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

Editor(s): Michelle Cheatham, Wright State University, USA; Isabel F. Cruz, University of Illinois at Chicago, USA; Jérôme Euzenat, INRIA & University of Grenoble, France; Catia Pesquita, University of Lisbon, Portugal
Solicited review(s): Andriy Nikolov, fluid Operations AG, Germany; Andreas Thalhammer, Karlsruhe Institute of Technology; Germany, Matteo Palmonari, University of Milan, Italy
Open review(s): Jérôme, Euzenat, INRIA & University of Grenoble, France

Ziqi Zhang [a,*], Anna Lisa Gentile [a], 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. This article addresses a number of limitations of state-of-the-art methods of Ontology Alignment: 1) they primarily address concepts and entities while relations are less well-studied; 2) many build on the assumption of the 'well-formedness' of ontologies which is unnecessarily true in the domain of Linked Open Data; 3) few have looked at schema heterogeneity from a single source, which is also a common issue particularly in very large Linked Dataset created automatically from heterogeneous resources, or integrated from multiple datasets. We propose a domain- and language-independent and completely unsupervised method to align equivalent relations across schemata based on their shared instances. We introduce a novel similarity measure able to cope with unbalanced population of schema elements, an unsupervised technique to automatically decide similarity threshold to assert equivalence for a pair of relations, and an unsupervised clustering process to discover groups of equivalent relations across different schemata. Although the method is designed for aligning relations within a single dataset, it can also be adapted for cross-dataset alignment where *sameAs* links between datasets have been established. Using three gold standards created based on DBpedia, we obtain encouraging results from a thorough evaluation involving four baseline similarity measures and over 15 comparative models based on variants of the proposed method. The proposed method makes significant improvement over baseline models in terms of F1 measure (mostly between 7% and 40%), and it always scores the highest precision and is also among the top performers in terms of recall. We also make public the datasets used in this work, which we believe make the largest collection of gold standards for evaluating relation alignment in the LOD context.

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 valuable knowledge source useful to many applications [30,15]. Following the rapid growth of the Web of Data is the increasingly pressing issue of heterogene-

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

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

ity, the phenomenon that multiple vocabularies exist to describe overlapping 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 [38,24,14]. This may hamper the usability of data over large scale and reduces 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 [12, 39] has contributed to a plethora of methods towards solving heterogeneity on the Semantic Web. A lot of these [34,27,35,21,25,29,41,9,7,36] are archived under the Ontology Alignment Evaluation Initiative (OAEI) [16]. However, we identify several limitations of the existing work. **First**, it has been criticized that most methods are tailored to cope with nicely structured and well defined ontologies [17], which are different from LOD ontologies characterized by noise and incompleteness [37,38,46,14,17,51]. Many features used by such methods may not be present in LOD ontologies.

Second, 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 [18,14,6]. Relation names are more diverse than concept names [6], and the synonymy and polysemy problems are also more typical [14,6]. This makes aligning relations in the LOD domain more challenging. Structural information of relations is particularly lacking [14,51], and the inconsistency between the intended meaning of schemata and their usage in data is more wide-spread [18,14,17]. Further, some emerging data linking problems such as linkkey discovery [2,45] can also benefit from the solutions of this problem since some methods depend on sets of mapped relations from different schemata.

Third, while it makes a lot of sense to study cross-dataset heterogeneity, solving heterogeneity from within a single dataset is also becoming increasingly important. Recent years have seen a rising trend of using (semi-)automatic Information Extraction techniques to create very large knowledge bases from semi- or unstructured text input [5,13,28] as they significantly reduces the tremendous human cost involved in traditional ontology engineering process. Due to polysemy in natural language, the extracted schemata are often heterogeneous. For example, in DBpedia[2], more than five relations are used to describe the name of a University, such as *dbpp*[3]*:uname*, *dbpp:name* and *foaf*[4]*:name*. In the ReVerb [13] database containing millions of facts extracted from natural language documents from the Web, the relation 'contain vitamin' has more than five expressions. The problem worsens when such datasets are exposed on the LOD cloud, as data publishes attempting to link to such datasets may struggle to conform to a universal schema. As a realistic scenario, the DBpedia mappings portal[5] is a community effort dedicated to solving heterogeneity within the DBpedia dataset itself.

Last, 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 [35, 29,44,19,6], while finding optimal thresholds requires tuning on expensive training data; unfortunately, the thresholds are often context-dependent and requires re-tuning for different tasks [22,40].

To address these issues, we introduce a completely unsupervised method for discovering equivalent relations for specific concepts, using only data-level evidence without any schema-level information. The method 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 cutoff thresholds based on patterns discovered in the data; (3) and an unsupervised clustering process that groups mutually equivalent relations, potentially discovering relation alignments among multiple schemata. The principle of the method is studying the shared instances between two relations. This makes it particularly suitable for matching relations across multiple ontologies annotating the same dataset, or for contributing matching ontologies when *sameAs* links between different datasets have been established.

[2] http://dbpedia.org/. All examples and data analysis based on DBpedia in this work uses its dataset in September 2013.

[3] dbpp:http://dbpedia.org/property/

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

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

For a thorough evaluation, 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 the proposed method 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. The proposed method successfully discovers equivalent relations across multiple schemata, and the similarity measure 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 competitive - it even outperforms the supervised models on one dataset in terms of F1.

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. Terminology and scope

An alignment between a pair of ontologies is a set of correspondences between entities across the ontologies [12,39]. 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*[6]. A correspondence asserts that 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 domain, data are not necessarily described by formal ontologies, but sometimes vocabularies that are simple renderings of relational databases [38]. 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 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 [12,39]. *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 measures has been widely employed for this purpose [6]. *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 [49]. *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.

In the following, we begin by a brief overview of the literature on the general topic of ontology alignment, then focus our discussion of related work on two dimensions that characterize this work: in the context of LOD, and aligning relations.

2.2. Ontology alignment in general

A large number of ontology alignment methods have been archived under the OAEI, which maintains a number of well-known public datasets and hosts annual evaluations. Work under this paradigm has been well-summarized in [12,39]. A predominant pattern shared by these work [34,27,35,21,25,29,41,9,7,36,19] is the strong preference of terminological and structural matchers to extensional methods [39]. Many of these show that ontology alignment can ben-

[6] 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.

efit from the use of background knowledge sources such as WordNet and Wikipedia [34,27,35,25,9,19]. There is also a trend of using a combination of different matchers, since it is argued that the suitability of a matcher is dependent on different scenarios and therefore combining several matchers could improve alignment quality [36].

These methods are not suitable for our task for a number of reasons. **First**, most methods are designed to align concepts and individuals, while adaptation to relations is not straightforward. **Second**, terminological and structural matchers require well-formed ontologies, which are not necessarily available in the context of LOD. As we shall discuss in the next sections, tougher challenges arise in the LOD domain and in the problem of aligning heterogeneous relations.

