached DBpedia query results are also released.

	Dev	Test	dbpm
Concepts	10	30	203
Relation pairs (P.)	2316	6657	5388
True positive P.	473	868	-
P. with incomprehensible relations (I.R.)	316	549	-
% of triples with I.R.	0.2%	0.2%	-
Schemata in datasets	dbo, dbpp, rdfs, skos, dc, geo, foaf		

Table 2
Dataset statistics

The output from (1) is then evaluated against the three gold standard datasets described above. To evaluate clustering, we derived gold standard clusters using the three pair-equivalence gold standards by assuming equivalence transitivity, i.e., if r_1 is equivalent to r_2, which is equivalent to r_3 in the gold standard then the three relations are grouped in a single cluster. We only consider clusters of positive pairs as the larger amount of negative pairs (which results in a large number of single-element clusters) may bias the evaluation.

5. Results and discussion

5.1. Difficulty of the task

Annotating relation equivalence is a non-trivial task. The annotation process costs many person-days with a resulting average IAA of 0.8, while the lowest bound is 0.68 and the highest is 0.87. It has been found that LOD datasets are characterized by a notable degree of noise. As Table 2 shows, about 8 to 14% of pairs contain incomprehensible relations. Such relations have peculiar names (e.g., *dbpp:v, dbpp:trW* of *dbo:University*) and ambiguous names (e.g., *dbpp:law, dbpp:bio* of *dbo:University*). They are undocumented and have little usage in data, which makes them difficult to interpret. Moreover, there is also a high degree of inconsistent usage of relations. A typical example is *dbo:railwayPlatforms* of *dbo:Station*. It is used to represent the number of platforms in a station, but also the types of platforms in a station. These findings are in line with [14].

Table 2 and Figure 10 both show that the dataset is overwhelmed by negative examples. On average, less than 25% of non-zero similarity pairs are true positives and in extreme cases this drops to less than 6% (e.g., 20 out of 370 relation pairs of *yago:EuropeanCountries*

	lev	jc	fu	lin
t	0.43	0.07	0.1	0.65
min	0.06	0.01	6×10^{-6}	0.14
max	0.77	0.17	0.31	1.0

Table 3

Optimal thresholds t for each baseline similarity measures derived from dev

	t	*min*	*max*
lgt, n=10	0.24	0.06	0.62
lgt, n=15	0.22	0.05	0.62
lgt, n=20	0.2	0.06	0.39
exp, k=0.55	0.32	0.06	0.62
exp, k=0.35	0.31	0.06	0.62
exp, k=0.25	0.33	0.11	0.62
$-n$, n=10	0.29	0.06	0.62
$-n$, n=15	0.28	0.07	0.59
$-n$, n=20	0.28	0.07	0.59

Table 4

Optimal thresholds (t) for different variants of the similarity measure of EQUATER derived from dev

are true positive). These findings suggest that finding equivalent relations on Linked Data is indeed a challenging task.

Table 3 shows the learned thresholds for each of the baseline similarity measures based on the dev data, and Table 4 shows the learned thresholds for different variants of EQUATER by replacing its kc component variables. In any case, the learned thresholds span across a wide range, suggesting that the optimal thresholds to decide equivalence are indeed data-specific, and finding these values can be difficult.

5.2. EQUATER performance

In Table 5 we show the results of EQUATER on the three datasets with varying n in the lgt knowledge confidence function. All figures are averages over all concepts in a dataset. Figure 11 shows the ranges of performance scores for different concepts in each dataset. Table 6 shows example clusters of equivalent relations discovered for different concepts. It shows that EQUATER manages to discover alignment between multiple schemata used in DBpedia.

On average, EQUATER obtains 0.65∼0.67 F1 in predicting pair equivalence on dev and 0.59∼0.61 F1 on $test$. These translate to 0.74 and 0.70 clustering accuracy on each dataset respectively. For $dbpm$, we obtain a recall between 0.66 and 0.68 for pair equivalence and 0.7 and 0.72 for clustering. It is interesting to note that EQUATER appears to be insensitive to the varying values of n. This stability is a desirable feature since it may be unnecessary to tune the model and therefore, the method is less prone to overfitting. This also confirms the hypothetical analogy between the amount of seed data needed for bootstrap relation learning and the amount of examples needed to obtain maximum knowledge confidence in EQUATER.

Figure 11 shows that the performance of EQUATER can vary depending on specific concepts. To understand the errors, we randomly sampled 100 false pos-

n of lgt	10	15	20
Pair equivalence			
dev, F1	0.67	0.66	0.65
test, F1	0.61	0.60	0.59
dbpm, R	0.68	0.66	0.66
Clustering			
dev, F1	0.74	0.74	0.74
test, F1	0.70	0.70	0.70
dbpm, R	0.72	0.70	0.70

Table 5

Results of EQUATER on all datasets. R - Recall

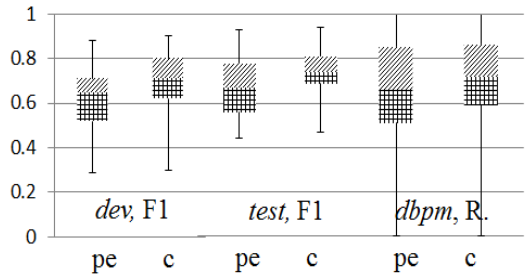

Fig. 11. Performance ranges on a per-concept basis for dev, $test$ and $dbpm$. R - Recall, pe - pair equivalence, c - clustering

Concept	Example cluster
dbo:Actor	dbpp:birthPlace, dbo:birthPlace, dbpp:placeOfBirth
dbo:Book	dbpp:name, foaf:name, dbpp:titleOrig, rdfs:label
dbo:Company	dbpp:website, foaf:website dbpp:homepage, dbpp:url

Table 6

Examples clusters of equivalent relations.

itive and 100 false negative examples from the $test$ dataset, and 200 false negative examples from the $dbpm$ dataset, then manually analyzed and divided them into several types[24]. The prevalence of each type is shown in Table 7.

Error Type	Prevalence
False Positives	
Semantically similar	52.4
Low variability	29.1
Entailment	14.6
Arguable gold standard	3.88
False Negatives	
Object representation	25.1
Different datatype	24.7
Noisy relation	19.4
Different lexicalisations	11.7
Sparsity	10.5
Limitation of method	5.67
Arguable gold standard	2.83

Table 7
Relative prevalence of error types.