2.3. Ontology alignment in the LOD domain

Among the three categories of matchers, extensional matchers are particularly favoured due to certain characteristics of Linked Data.

First and foremost, vocabulary definitions are often highly inconsistent and incomplete [17]. Textual features such as labels and comments for concepts and relations that are used by terminological matchers 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.

Even when such information is available, the meaning of schemata may be dependent on their actual usage pattern in the data [37,17,51] (e.g., *foaf:Person* may represent researchers in a scientific publication dataset, but artists in a music dataset). This means that strong similarity measured at terminological or structural level does not always imply strict equivalence. This problem is found to be particularly prominent in the LOD domain [37,46,17]. Empirically, Jain et al. [23] and Cruz et al. [7] 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.

Second, the Linked Data environment is characterized by large volumes of data and the availability of many interconnected information sources [37,38,44, 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 [37,46].

The majority of state-of-the-art in the LOD domain employed extensional matchers. Nikolov et al. [37] proposed to recursively compute concept mappings and entity mappings based on each other. Concept mappings are firstly created based on the overlap of their individuals; such mappings are later used to support mapping individuals in LOD ontologies. The intuition is that mappings between entities at both levels influence each other. Similar idea was explored by Suchanek et al. [44], who 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 is not addressed.

Parundekar et al. [38] 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, such as the DBpedia ontology, by using the notion of 'restriction class'. A restriction class is defined by combining value-restricted-properties, such as *(rdfs[8]:type=Feature, featureCode=gn[9]:A.PCLI)*. This effectively defines more fine-grained concepts that can then be aligned with other ontologies. A similar matcher as Nikolov et al. [37] is used. Slabbekoorn et al. [43] explored a similar problem: matching a domain-specific ontology to a general purpose ontology. It is unclear if these approaches can be applied to relation matching or not.

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. [8] created a customization of the AgreementMaker system [9] 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. [11] also used extensional matchers in the LOD context but focusing on improving computation efficiency of the algorithms.

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

[8] rdfs=http://www.w3.org/2000/01/rdf-schema#

[9] gn=http://www.geonames.org/ontology#

2.4. Matching relations

Compared to concepts, aligning relations is generally considered to be harder [18,14,6]. 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 [6]. 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 [14,6].

Same relation names are found to bear different meanings in different contexts [18]. For example, in the DBpedia dataset 'before' is used to describe relationship between consecutive space missions, or Roman emperors such as Nero and Claudius. If a user creates a query using such relations without enough additional constraints, the result may contain irrelevant records since the user might be only interested in particular aspects of these polysemous relations [14]. 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[10] 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 can be seriously hampered if applied to matching relations, particularly in the LOD domain. Indeed, Cheatham et al. [6] compared a wide selection of string similarity measures in several tasks and showed their performance on matching relations to be inferior to matching concepts. Thus in line with Nikolov et al. [37] and Duan et al. [11], we argue in favour of extensional matchers.

We notice that 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. An extensional matcher similar to the Jaccard function compares the overlap in the subject sets of two relations, balanced by the overlap in their object sets. Zhao et al. [50,51] first created triple sets each corresponding to a specific subject that is an individual, such as *dbr*[11]*: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. Zhao et al. [52] used nine WordNet-based similarity measures to align properties from different ontologies. They firstly use NLP tools to extract terms from properties, then compute similarity between the groups of terms from the pair of properties. These similarity measures make use of the terminological and structural features of the WordNet lexical graph.

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 measure and Dice coefficient [22,11,14,17]. However, 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 [46]. 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 [37].

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

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

[11] dbr:http://dbpedia.org/resource/

ing relations in the LOD domain, and current methods [14,50,51] have been evaluated on smaller datasets than those used in this study.

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 [35,29,44,19,6]. Such thresholds are typically decided based on some annotated data (e.g., [29,41,18]), 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 [22,41,40]. For example, Seddiqui et al. [41] showed that for the same matcher, previously reported thresholds in related work may not perform satisfactorily on new tasks. For extensional matchers, finding best thresholds can be difficult since they too strongly depend on the collection of data [22,11].

One study that reduces the need of supervision in learning thresholds is based on active learning by Shi et al. [42]. The system automatically learns similarity thresholds by repeatedly asking feedback from a user. However, this method still requires certain supervision. Another approach adopted in Duan et al. [11] and Fu et al. [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 Duan et al. this varied from 1 to 86 in the ground truth). To the best of our knowledge, our work is the first that automatically detects thresholds without using annotated training data.

2.6. Remark

To conclude, 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 of data instances, advocate for a renewed inspection of existing ontology alignment methods. We believe that the solution rests on 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 [47] 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 [3,48]. The work described in this article is built on our prototype but extends it in several dimensions: (1) a revised and extended extensional matcher; (2) a method of automatic threshold detection without need of training 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 and studied by the majority of related work; secondly, hierarchical structures for relations are very rare, especially in the LOD domain.

3. Methodology

3.1. Task formalization

Our domain- and language-independent method for aligning heterogeneous relations belongs to the category of extensional matchers, 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.

[12] dbo:http://dbpedia.org/ontolgy/

<x, r, y>	*<dbr:Sydney_Opera_House, dbo[12]:openningDate, '1973'>, <dbr:Royal_Opera_House, dbo:openningDate, '1732'>, <dbr:Sydney_Opera_House, dbpp: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

We start by taking 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 for the concept. To approach the first goal, we firstly introduce a data-driven similarity measure (Section 3.2). For a specific concept we hypothesize that 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 chances. Therefore, we propose to automatically split the relation pairs into two groups based on patterns in their similarity scores and assert equivalence for pairs from the smaller group with higher similarity (Section 3.3). This also allows us to detect concept-specific thresholds. For the second goal, we apply unsupervised clustering to the set of equivalent pairs and create clusters of mutually equivalent relations (Section 3.4) for a concept. The clustering process discovers equivalence transitivity or invalidates pair-wise equivalence. This may also discover alignments among multiple schemata at the same time.

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 [47] [13].

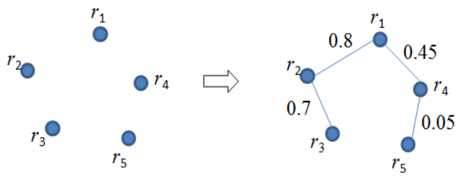

Fig. 1. The similarity measure computes a numerical score for pairs of relations. r_3 and r_5 has a score of 0.

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.

$$arg_{\cap}(r_1, r_2) = arg(r_1) \cap arg(r_2) \tag{1}$$

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

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

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 for discovering similar concepts [37] and studying subsumption relations between concepts [38,44].

[13] Notations have been changed.

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.