5.2.1. False positives

The first main source of errors are due to high degree of **semantic similarity**: e.g., *dbpp:residence* and *dbpp:birthPlace* of *dbo:TennisPlayer* are highly semantically similar but non-equivalent. The second type of errors is due to **low variability** in the objects of a relation: semantically dissimilar relations can have the same datatype and have many overlapping values by coincidence. The overlap is caused by some relations having a limited range of object values, which is especially typical for relations with boolean datatype because they only have two possible values. The third type of errors is **entailment**, e.g., for *dbo:EuropeanCountries dbpp:officialLanguage* entails *dbo:language* because official languages of a country are a subset of languages spoken in a country. These could be considered as cases of subsumption, which accounts for less than 15%. Finally, some of the errors are **arguably due to imperfect gold standard**, as analysers sometimes disagree with the annotations (see Table 8).

5.2.2. False negatives

The first type of common errors is due to **representation of objects**. For instance, for *dbo:AmericanFootballPlayer*, *dbo:team* are associated with mostly resource URIs (e.g., 'dbr:Detroit_Lions') while *dbpp:teams* are mostly associated with lexicalization of literal objects (e.g.,'* Detroit Lions') that are typically names of the resources. The second type is due to **different datatypes**, e.g., for *dbo:Building*, *dbpp:startDate* typically have literal objects indicating years, while *dbo:buildingStartDate* usually has precisely literal date values as objects. Thirdly, the **lexicalization of objects can be different**. An example for this category is *dbpp:dialCode* and *dbo:areaCode* of *dbo:Settlement*, the objects of the two relations are represented in three different ways, e.g. '0044', '+44', '44'. Many false negatives are due to **sparsity**: e.g., *dbpp:oEnd* and *dbo:originalEndPoint* of *dbo:Canal* have in total only 2 triples. There are also **noisy relations**, whose lexicalization appears to be inconsistent with how it is used. Usually the lexicalization is ambiguous, such as the *dbo:railwayPlatforms* example discussed before. Some errors are simply due to the **limitation of our method**, i.e., our method still fails to identify equivalence even if sufficient, quality data are available, possibly due to inappropriate automatic threshold selection. And further, **arguable gold standard** also exist (e.g., *dbpp:champions* and *dbo:teams* of *dbo:SoccerLeague* are mapped to each other in the $dbpm$ dataset.

We then also manually inspected some worst performing concepts in the *dbpm* dataset, and noticed that some of them are due to extremely small gold standard. For example, *dbo:SportsTeamMember* and *dbo:Monument* have only 3 true positives each in their gold standard and as a result, EQUATER scored 0 in recall. However, we believe that these gold standards are largely incomplete. For example, we consider most proposals by EQUATER in Table 8 to be correct.

Some of the error types mentioned above could be rectified by modifying EQUATER. For example, we could combine string similarity metrics, which may help errors due to **representation of objects** and **different lexicalization of objects**. Regular expressions could be used to parse values in order to match data at semantic level, e.g., for dates, weights, and lengths. These could be useful to solve errors due to **different datatypes**. Other error groups are much harder to prevent: even annotators often struggled to distinguish between semantically similar and equivalent relations or to understand what a relation is supposed to mean.

[24] Analysis based on the DBpedia SPARQL service as by 31-10-2013. Inconsistency should be anticipated if different versions of datasets are used.

Pair equivalence									
	dev F1			*test* F1			*dbpm* R.		
thd / msr	*jk*	*km*	*s*	*jk*	*km*	*s*	*jk*	*km*	*s*
lev	0.16∼18	0.16∼18	0.17∼19	0.12∼14	0.12∼14	0.13∼15	0.07∼08	0.08∼09	0.07∼08
f	0.18∼20	0.20∼22	0.09∼11	0.20∼21	**0.22∼23**	0.10∼11	0.21∼23	0.24∼26	0.03∼04
lin	0.27∼29	**0.29∼31**	0.28∼30	0.19∼21	0.19∼21	0.19∼21	**0.39∼40**	0.37∼38	0.38∼39
jc	0.09∼11	0.11∼12	0.01∼03	0.10∼11	0.12∼13	0.07∼08	0.09∼10	0.10∼12	*-0.05∼-0.04*
Clustering									
	dev F1			*test* F1			*dbpm* R.		
thd / msr	*jk*	*km*	*s*	*jk*	*km*	*s*	*jk*	*km*	*s*
lev	0.35	0.36	0.36	0.40	0.41	0.39	0.02∼04	0.04∼06	0.03∼05
f	0.15∼16	0.17∼18	0.06∼07	0.23	0.25	0.11	0.21∼23	0.24∼26	0.01∼03
lin	0.45	**0.47**	**0.47**	0.41	**0.42**	**0.42**	0.37∼39	0.37∼39	**0.39∼0.41**
jc	0.06∼07	0.07∼08	0.00∼01	0.14∼15	0.17	0.06	0.08∼10	0.09∼11	*-0.07∼-0.05*

Table 9

Improvement of EQUATER over different baselines. The highest improvements on each dataset are highlighted in **bold**. Negative improvements are highlighted in *italic*.

r_1	r_2	$\#x, y$ argument pairs
dbo:synonym	*dbp:otherName*	6
rdfs:label	*foaf:name*	10
rdfs:label	*dbp:name*	10
rdfs:comment	*dbo:abstract*	41
dbp:material	*dbo:material*	10
dbp:city	*dbo:city*	5

Table 8

The equivalent relations for *dbo:Monument* proposed by EQUATER but considered false positive according to the gold standard.

5.3. EQUATER v.s. baseline

Next, in Table 9 we show the improvement of EQUATER over different models that use a baseline similarity measure. Since the performance of EQUATER depends on the parameter n in the similarity measure, we show the ranges between minimum and maximum improvement due to the choice of n.

It is clear from Table 9 that EQUATER (unsupervised) significantly outperforms most baseline models, either supervised or unsupervised. Exceptions are noted against jc_s in the clustering task on the dev dataset, where EQUATER achieves comparable results; and on the $dbpm$ dataset in both pair equivalence and clustering tasks, where EQUATER underperforms jc_s in terms of recall. However, as discussed before the $dbpm$ gold standard has many issues; furthermore, we are unable to evaluate precision on this dataset while results on the dev and $test$ sets suggest EQUATER has more balanced performance. The relatively larger improvement over unsupervised baselines than over supervised baselines may suggest that the scores produced by EQUATER may exhibit a more 'separable' pattern (e.g., like Figure 6) of distribution for unsupervised threshold detection.

Figures 12a and 12b compares the balance between precision and recall of EQUATER against baselines on the dev and $test$ datasets. For EQUATER we use three different shapes to represent models with different n values in lgt; for baseline models we use different shapes to represent different similarity measures and different colours (black, white and grey) to represent different thd choices. It is clear that EQUATER always outperforms any baselines in terms of precision, and also finds the best balance between precision and recall thus resulting in the highest F1.

Interesting to note is the inconsistent performance of string similarity baselines (lev_{jk}, lev_{km}, lev_s) in pair equivalence experiments and clustering experiments. While in pair equivalence experiments they obtain between 0.45 and 0.5 F1 (second best among baselines) on both dev and $test$ with arguably balanced precision and recall, in clustering experiments the fig-

ures sharply drop to 0.3∼0.4 (second worst among baselines) skewed towards very high recall and very low precision. This suggests that the string similarity scores are non-separable by clustering algorithms, creating larger clusters that favour recall but precision.

Very similar pattern is also noted for the semantic similarity baselines (lin_{jk}, lin_{km}, lin_s). In fact, semantic similarity and string similarity baselines generally obtain much worse results than other baselines that belong to extensional matchers, a strong indication that the latter are better fit for aligning relations in the LOD domain. This can be partially attributed to the fact that the relation URIs can be very noisy and many do not comply with naming conventions and rules (e.g., 'birthplace' instead of 'birthPlace').

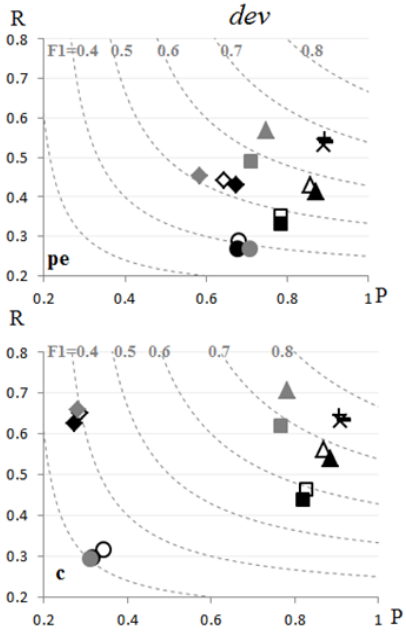

Fig. 12a. Balance between precision and recall for EQUATER and baselines on dev. pe - pair equivalence, c - clustering. The dotted lines are F1 references.

5.4. Variations of EQUATER components

In this section, we compare EQUATER against several alternative designs based on the alternative choices of knowledge confidence (kc) functions and threshold detection (thd) methods. We pair different kc models described in Section 4.3 with different threshold detection methods described in Section 4.2 to create variants of EQUATER and compare them against the original EQUATER method. In addition, we also create

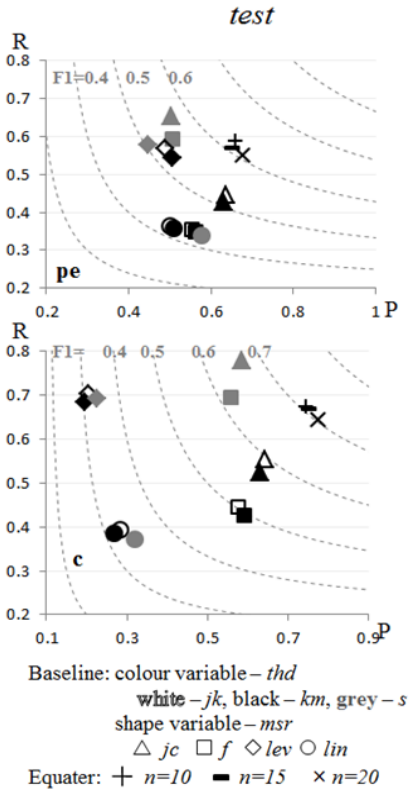

Fig. 12b. Balance between precision and recall for EQUATER and baselines on $test$.

the similarity measure $e^{\not{kc}}$, which only takes ta and sa without the knowledge confidence factor. Combined with different choices of thd we obtain models that represent our earlier prototype in [49]. Moreover, we also select jc as the best performing baseline similarity measure and use corresponding baseline settings (jc_{jk}, jc_{km}, jc_s) as comparative references.

5.4.1. Alternative kc models

Figure 13a compares variations of EQUATER by alternating kc functions under each thd method. Since each of the functions lgt, exp and $-n$ requires a parameter to be set, we show the ranges of performance gained with different settings of their parameter. These are represented as black caps on top of each bar. The bigger the cap, the wider the range between the minimum and the maximum performance obtainable by tuning these parameters. Firstly, under the same chosen threshold detection method, settings with $\not{kc}$ outperform the best baseline in most cases. This suggests that ta and sa are indeed more effective indicators of relation equivalence than other metrics, and also suggests that the issue of unbalanced populations of