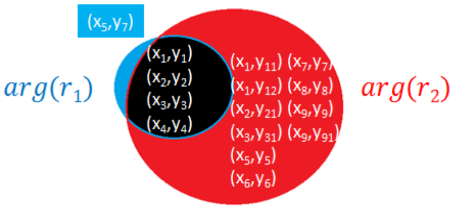

Fig. 2. Illustration of triple agreement.

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.

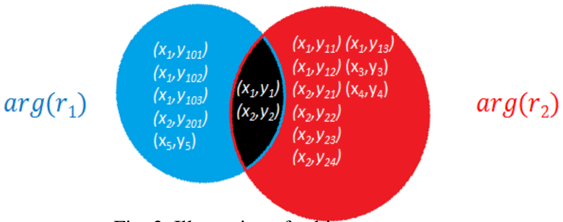

Fig. 3. Illustration of subject agreement.

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.

$$sub_{\cap}(r_1, r_2) = args_s(r_1) \cap args_s(r_2) \tag{3}$$

$$sub_{\cup}(r_1, r_2) = args_s(r_1) \cup args_s(r_2) \tag{4}$$

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

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

, 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

$$sa(r_1, r_1) = \alpha(r_1, r_2) \cdot \beta(r_1, r_2) \tag{7}$$

, and effectively:

$$sa(r_1, r_1) = \frac{\sum_{x \in sub_{\cap}(r_1, r_2)} \begin{cases} 1 & \text{if } \exists y : (x, y) \in arg_{\cap}(r_1, r_2) \\ 0 & \text{otherwise} \end{cases}}{|sub_{\cup}(r_1, r_2)|} \tag{8}$$

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 (i.e., $x \in sub_{\cap}(r_1, r_2)$ under summation), a large proportion of which will also share at least one object (i.e., $\exists y : (x, y) \in arg_{\cap}(r_1, r_2)$). 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 compute scores of similarity from different dimensions, as argued by Isaac et al. [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. Intuitively, 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 sub-tasks: (1) quantifying knowledge and (2) defining 'adequacy'.

The *quantification of knowledge* can be built on the principle of human *inductive learning*. 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 this context, 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).

We propose to model this problem based on the theory by Dewey [10], who suggested 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.

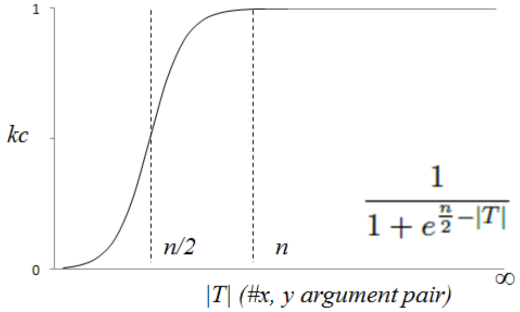

Fig. 4. The logistic function modelling knowledge confidence

Empirically, we model such a curve using a logistic function shown in Equation 9, where T denotes the set of argument pairs of a relation 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 (i.e., $|T|$), 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). The choice of n is to be discussed in Section 4.3.

$$kc(|T|) = lgt(|T|) = \frac{1}{1+e^{\frac{n}{2}-|T|}} \tag{9}$$

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 threshold 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:

$$ta^{kc}(r_1,r_2) = max\{\frac{|arg_{\cap}(r_1,r_2)|}{|arg(r_1)|} \cdot kc(|arg(r_1)|), \frac{|arg_{\cap}(r_1,r_2)|}{|arg(r_2)|} \cdot kc(|arg(r_2)|)\} \tag{10}$$

$$\begin{aligned} &sa^{kc}(r_1, r_1) = \\ &\alpha(r_1, r_2) \cdot \beta(r_1, r_2) \cdot kc(|arg_{\cap}(r_1, r_2)|) \end{aligned} \tag{11}$$

, where the choice of the kc model can be either lgt, or some alternative models to be detailed in Section 4. The choice of the kc model does not break the mathematical consistency of the formula. In Equation 10, 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. In Equation 11, our confidence about an sa score depends on the knowledge of the shared argument pairs between r_1 and r_2. The modification effect of kc is that with insufficient examples ($|T|$ the argument pairs of a relation), the triple agreement and subject agreement scores are penalized depending on the the size of $|T|$ and the formulation of kc. The penalty is reduced to neglectable when $|T|$ is greater than certain number.

Both Equations return a value between 0 and 1.0. Finally, the similarity between r_1 and r_2 is:

$$e(r_1, r_2) = \frac{ta^{kc}(r_1, r_2) + sa^{kc}(r_1, r_2)}{2} \tag{12}$$

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 a part of 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.

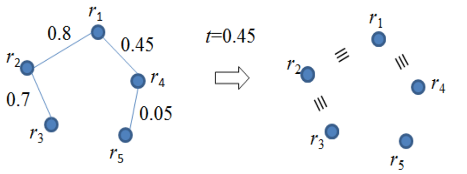

Fig. 5. Deciding a threshold beyond which pairs of relations are considered to be truely equivalent.

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 ones that also have non-zero similarity. The latter can happen due to imperfections in data such as schema misuse or merely by chance. 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[14]. Hence our goal is to split the pairs of relations into two groups based on their similarity scores, where the scores in the smaller group should be significantly higher than those in the other larger group. Then we assume that the pairs contained by the smaller group are truly equivalent pairs and the larger group contains noise and is discarded.

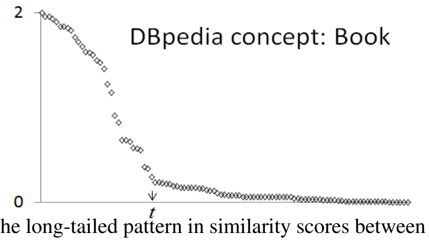

Fig. 6. The long-tailed pattern in similarity scores between relations computed using e. t could be the boundary threshold.

We do this 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 determines the best arrangement of data values into different classes. It seeks to reduce within-class variance while maximizing between-class variance. Given a set of data points and i the expected number of groups in the data, the algorithm starts by splitting 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. At the end of each iteration, a new splitting is created based on the two figures and the process is repeated until the sum of the within class deviations reaches a minimal value. The resulting optimal classification is called Jenks natural breaks.

[14] We manually checked a random set of 40 concepts (approximately 17% of all) in our collection of datasets and found 38 show a strong pattern like this while the other two seem to be borderline.

Essentially, Jenks natural breaks is k-means [32] applied to univariate data.

Empirically, given the set of relation pairs for a concept C we apply Jenks natural breaks to the similarity scores with $i = 2$ to break them into two groups. We expect to obtain a smaller group of pairs with significantly higher similarity scores than another larger group of pairs, and we consider those pairs from the smaller group as truly equivalent while those from the larger group as non-equivalent.

Although it is not necessary to compute a threshold of similarity (denoted as t) under this method, this can be done by simply selecting the maximum similarity score in the larger group[15] as similarity scores of all equivalent pairs in the smaller group should be greater than this value. We will use this method to derive our threshold later in experiments when compared against other methods.