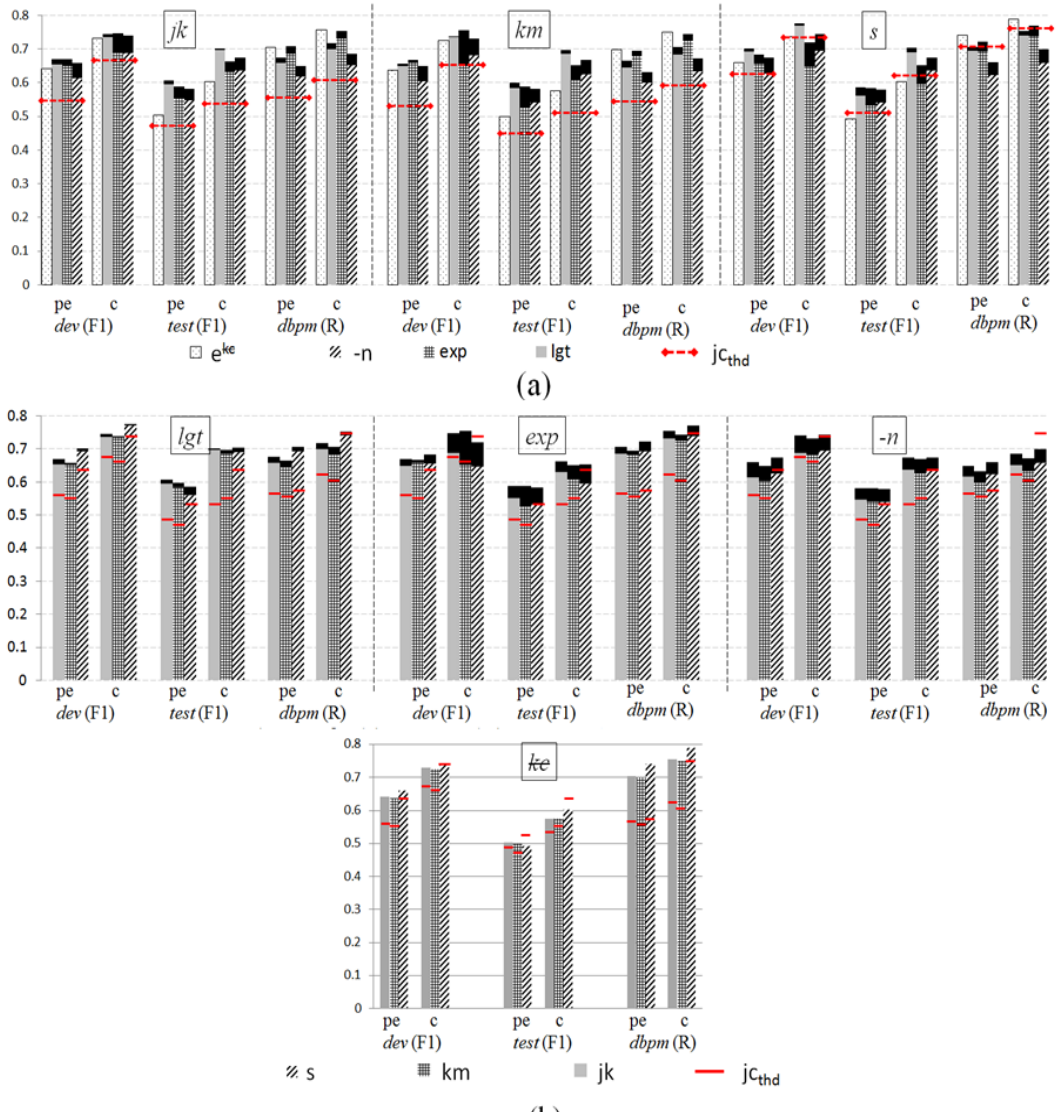

Fig. 13. Comparing variations of EQUATER by (a) alternating kc functions under each threshold detection method, and (b) alternating thd methods under each knowledge confidence function (incl. without kc).

schemata in the LOD domain is very common. Secondly, we can see that the accuracy of EQUATER as measured by F1 does benefit from the integration of kc functions; the changes are also substantial on $test$ set. While combining results on the $dbpm$ set, it seems that kc functions may trade off recall for precision to achieve overall higher F1. Thirdly, in terms of the three kc functions, the performance given by the exp and $-n$ model appears to be volatile since changing their parameters caused considerable variation of performance in most cases. This also caused several variants of EQUATER to underperform the baseline with the same thd setting.

By analyzing the precision and recall trade-off for different kc functions, it shows that without kc, the similarity measure of EQUATER tends to favour high-recall but perhaps lose too much precision. Any kc function thus has the effect of re-balancing towards precision. Among the three, the exp function generally favours recall over precision, the threshold based model favours precision over recall, while the lgt function finds the best balance. Details of this part of analysis can be found in A.

5.4.2. *Alternative thd methods*

Figure 13b is a re-arranged view of Figure 13a, in the way that it compares variations of EQUATER

by alternating the thd methods under each kc function. This gives a better view for comparing different choices of thd. Generally it appears that regardless of the kc function, the jk method has slight advantage over km. The unsupervised jk variants of EQUATER also obtain close performance to their supervised counterparts in many cases; and even best the performance on the $test$ set (excluding ~~kc~~). This suggests Jenks Natural Breaks an effective method for automatic threshold detection for EQUATER.

5.4.3. Limitations of EQUATER

The current version of EQUATER is limited in a number of ways. First and foremost, being an extensional matcher, it requires relations to have shared instances to work. This is usually a reasonable requirement for individual dataset, and hence experiments based on DBpedia have shown it to be very effective. However, in a cross-dataset context, concepts and instances will have to be aligned first in order to apply EQUATER. This is because often, different datasets use different URIs to refer to the same entities; as a result, counting overlap of a relation's arguments will have to go beyond syntactic level.

A basic and simplistic solution could be a preprocess that maps concepts and instances from different datasets using existing 'sameAs' mappings, as done by Parundekar et al. [40] and Zhao et al. [52]. Unfortunately, when such mappings are unavailable, EQUATER may require other methods to firstly align concepts and instances in order to work effectively for cross-dataset settings. Therefore, we consider the second major limitation of EQUATER as it being a partial ontology alignment method addressing a specific but practical issue - aligning relations only. Ideally, EQUATER should be extended to iteratively align relations based on aligned concepts and instances and vice-versa.

Despite these limitations, we consider EQUATER a valuable contribution to the literature as it targets specifically at the gap in related work, and lessons learned could be useful for future development.

6. Conclusions

This article explored the problem of aligning heterogeneous resources in LOD datasets. Heterogeneity decreases the quality of the data and may eventually hamper its usability over large scale. It is a major research problem concerning the Semantic Web community and significant effort has been made to address this problem in the area of ontology alignment. While most work studied mapping concepts and individuals, relation heterogeneity in LOD datasets is becoming an increasingly pressing issue but still remains much less studied. The annotation practice undertaken in this work has shown that the task is even challenging to humans.

This article makes particular contribution to this problem by introducing EQUATER - a domain- and language-independent and unsupervised method to align relations based on their shared instances. Currently, EQUATER fits best with aligning relations from different schemata used in a single Linked Dataset, a practical problem that has emerged with the increasing collaborative effort in creating very large Linked Datasets. It can potentially be used in cross-dataset settings, provided that concepts and instances across the datasets are aligned to ensure relations have shared instances.

A series of experiments have been designed to thoroughly evaluate EQUATER in two tasks: predicting relation pair equivalence and discovering clusters of equivalent relations. These experiments have confirmed the advantage of EQUATER: compared to baseline models including both supervised and unsupervised versions, it makes significant improvement in terms of F1 measure, and always scores the highest precision. Compared to different variants of EQUATER, the logistic model of knowledge confidence achieves the best scores in most cases and is seen to give stable performance regardless of its parameter setting, while the alternatives suffer from higher degree of volatility that occasionally causes them to underperform baselines. The Jenks Natural Breaks method for automatic threshold detection also proves to have slight advantage than the k-means alternative, and even outperformed the supervised method on the $test$ set. Although EQUATER does not achieve the best recall on the $dbpm$ dataset, we believe its results are still encouraging and that it can achieve the most balanced performance had we been able to evaluate precision. Overall we believe that it may potentially speed up the practical mapping task currently concerning the DBpedia community.

As future work, we will explore methods to address the previously discussed limitations of EQUATER.

Acknowledgement Part of this research has been sponsored by the EPSRC funded project LODIE:

Linked Open Data for Information Extraction, EP/J019488/1

References

[1] Javier Artiles, Julio Gonzalo, and Satoshi Sekine. The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval '07, pages 64–69, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics.

[2] E. Blomqvist, Z. Zhang, A. Gentile, I. Augenstein, and F. Ciravegna. Statistical knowledge patterns for characterizing linked data. In *Workshop on Ontology and Semantic Web Patterns (4th edition) - WOP2013 at ISWC2013*, 2013.

[3] T. Caliński and J. Harabasz. A dendrite method for cluster analysis. *Cluster Analysis Communications in Statistics*, 3(1):1–27, 1974.

[4] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr., and T.M. Mitchell. Toward an architecture for never-ending language learning. In *Proceedings of the Conference on Artificial Intelligence (AAAI)*, pages 1306–1313. AAAI Press, 2010.

[5] Michelle Cheatham and Pascal Hitzler. String similarity metrics for ontology alignment. In *Proceedings of the 12th International Semantic Web Conference*, ISWC'13, pages 294–309, Berlin, Heidelberg, 2013. Springer-Verlag.

[6] Isabel F. Cruz, Matteo Palmonari, Federico Caimi, and Cosmin Stroe. Towards 'on the go' matching of linked open data ontologies. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, and Bin He, editors, *IJCAI Workshop Discovering Meaning On the Go in Large and Heterogeneous Data (LHD-11)*, Barcelona, Spain, July 2011.

[7] Isabel F. Cruz, Matteo Palmonari, Federico Caimi, and Cosmin Stroe. Building linked ontologies with high precision using subclass mapping discovery. *Artificial Intelligence Review*, 40(2):127–145, aug 2013.

[8] Isabel F. Cruz, Cosmin Stroe, Michele Caci, Federico Caimi, Matteo Palmonari, Flavio Palandri Antonelli, and Ulas C. Keles. Using agreementmaker to align ontologies for oaei 2010. In *Proceedings of ISWC International Workshop on Ontology Matching (OM)*, CEUR Workshop Proceedings, 2010.

[9] Russell Dewey. Chapter 7: Cognition. *Psychology: An Introduction*, 2007.

[10] Hong-Hai Do and Erhard Rahm. Coma: A system for flexible combination of schema matching approaches. In *Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB '02, pages 610–621. VLDB Endowment, 2002.

[11] AnHai Doan, Pedro Domingos, and Alon Y. Halevy. Reconciling schemas of disparate data sources: A machine-learning approach. *SIGMOD Rec.*, 30(2):509–520, May 2001.

[12] Songyun Duan, Achille Fokoue, Oktie Hassanzadeh, Anastasios Kementsietsidis, Kavitha Srinivas, and Michael J. Ward. Instance-based matching of large ontologies using locality-sensitive hashing. In *Proceedings of the 11th international conference on The Semantic Web - Volume Part I*, ISWC'12, pages 49–64, Berlin, Heidelberg, 2012. Springer-Verlag.

[13] J. Euzenat and P Shvaiko. *Ontology Matchin*. Springer, 2007.

[14] Linyun Fu, Haofen Wang, Wei Jin, and Yong Yu. Towards better understanding and utilizing relations in dbpedia. *Web Intelligence and Agent Systems*, 10(3):291–303, 2012.

[15] Anna Lisa Gentile, Ziqi Zhang, Isabelle Augenstein, and Fabio Ciravegna. Unsupervised wrapper induction using linked data. In *Proceedings of the seventh international conference on Knowledge capture*, K-CAP '13, pages 41–48, New York, NY, USA, 2013. ACM.

[16] Bernardo Cuenca Grau, Zlatan Dragisic, Kai Eckert, Jeréome Euzenat, Alo Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jimenez-Ruiz11, Andreas Oskar Kempf, Patrick Lambrix, Christian Meilicke, Andriy Nikolov, Heiko Paulheim, Dominique Ritze, Francçois Scharffe, Pavel Shvaiko, Cássia Trojahn, and Ondřej Zamazal. Preliminary results of the ontology alignment evaluation initiative 2013. In *The Eighth International Workshop on Ontology Matching at ISWC2013*, OM'13, 2013.

[17] Toni Gruetze, Christoph Böhm, and Felix Naumann. Holistic and scalable ontology alignment for linked open data. In Christian Bizer, Tom Heath, Tim Berners-Lee, and Michael Hausenblas, editors, *Proceedings of the 5th Linked Data on the Web (LDOW) Workshop at the 21th International World Wide Web Conference (WWW)*, CEUR Workshop Proceedings. CEUR-WS.org, 2012.

[18] Lushan Han, Tim Finin, and Anupam Joshi. Gorelations: an intuitive query system for dbpedia. In *Proceedings of the 2011 joint international conference on The Semantic Web*, JIST'11, pages 334–341, Berlin, Heidelberg, 2012. Springer-Verlag.

[19] Sven Hertlingand and Heiko Paulheim. Wikimatch - using wikipedia for ontology matching. In *proceedings Ontology Matching : Proceedings of the 7th International Workshop on Ontology Matching (OM-2012) collocated with the 11th International Semantic Web Conference (ISWC-2012)*, ISWC'12, pages 37–48, 2012.