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 there are no shared instances between the two relations to directly support a non-zero similarity 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).

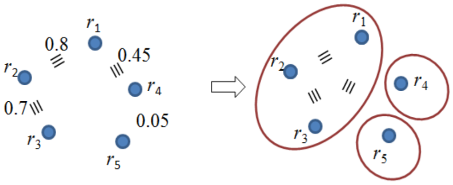

Fig. 7. Clustering discovers transitive equivalence and invalidates weak links.

To address such issues we cluster mutually equivalent relations for a concept. Essentially clustering brings in additional context to decide pair-wise equivalence, which allows us to 1) discover equivalence that may be missed by the proposed similarity measure and the threshold detection method, and 2) discard equivalence assertion the similarity of which appears relatively weak to join a cluster. Potentially, this also allows creating alignments between multiple schemata at the same time. Given the set of equivalent relation pairs discovered before, $\{r_i, r_j : e(r_i, r_j) > t\}$, 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:

$$m_{i,j} = 1.0 - e(r_i, r_j) \tag{13}$$

where 1.0 is the maximum possible similarity given by Equation 12. Then we use the group-average agglomerative clustering algorithm [33] that takes M as input and creates clusters of equivalent relations. This is a state-of-the-art 'bottom-up' clustering algorithm that follows an iterative approach beginning with each data point (i.e., a relation) as a separate cluster. In each iteration, clusters can be merged based on their distance (using M). This repeats for several iterations until all data points are grouped into a single cluster. This process creates a hierarchical arrangement of clusters, called a dendrogram. Next, deciding the optimal clusters for the data involves cutting the dendrogram at an appropriate level, which can be data-dependent. We compute this using the well-known Calinski and Harabasz [4] stopping rule. In general, the idea is to find the level at which the variance ratio criterion - based on the between-cluster variance and the within-cluster variance - is maximized.

At the end of this process, we obtain groups of relations that are mutually equivalent in the context of a specific concept C, such as that shown in the right part of Figure 7.

4. Experiment Settings

We design a series of experiments to thoroughly evaluate our method in terms of the two goals described at the beginning of Section 3, i.e., its capability of predicting equivalence of two relations of a concept (*pair equivalence*) and clustering mutually equivalent relations (*clustering*) for a concept. Different settings

[15] Or the minimum in the smaller group. Practically there is no difference as the purpose of a threshold is to separate equivalent pairs from non-equivalent ones.

are created along three dimensions by selecting from several choices of 1) similarity measures, 2) threshold detection methods and 3) different knowledge confidence models 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 (***lev***) that proves to be one of the best performing terminological matcher for aligning both relations and classes [6]. Specifically, we measure the string similarity (or distance) between the URIs of two relations, but we remove namespaces from relation URIs before applying the measure. 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'). Note that this method is also used by Zhao et al. [52] for aligning relations.

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

$$fu(r_1, r_2) = \frac{|arg_s(r_1) \cap arg_s(r_2)|}{|arg_s(r_1) \cup arg_s(r_2)|} \cdot \frac{|arg_o(r_1) \cap arg_o(r_2)|}{|arg_o(r_1) \cup arg_o(r_2)|} \quad (14)$$

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 [22,11,40]. 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:

$$jc(r_1, r_2) = \frac{\sqrt{|arg(r_1) \cap arg(r_2)| \cdot (|arg(r_1) \cap arg(r_2)| - 0.8)}}{|arg(r_1) \cup arg(r_2)|} \quad (15)$$

4.2. Methods of detecting thresholds

We compare three different methods of threshold detection. The first is Jenks Natural Breaks (***jk***) that is used in the proposed method, discussed in Section 3.3. For the second method we use the k-means clustering algorithm (***km***) for unsupervised threshold detection. As discussed before, the two methods are very similar, with the main distinction being that jk is particularly suitable for univariate data. Hence we derive the threshold in the same way by choosing the boundary value that separates the two clusters. 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) as equivalent or not. Then given a similarity measure, we use it to score each pair and rank pairs by scores. A good measure is expected to rank equivalent pairs higher than inequivalent ones. Next, we calculate accuracy at each rank taking into account the number of equivalent v.s. inequivalent pairs by that rank, and it is expected that maximum accuracy can be reached at one particular rank. 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. Knowledge confidence models kc

We compare the proposed 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 16, where k is a scalar that controls the speed of convergence and $|T|$ returns the number of observed examples in terms of argument pairs.

$$kc(|T|) = exp(|T|) = 1 - e^{-|T| \cdot k} \tag{16}$$

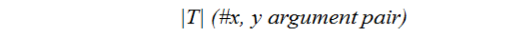

Fig. 8. The exponential function modelling knowledge confidence

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 close to 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. [5] 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 model 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 variant of the proposed similarity measure without kc, denoted by $\boldsymbol{e}^{\cancel{kc}}$, which simply combines ta and sa in their original forms. Note that this can be considered as the prototype similarity measure[16] we developed earlier [47].

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, knowledge confidence model, and threshold detection respectively). Note that the variable kc only applies to the proposed similarity measure, not baseline measures. Thus jc_s means scoring relation pairs using the corrected Jaccard function (jc), then find threshold based on training data (supervised, s); while e^{lgt}_{jk} is the proposed method in its original form, i.e., using the proposed 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.

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

4.5.1. DBpedia

Although a number of benchmarking datasets are published under the OAEI, as discussed before, they are not suitable for our task since they do not repre-

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

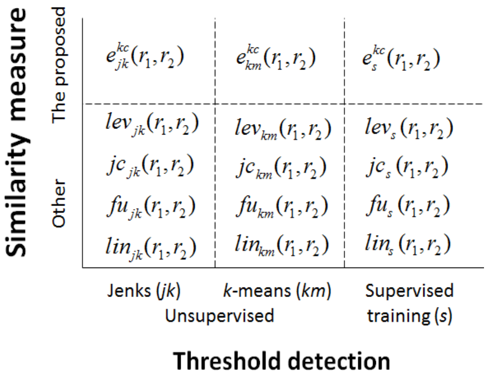

Fig. 9. Different settings based on the choices of three dimensions. *kc* is a variable whose value could be *lgt* (Equation 9), *exp* (Equation 16), or $-n$.

sent the particular characteristics in the LOD domain and the number of aligned relations is also very small - less than 2‰(56) of mappings found in their gold standard datasets are equivalent relations[17]. Therefore, we study the problem of heterogeneous relations on DBpedia, the largest LOD dataset serving as a hub for connecting multiple sources in the LOD domain. DBpedia is also a representative example of relation heterogeneity [18,14]. Multiple vocabularies are used in the dataset, including RDFS, Dublin Core[18], WGS84 Geo[19], FOAF, SKOS[20], the DBpedia ontology, original Wikipedia templates and so on. The DBpedia ontology version 3.9 covers 529 concepts and 2,333 different relations[21]. Heterogeneity is mostly 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 template usually defines a concept and its properties[22]. 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 a pressing issue that has attracted particular effort, which is known as the DBpedia mappings portal. The goal of the portal is to invite 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. It is known that manually creating such mappings requires significant work, and as a result, as by September 2015, less than 65% of mappings between Wikipedia template properties and relations in the DBpedia ontology are complete[23]. Hence the community can significantly benefit from an automatic mapping system.