[20] George Hripcsak and Adam Rothschild. Agreement, the f-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, 12:296–298, 2005.

[21] Wei Hu, Yuzhong Qu, and Gong Cheng. Matching large ontologies: A divide-and-conquer approach. *Data Knowledg. Engineering*, 67(1):140–160, October 2008.

[22] Antoine Isaac, Lourens Van Der Meij, Stefan Schlobach, and Shenghui Wang. An empirical study of instance-based ontology matching. In *Proceedings of the 6th international The semantic web and 2nd Asian conference on Asian semantic web conference*, ISWC'07/ASWC'07, pages 253–266, Berlin, Heidelberg, 2007. Springer-Verlag.

[23] Prateek Jain, Pascal Hitzler, Amit P. Sheth, Kunal Verma, and Peter Z. Yeh. Ontology alignment for linked open data. In *Proceedings of the 9th International Semantic Web Conference*, ISWC2010, pages 402–417, Berlin, Heidelberg, 2010. Springer-Verlag.

[24] Prateek Jain, Peter Z. Yeh, Kunal Verma, Reymonrod G. Vasquez, Mariana Damova, Pascal Hitzler, and Amit P. Sheth. Contextual ontology alignment of lod with an upper ontology: A case study with proton. In *Proceedings of the 8th Extended Semantic Web Conference on The Semantic Web: Research and Applications - Volume Part I*, ESWC'11, pages 80–92, Berlin, Heidelberg, 2011. Springer-Verlag.

[25] Yves R. Jean-Mary, E. Patrick Shironoshita, and Mansur R. Kabuka. Ontology matching with semantic verification. *Web Semantics*, 7(3):235–251, September 2009.

[26] George Jenks. The data model concept in statistical mapping. *International Yearbook of Cartography*, 7:186–190, 1967.

[27] Jaewoo Kang and Jeffrey F. Naughton. On schema matching with opaque column names and data values. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD '03, pages 205–216, New York, NY, USA, 2003. ACM.

[28] Patrick Lambrix and He Tan. Sambo-a system for aligning and merging biomedical ontologies. *Web Semantics*, 4(3):196–206, September 2006.

[29] Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. RiMOM: A Dynamic Multistrategy Ontology Alignment Framework. *IEEE Transactions on Knowledge and Data Engineering*, 21(8):1218–1232, August 2009.

[30] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. Annotating and searching web tables using entities, types and relationships. *Proceedings of the VLDB Endowment*, 3(1-2):1338–1347, 2010.

[31] D. Lin. An Information-Theoretic Definition of Similarity. In *Proceedings of the 5th International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, 1998b. Morgan Kaufmann Publishers Inc.

[32] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[33] Jayant Madhavan, Philip A. Bernstein, AnHai Doan, and Alon Halevy. Corpus-based schema matching. In *Proceedings of the 21st International Conference on Data Engineering*, ICDE '05, pages 57–68, Washington, DC, USA, 2005. IEEE Computer Society.

[34] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 49–58, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[35] Fionn Murtagh. Multidimensional clustering algorithm. *in COMPSTAT Lectures 4. Wuerzburg: Physica-Velag*, 1985.

[36] Miklos Nagy, Maria Vargas-Vera, and Enrico Motta. Multi agent ontology mapping framework in the aqua question answering system. In *Proceedings of the 4th Mexican International Conference on Advances in Artificial Intelligence*, MICAI'05, pages 70–79, Berlin, Heidelberg, 2005. Springer-Verlag.

[37] Miklos Nagy, Maria Vargas-Vera, and Enrico Motta. Dssim — managing uncertainty on the semantic web. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, and Bin He, editors, *The 2nd International Workshop on Ontology Matching (OM-2007)*, volume 304 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[38] DuyHoa Ngo, Zohra Bellahsene, and Remi Coletta. A generic approach for combining linguistic and context profile metrics in ontology matching. In *Proceedings of the 2011th Confederated International Conference on On the Move to Meaningful Internet Systems - Volume Part II*, OTM'11, pages 800–807, Berlin, Heidelberg, 2011. Springer-Verlag.

[39] Andriy Nikolov, Victoria Uren, Enrico Motta, and Anne Roeck. Overcoming schema heterogeneity between linked semantic repositories to improve coreference resolution. In *Proceedings of the 4th Asian Conference on The Semantic Web*, ASWC '09, pages 332–346, Berlin, Heidelberg, 2009. Springer-Verlag.

[40] Rahul Parundekar, Craig A. Knoblock, and José Luis Ambite. Linking and building ontologies of linked data. In *Proceedings of the 9th International Semantic Web Conference on The Semantic Web - Volume Part I*, ISWC'10, pages 598–614, Berlin, Heidelberg, 2010. Springer-Verlag.

[41] Shvaiko Pavel and Jerome Euzenat. Ontology matching: State of the art and future challenges. *IEEE Trans. on Knowl. and Data Eng.*, 25(1):158–176, January 2013.

[42] Balthasar A. C. Schopman, Shenghui Wang, Antoine Isaac, and Stefan Schlobach. Instance-based ontology matching by instance enrichment. *J. Data Semantics*, 1(4):219–236, 2012.

[43] Md. Hanif Seddiqui and Masaki Aono. An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. *Web Semantics*, 7(4):344–356, December 2009.

[44] G. Shafer. *A Math. Theory of Evidence*. Princeton Univ. Press, 1976.

[45] Kristian Slabbekoorn, Laura Hollink, and Geert-Jan Houben. Domain-aware ontology matching. In *Proceedings of the 11th International Conference on The Semantic Web - Volume Part I*, ISWC'12, pages 542–558, Berlin, Heidelberg, 2012. Springer-Verlag.

[46] Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. Paris: Probabilistic alignment of relations, instances, and schema. *Proc. VLDB Endow.*, 5(3):157–168, November 2011.

[47] Johanna Völker and Mathias Niepert. Statistical schema induction. In *Proceedings of the 8th Extended Semantic Web Conference on The Semantic Web: Research and Applications - Volume Part I*, ESWC'11, pages 124–138, Berlin, Heidelberg, 2011. Springer-Verlag.

[48] Z. Zhang, A. Gentile, E. Blomqvist, I. Augenstein, and F. Ciravegna. Statistical knowledge patterns: Identifying synonymous relations in large linked datasets. In *The 12th International Semantic Web Conference and the 1st Australasian Semantic Web Conference*, Sydney, Australia, 2013.

[49] Ziqi Zhang, Anna Lisa Gentile, Isabelle Augenstein, Eva Blomqvist, and Fabio Ciravegna. Mining equivalent relations from linked data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics - Short Papers (ACL Short Papers 2013)*, ACL'13. ACL, 2013.

[50] Ziqi Zhang, Anna Lisa Gentile, and Fabio Ciravegna. Recent advances in methods of lexical semantic relatedness - a survey. *Natural Language Engineering*, FirstView:1–69, 4 2012.

[51] Lihua Zhao and Ryutaro Ichise. Graph-based ontology analysis in the linked open data. In *Proceedings of the 8th International Conference on Semantic Systems*, I-SEMANTICS '12, pages 56–63, New York, NY, USA, 2012. ACM.

[52] Lihua Zhao and Ryutaro Ichise. Mid-ontology learning from linked data. In *Proceedings of the 2011 Joint International Conference on The Semantic Web*, JIST'11, pages 112–127, Berlin, Heidelberg, 2012. Springer-Verlag.

[53] Jiwei Zhong, Haiping Zhu, Jianming Li, and Yong Yu. Conceptual graph matching for semantic search. In *Proceedings of the 10th International Conference on Conceptual Structures: Integration and Interfaces*, ICCS '02, pages 92–196, London, UK, UK, 2002. Springer-Verlag.

Appendix

A. Precision and recall obtained with different kc functions

Figure 14 complements Figure 13a by comparing the balance between precision and recall for different variants of EQUATER using the dev and $test$ sets. We use different shapes to represent different kc functions and different colours (black, white and grey) to represent different parameter settings for each kc function. It is clear that without kc functions, the similarity measure of EQUATER tends to favour high-recall but perhaps lose too much precision. All kc functions have the effect to balance towards precision due to the constraints on the number of examples required to compute similarity confidently. Among the three, the exp model generally produces the highest recall with trade-off of precision. To certain extent, this confirms our belief that the knowledge confidence score under the exponential model may converge too fast: it may be over-confident in small set of examples, causing EQUATER to over-predict equivalence. On the other hand, the threshold based model trades off recall for precision. The variants with the lgt model generally find the best balance - in fact, under unsupervised settings, achieve best or close-to-best precision.

The lgt model also warrants more stability since changing parameters caused little performance variation (note that the different coloured squares are generally cluttered, while the different coloured triangles and diamonds are far away). Although occasionally variants with the exp model may outperform those based on lgt (e.g., when $thd = km$ in the clustering experiment on dev), the difference is small and their performance is more dependent on the setting of the parameter in these cases and can sometimes underperform baselines. Based on these observations, we argue that the lgt model of knowledge confidence is better than exp, $-n$, or ~~ke~~.

B. Exploration during the development of EQUATER's similarity measure

In this section we present some earlier analysis that helped us during the development of EQUATER. These analysis helped us to identify useful features for evaluating relation equivalence, as well as unsuccessful features which we abandoned in EQUATER. We analyzed EQUATER's components ta and sa from a different perspective to understand if they could be useful indicators of equivalence B.1. We also explored another dimension - the ranges of relations B.2. The intuition is that ranges provide additional information about relations. Unfortunately our analysis showed that ranges derived for relations from data are highly inconsistent and therefore, they are not discriminative features for this task. As a result they were not used by EQUATER. We carried out all analysis using the dev dataset only.

B.1. ta and sa

We applied ta and sa separately to each relation pair in the dev dataset, then studied the distribution of ta and sa scores for true positives and true negatives. Figure 15 shows that both ta and sa create different distributional patterns of scores for positive and negative examples in the data. Specifically, the majority of true positives receive a ta score of 0.2 or higher and an sa score of 0.1 or higher, the majority of true negatives receive a $ta < 0.15$ and $sa < 0.1$. Based on such distinctive patterns we concluded that ta and sa could be useful indicators in discovering equivalent relations.

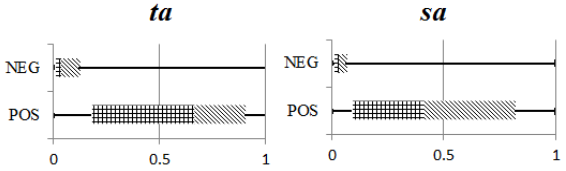

Fig. 15. Distribution of ta and sa scores for true postive and true negative examples in dev.

B.2. Ranges of relations

We also explored several ways of deriving ranges of a relation to be considered in measuring similarity. One simplistic method is to use ontological definitions. For example, the range of *dbo:birthPlace* of the concept *dbo:Actor* is defined as *dbo:Place* according to the DBpedia ontology. However, this does not work for relations that are not defined formally in ontologies, such as any predicates with the *dbpp* namespaces, which are very common in the datasets.

Instead, we chose to define ranges of a relation based on its objects $r(y)$ in data. One approach is to extract classes of their objects $y : r(y)$ and expect a dominant class for all objects of this relation. Thus we started by querying the DBpedia SPARQL endpoint with the following queries:

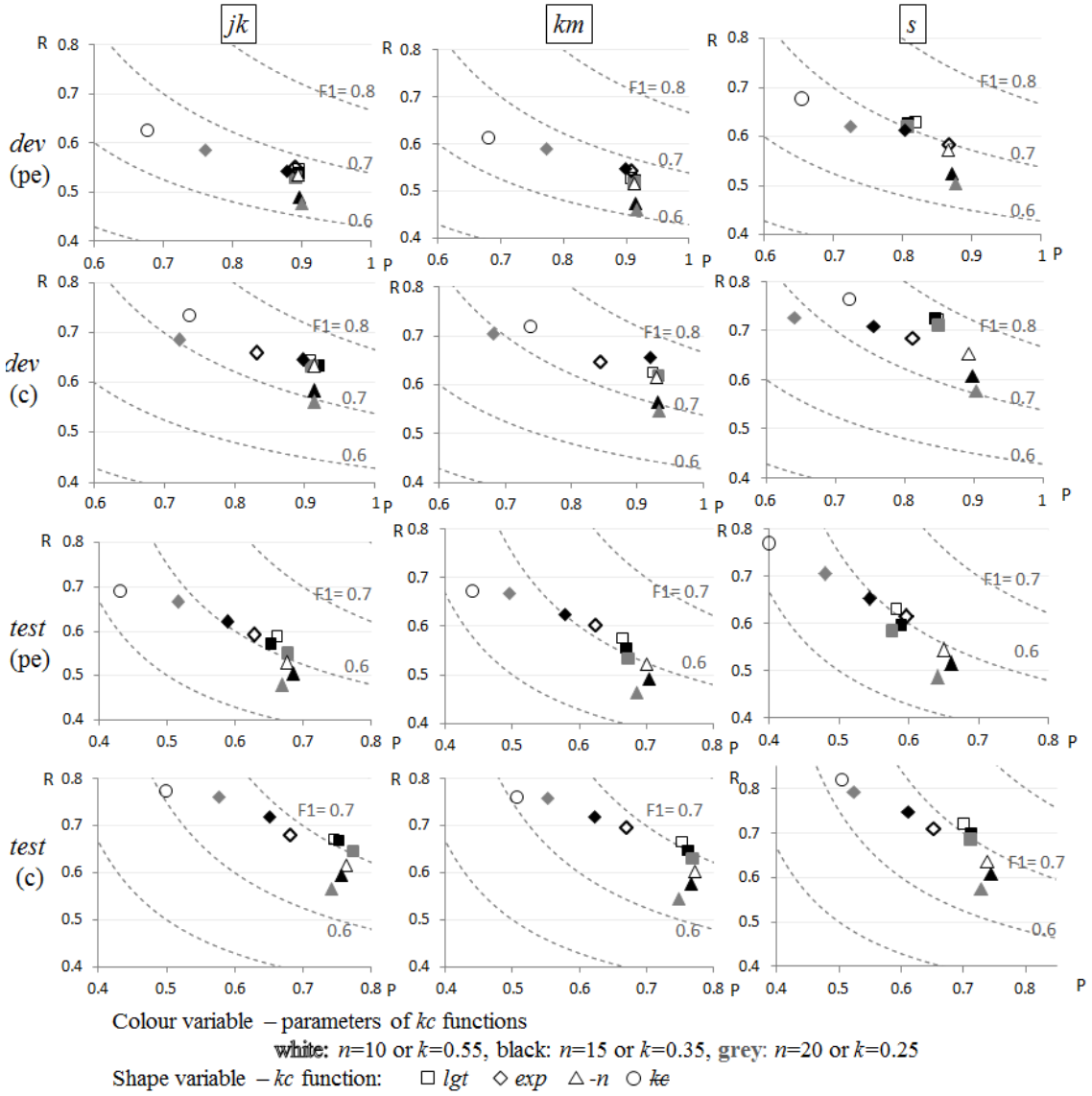

Fig. 14. Balance between precision and recall for EQUATER and its variant forms. pe - pair equivalence, c - clustering. The dotted lines are F1 references.

```
SELECT ?o ?range WHERE {
?s [RDF Predicate URI] ?o .
?s a [Concept URI] .
OPTIONAL {?o a ?range .}
}
```

Next, we counted the frequency of each distinct value for the variable *?range* and calculated its fraction with respect to all values. We found three issues that make this approach unreliable. First, if a subject *s* had an *rdfs:type* triple defining its type *c*, (e.g., *s rdfs:type c*), it appears that DBpedia creates additional *rdfs:type* triples for the subject with every superclass of *c*. For example, there are 20 *rdfs:type* triples for *dbr:Los_Angeles_County,_California* and the objects of these triples include *owl:Thing*, *yago:Object10000 2684* and *gml:_Feature* (gml: Geography Markup Language). These triples will significantly skew the data statistics, while incorporating ontology-specific knowledge to resolve the hierarchies can be an expensive process due to the unknown number of ontologies involved in the data. Second, even if we are able to choose always the most specific class according to each involved ontology for each subject,

we notice a high degree of inconsistency across different subjects in the data. For example, this gives us 13 most specific classes as candidate ranges for *dbo:birthPlace* of *dbo:Actor*, and the dominant class is *dbo:Country* representing just 49% of triples containing the relation. Other ranges include *scg:Place, dbo:City, yago:Location* (scg: schema.org) etc. The third problem is that for values of *?o* that are literals, no ranges will be extracted in this way (e.g., values of *?range* extracted using the above SPARQL template for relation *dbpp:othername* are empty when *?o* values are literals).

For these reasons, we abandoned the two methods but proposed to use several simple heuristics to classify the objects of triples into several categories based on their datatype and use them as ranges. Thus given the set of argument pairs $r(x, y)$ of a relation, we classified each object value into one of the six categories: *URI*, *number*, *boolean*, *date or time*, *descriptive texts* containing over ten tokens, and *short string* for everything else. A similar scheme is used in [51]. Although these range categories are very high-level, they should cover all data and may provide limited but potentially useful information for comparing relations.

We developed a measure called *maximum range agreement*, to examine the degree to which both relations use the same range in their data. Let RG_{r_1,r_2} denote the set of shared ranges discovered for the relation r_1 and r_2 following the above method, and $frac(rg^i_{r_1})$ denote the fraction of triples containing the relation r_1 whose range is the ith element in RG_{r_1,r_2}, we defined maximum range agreement (mra) of a pair of relations as:

$$mra(r_1, r_2) = \begin{cases} 0, & \text{if } RG_{r_1,r_2} = \emptyset \\ max\{frac(rg^i_{r_1}) + frac(rg^i_{r_2})\}, & \text{otherwise} \end{cases} \quad (16)$$

The intuition is that if two relations are equivalent, each of them should have a dominant range as seen in their triple data (thus a high value of $frac(rg^i_r)$ for both r_1 and r_2) and their dominant ranges should be consistent. Unfortunately, as Figure 16 shows, mra has little discriminating power in separating true positives from true negatives. As a result, we did not use it in EQUATER. In the error analysis, the errors due to incompatible datatypes may potentially benefit from range information of relations. However, the proposed six categories of ranges may have been too general to be useful.

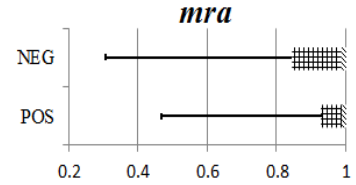

Fig. 16. Distribution of mra scores for true postive and true negative examples in dev.