4.5.2. Datasets

We collected **three** datasets for experiments. The **first dataset (*dbpm*)** 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. 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, 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.

For these reasons, we manually created the **second** and the **third** datasets based on 40 DBpedia (DBpedia ontology version 3.8) and YAGO[24] concepts. The choices of such concepts are based on the QALD1 question answering dataset[25] 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[26].

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

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

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

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

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

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

[23] http://mappings.dbpedia.org/server/statistics/en/, visited on 15-09-2015

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

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

[26] Note that DBpedia by default returns a maximum of 50,000 triples per query. We did not incrementally build the exhaustive re-

```
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 describe 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 to include only 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 contain all true positives of the concept since there can be equivalent pairs of relations receiving a zero similarity score by all measures. However, this should be a reasonable approximation. Moreover it would be extremely expensive to annotate the set of all pairs completely.

The data is annotated by four computer scientists and then randomly split into a **development set (*dev*)** containing 10 concepts and a **test set (*test*)** containing 30 concepts. The *dev* set is used for analyzing the patterns in the data, developing components of the method, and learning cutoff thresholds by the supervised experimental settings; the *test* set is used for evaluation. All datasets and associated resources used in this study are publicly available[27]. 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.

4.5.3. *Difficulty of the task*

Annotating relation equivalence is a non-trivial task. The process took three weeks, where one week was spent on creating guidelines. Annotators queried DBpedia for triples containing the relation to assist their interpretation. However, a notable number of relations are still incomprehensible. As Table 2 shows, this accounts for about 8 to 14% of data. 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. 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 Hripcsak et al. [20], and the average IAA is 0.8 while the lowest bound is 0.68 and the highest is 0.87.

	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

Moreover, there is also a high degree of inconsistent usage of relations. A typical example is *dbo:railway Platforms* 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 Fu et al. [14].

Table 2 and Figure 10 both show that the dataset is overwhelmed by negative examples (i.e., true negatives). 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* are true positive). These findings suggest that finding equivalent relations on Linked Data is indeed a challenging task.

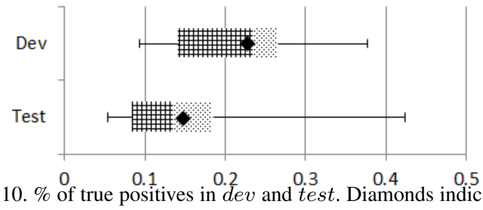

Fig. 10. % of true positives in dev and $test$. Diamonds indicate the mean.

4.6. *Running experiment*

Each setting (see Section 4.4) to be evaluated starts with one concept at a time to query the DBpedia

sult set for each concept since we believe the data size is sufficient for experiment purposes.

[27] http://staffwww.dcs.shef.ac.uk/people/Z.Zhang/resources/swj2015/data_release.zip. The cached DBpedia query results are also released.

	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 the dev set

SPARQL endpoint to obtain triples related to the concept (see the query template before). Querying DBpedia is the major bottleneck throughout the process and therefore, we cache query results and re-use them for different settings. Next, candidate relation pairs are generated from the triples and 1) their similarity is computed using the measure of each setting; then 2) relations are clustered based on pair-wise similarity, to generate clusters of mutually equivalent relations for the concept.

The output from (1) is then evaluated against the three gold standard datasets described above. Only true positive pairs are considered as the larger amount of true negatives may bias results. 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, and r_2 is equivalent to r_3 then we assume r_1 is also equivalent to r_3 and group the three relations in a single cluster. For the same reason, we consider only clusters of positive pairs.

We ran experiments on a multi-core laptop computer with an allocated 2GB of memory. However, the system is not programmed to utilize parallel computing as the actual computation (i.e., excluding querying DBpedia) is fast. When caching is used, on average it takes 40 seconds to process a concept, which has an average of 264 pairs of relations.

5. Results and discussion

We firstly show the learned thresholds based on the dev set for each measure. Table 3 shows the learned thresholds for the baseline similarity measures, and Table 4 shows the thresholds for different variants of the proposed method by replacing the knowledge confidence component kc. 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.

	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 proposed similarity measure derived from the dev set

5.1. Performance of the proposed method

In Table 5 we show the results of our proposed method on the three datasets with varying n in the lgt knowledge confidence model. 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 our method manages to discover alignment between multiple schemata used in DBpedia.

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 the proposed method on all datasets. R - Recall

On average, our method obtains 0.65$\sim$0.67 F1 in predicting pair equivalence on the dev set and 0.59$\sim$0.61 F1 on the $test$ set. These translate to 0.74 and 0.70 clustering accuracy on each dataset respectively. For the $dbpm$ set, 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 our method appears to be insensitive to the varying values of n. This stability is a desirable feature since it may be unnec-

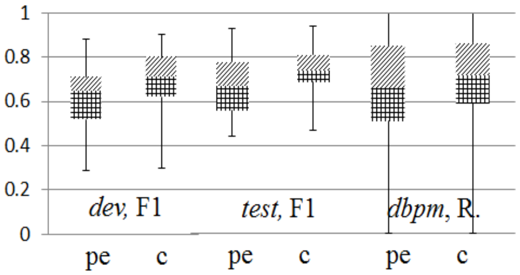

Fig. 11. Performance ranges on a per-concept basis for *dev*, *test* and *dbpm*. R - Recall, pe - pair equivalence, c - clustering

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.

essary to tune the model and therefore, the method is less prone to overfitting. This also confirms the hypothetical link between the amount of seed data needed for bootstrapping relation learning and the amount of examples needed to obtain maximum knowledge confidence in our method.

Figure 11 shows that the performance of our method can vary depending on specific concepts. To understand the errors, we randomly sampled 100 false positive 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[28]. The prevalence of each type is shown in Table 7.

5.1.1. 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).

5.1.2. False negatives

The first type of common errors is due to **representation of objects**. For instance, for *dbo:American FootballPlayer*, *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, our method scored 0 in recall. However, we believe that these gold standards are largely incomplete. For example, we consider most

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

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.

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* discovered by the proposed method are false positives according to the gold standard.

equivalent relation pairs proposed by our method as shown in Table 8 to be correct.

Some of the error types mentioned above could be rectified with by modifying the proposed method in certain ways. String similarity measures 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.

5.2. Performance against baselines

Next, Table 9 shows the improvement of the proposed method over different models that use a baseline similarity measure. Since the performance of the method 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 the proposed method significantly outperforms most baseline models, either supervised or unsupervised. Exceptions are noted against jc_s in the clustering task on the dev dataset, where the method achieves comparable results; and on the $dbpm$ dataset in both pair equivalence and clustering tasks, where it 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 that the method has more balanced performance. The relatively larger improvement over unsupervised baselines than over supervised baselines may suggest that the scores produced by the proposed similarity measure 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 the proposed method against baselines on the dev and $test$ datasets. We use three different shapes to represent variants with different n values in the knowledge confidence model lgt; for baseline models we use different shapes to represent different similarity measures and different colours (black, white and grey) to represent different threshold detection methods thd. It is clear that the proposed method 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 the dev and $test$ sets with arguably balanced precision and recall, in clustering experiments the figures 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 over 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

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 the proposed method over baselines. The highest improvements on each dataset are in **bold**. Negative changes are in *italic*. See Section 4.4 and Figure 9 for notations.

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’).

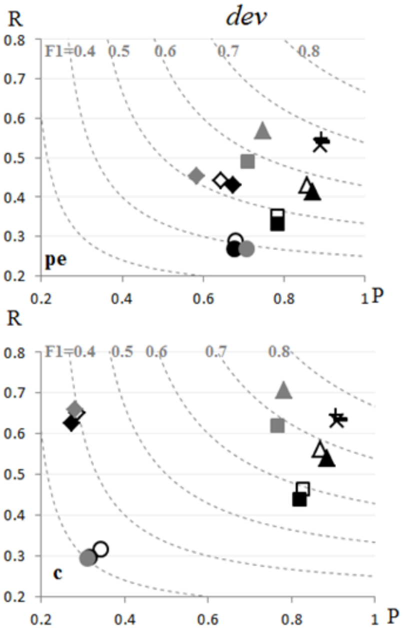

Fig. 12a. Balance between precision and recall for the proposed method and baselines on the *dev* set. pe - pair equivalence, c - clustering. The dotted lines are F1 references.

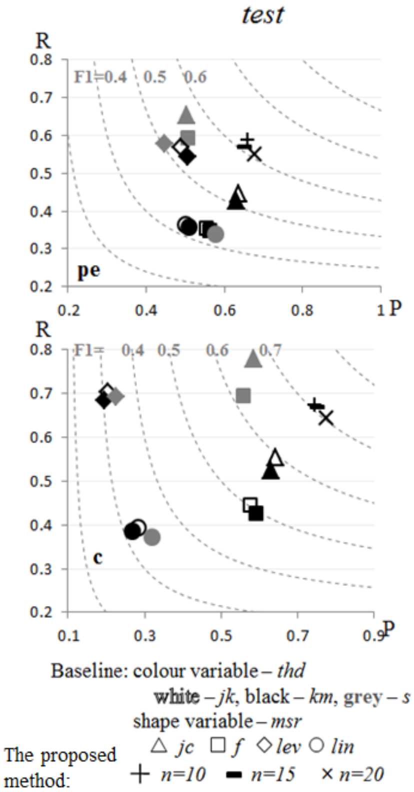

Fig. 12b. Balance between precision and recall for the proposed method and baselines on the *test* set.

5.3. Variants of the proposed method

In this section, we compare the proposed method against several alternative designs based on the alternative choices of knowledge confidence (kc) models 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 the method and compare them against the method in its original form. In addition, we also compared against the similarity measure $e^{\cancel{kc}}$, which only takes ta and sa without the knowledge confidence factor. 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.

5.3.1. Alternative knowledge confidence models kc

Figure 13a compares variants of the proposed method by alternating the kc model under each threshold detection method thd. The best performing baseline similarity measure is marked as a horizontal dotted line across each groups of bars. Since each of the kc models lgt, exp and $-n$ requires a parameter to be set, we show the ranges of performance gained under different parameter settings. 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 thd method (i.e., within the sections of jk (Jenks natural breaks), km (k-means), or s (supervised) in Figure 13a) , the variants without knowledge confidence models ($e^{\cancel{kc}}_{jk}$, $e^{\cancel{kc}}_{km}$ and $e^{\cancel{kc}}_{s}$) outperform the best baseline in most cases. This suggests that triple agreement ta and subject agreement sa are indeed more effective indicators of relation equivalence than other measures, and also suggests that the issue of unbalanced populations of schemata in the LOD domain is very common. Secondly, we can see that the F1 accuracy obtained on the dev and $test$ sets does benefit from the addition of the kc modifier (i.e., comparing e^{-n}_{thd}, e^{lgt}_{thd}, and e^{exp}_{thd} against $e^{\cancel{kc}}_{thd}$, within each groups of bars). The changes are also substantial on the $test$ set. However, the recall obtained on the $dbpm$ set seems to degrade slightly when the kc modifier is used. This seems to suggest that kc models may trade off recall for precision to achieve overall higher F1. Thirdly, in terms of the three kc models, the performance given by the exp and $-n$ models appears to be volatile since changing their parameters caused considerable variation of performance in most cases (note that the black caps on top of the bars corresponding to variants based on these two kc models are thicker than those corresponding to variants using lgt). When the supervised thresholds are used (s), in the clustering experiments on both the dev and $test$ sets, e^{-n}_{s} and e^{exp}_{s} even underperformed the best baseline in terms of F1.

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

5.3.2. Alternative threshold detection methods thd

Figure 13b is a re-arranged view of Figure 13a, in the way that it compares variants of the proposed method by alternating the thd method under each group of the same knowledge confidence model kc. This gives a better view for comparing different choices of thd. Generally it appears that regardless of the kc model, the Jenks natural breaks (jk) method has slight advantage over k-means (km), which is likely due to its particular fit with univariate data. In many cases, e^{lgt}_{jk}, e^{exp}_{jk} and e^{-n}_{jk} also obtain close and sometimes higher performance to their supervised counterparts e^{lgt}_{s}, e^{exp}_{s} and e^{-n}_{s} respectively. This suggests Jenks Natural Breaks an effective method for automatic threshold detection in the proposed method.

5.3.3. Limitations

The current version of the proposed method 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 our method. 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 pre-process that maps concepts and instances from different datasets using existing *sameAs* mappings, as done by Parundekar et al. [38] and Zhao et al. [50].

However, when incorrect *sameAs* mappings are present, they can impact on the accuracy of the pro-

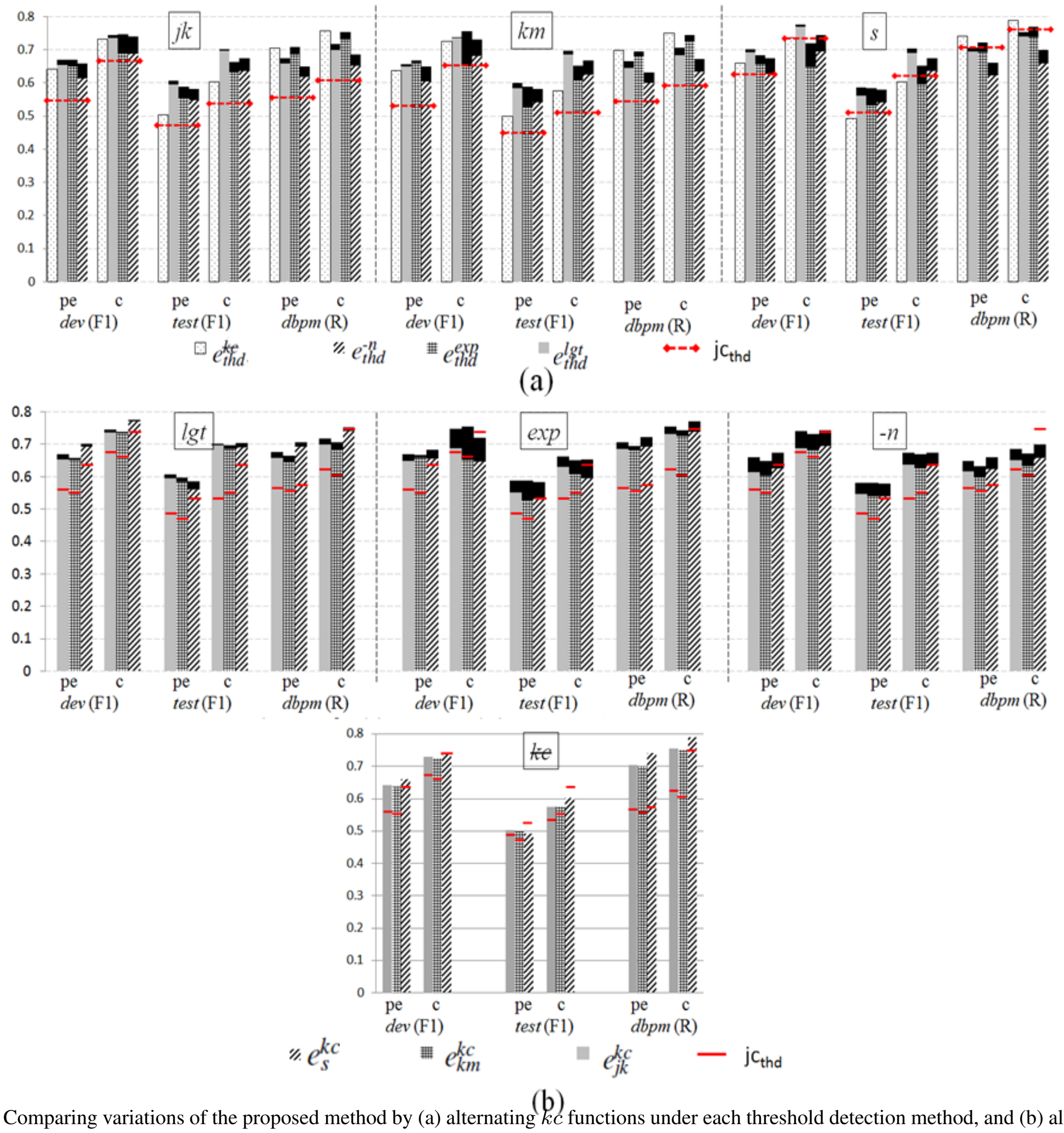

Fig. 13. Comparing variations of the proposed method by (a) alternating kc functions under each threshold detection method, and (b) alternating thd methods under each knowledge confidence function (incl. without kc).

posed method and hence this is our **second limitation**. Generally speaking, there are two cases of incorrect *sameAs* mappings, i.e., at *concept* level or *instance* level. Recall that the proposed similarity measure looks at relations of a specific concept and operates on a relation's argument pairs (see triple agreement, ta and subject agreemetn, sa), where the subjects are instances of the concept, thus it is more prone to errors in the second case.

In the first case, suppose we have an incorrect concept mapping C_a *sameAs* C_b. The implication is that when we query for triples containing instances of C_a at the beginning of the method, we also obtain irrelevant triples for C_b. Let r_a denote any relation for C_a, and r_b denote any relation for C_b, the proposed method may make mistake if it predicts $r_a \equiv r_b$. This is possible when r_a and r_b has overlapping argument pairs, which then requires an overlap between the instances of C_a and C_b. In other words, a reasonable number of *sameAs* links must have already been established between the instances of C_a and C_b (see discussion below). Otherwise, an incorrect mapping at concept level should not impact on the proposed method.

In the second case, incorrectly mapped instances could indeed influence the method under two conditions. First, both ta and sa depend on a relation's argument pairs, thus the incorrect instance alone (either as the subject or object of a triple) is insufficient to influence the method unless it contributes to form an argument pair which is identical for two relations (i.e., the object or subject that appears with the incorrect *sameAs* instance in the triple must be found for both relations) . Second, a reasonable number of incorrectly mapped instances must be present and must satisfy the above condition, as the knowledge confidence modifier will penalize the ta and sa scores without sufficient supporting evidence. Nevertheless, any extensional matchers that utilize $sameAs$ links across datasets are likely to suffer from incorrect mappings at instance level.

6. Conclusions

This article explored the problem of aligning heterogeneous relations in LOD datasets, particularly focusing on heterogeneity from within a single dataset. 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 in the context of cross-datasets, solving relation heterogeneity and in particular, in a single very large LOD dataset 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 with a domain- and language-independent and unsupervised method to align relations based on their shared instances in a dataset. In its current form, the method fits best with aligning relations from different schemata used in a single Linked Dataset, but 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 the method in two tasks: predicting relation pair equivalence and discovering clusters of equivalent relations. These experiments have confirmed the advantage of the method: 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 the proposed method, 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 a 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 the proposed method 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 the possibility of utilizing *sameAs* links between datasets to address cross-dataset relation alignment with focus on the previously discussed issues, and also aim to extend the method into to a full ontology alignment system.

Acknowledgement Part of this research has been sponsored by the EPSRC funded project LODIE: Linked Open Data for Information Extraction, EP/J019488/1. We thank reviewers for their invaluable comments that have helped us improve this article substantially.

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] Manuel Atencia, Jérôme David, and Jérôme Euzenat. Data interlinking through robust linkkey extraction. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *Proceedings of the 21st European conference on artificial intelligence (ECAI)*, pages 15–20, Amsterdam, NL, 2014. IOS press.

[3] Eva Blomqvist, Ziqi Zhang, Anna Lisa. Gentile, Isabelle Augenstein, and Fabio Ciravegna. Statistical knowledge patterns for characterizing linked data. In *Proceedings of the 4th workshop on Ontology and Semantic Web Patterns, at International Semantic Web Conference 2013*. CEUR-WS.org, 2013.

[4] Tadeusz Caliński and Jerzy Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974.

[5] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom M. Mitchell. Toward an architecture for never-ending language learning. In Maria Fox and David Poole, editors, *Proceedings of the 24th AAAI Conference*

on Artificial Intelligence, pages 1306–1313, California, USA, 2010. AAAI Press.

[6] Michelle Cheatham and Pascal Hitzler. String similarity metrics for ontology alignment. In Harith Alani, Lalana Kagal, Achille Fokoue, Paul Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha Noy, Chris Welty, and Krzysztof Janowicz, editors, *Proceedings of the 12th International Semantic Web Conference*, pages 294–309, Berlin, Heidelberg, 2013. Springer-Verlag.

[7] Isabel F. Cruz, Matteo Palmonari, Federico Caimi, and Cosmin Stroe. Towards 'on the go' matching of linked open data ontologies. In Pavel Shvaiko, JÃľrÃť me Euzenat, Fausto Giunchiglia, and Bin He, editors, *Proceedings of the workshop on Discovering Meaning On the Go in Large and Heterogeneous Data, at the 22nd International Joint Conference on Artificial Intelligence*, California, USA, July 2011. AAAI Press.

[8] 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.

[9] 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 the 5th International Workshop on Ontology Matching, at the 9th International Semantic Web Conference*. CEUR-WS.org, 2010.

[10] Russell Dewey. Chapter 7: Cognition. In *Psychology: An Introduction*. Psych Web, http://www.intropsych.com/, 2007.

[11] 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 Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist, editors, *Proceedings of the 11th international conference on The Semantic Web*, pages 49–64, Berlin, Heidelberg, 2012. Springer-Verlag.

[12] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer-Verlag, Heidelberg (DE), 2nd edition, 2013.

[13] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[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 7th International Conference on Knowledge Capture*, pages 41–48, New York, NY, USA, 2013. ACM.

[16] Bernardo Cuenca Grau, Zlatan Dragisic, Kai Eckert, Jeréome Euzenat, AlïňĄo Ferrara, Roger Granada, Valentina Ivanova, Ernesto Jimenez-Ruiz11, Andreas Oskar Kempf, Patrick Lambrix, Christian Meilicke, Andriy Nikolov, Heiko Paulheim, Dominique Ritze, Francǫis Scharffe, Pavel Shvaiko, Cássia Trojahn, and Ondřej Zamazal. Preliminary results of the ontology alignment evaluation initiative 2013. In *Proceedings of the 8th International Workshop on Ontology Matching, at the 12th International Semantic Web Conference*. CEUR-WS.org, 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 workshop on Linked Data on the Web, at the 21th International World Wide Web Conference*. CEUR-WS.org, 2012.

[18] Lushan Han, Tim Finin, and Anupam Joshi. GoRelations: an intuitive query system for dbpedia. In Jeff Pan, Huajun Chen, Hong-Gee Kim, Juanzi Li, Zhe Wu, Ian Horrocks, Riichiro Mizoguchi, and Zhaohui Wu, editors, *Proceedings of the 1st Joint International Conference on The Semantic Web*, pages 334–341, Berlin, Heidelberg, 2012. Springer-Verlag.

[19] Sven Hertlingand and Heiko Paulheim. WikiMatch - using Wikipedia for ontology matching. In *Proceedings of the 7th International Workshop on Ontology Matching, at the 11th International Semantic Web Conference*, pages 37–48. CEUR-WS.org, 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 and Knowledge 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 Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *Proceedings of the 6th International Semantic web and the 2nd Asian Semantic web conference*, 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 Peter Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Pan, Ian Horrocks, and Birte Glimm, editors, *Proceedings of the 9th International Semantic Web Conference*, 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 Grigoris Antoniou, Marko Grobelnik, Elena Simperl, Bijan Parsia, Dimitris Plexousakis, Pieter De Leenheer, and Jeff Pan, editors, *Proceedings of the 8th Extended Semantic Web Conference*, 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. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 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] Patrick Lambrix and He Tan. SAMBO-a system for aligning and merging biomedical ontologies. *Journal of Web Semantics*, 4(3):196–206, September 2006.

[28] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, SÃűren Auer, and Christian Bizer. DBpedia âĂŞ a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015.

[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, 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] Dekang Lin. An information-theoretic definition of similarity. In *Proceedings of the 5th International Conference on Machine Learning*, pages 296–304, San Francisco, CA, USA, 1998b. Morgan Kaufmann Publishers Inc.

[32] James B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

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

[34] Miklos Nagy, Maria Vargas-Vera, and Enrico Motta. Multi agent ontology mapping framework in the AQUA question answering system. In Alexander Gelbukh, Álvaro de Albornoz, and Hugo Terashima-Marín, editors, *Proceedings of the 4th Mexican International Conference on Advances in Artificial Intelligence*, pages 70–79, Berlin, Heidelberg, 2005. Springer-Verlag.

[35] Miklos Nagy, Maria Vargas-Vera, and Enrico Motta. DSSim - managing uncertainty on the semantic web. In *Proceedings of the 2nd International Workshop on Ontology Matching, at the 6th International Semantic Web Conference*, CEUR Workshop Proceedings. CEUR-WS.org, 2007.

[36] Duyhoa Ngo, Zohra Bellahsene, and Remi Coletta. A generic approach for combining linguistic and context profile metrics in ontology matching. In Robert Meersman, Tharam Dillon, Pilar Herrero, Akhil Kumar, Manfred Reichert, Li Qing, Beng-Chin Ooi, Ernesto Damiani, Douglas C. Schmidt, Jules White, Manfred Hauswirth, Pascal Hitzler, and Mukesh Mohania, editors, *Proceedings of the Confederated International Conference on On the Move to Meaningful Internet Systems*, pages 800–807, Berlin, Heidelberg, 2011. Springer-Verlag.

[37] Andriy Nikolov, Victoria Uren, Enrico Motta, and Anne Roeck. Overcoming schema heterogeneity between linked semantic repositories to improve coreference resolution. In Asunción Gómez-Pérez, Yong Yu, and Ying Ding, editors, *Proceedings of the 4th Asian Conference on The Semantic Web*, pages 332–346, Berlin, Heidelberg, 2009. Springer-Verlag.

[38] Rahul Parundekar, Craig A. Knoblock, and José Luis Ambite. Linking and building ontologies of linked data. In Peter Patel-Schneider, Yue Pan, Pascal Hitzler, Peter Mika, Lei Zhang, Jeff Pan, Ian Horrocks, and Birte Glimm, editors, *Proceedings of the 9th International Semantic Web Conference*, pages 598–614, Berlin, Heidelberg, 2010. Springer-Verlag.

[39] 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.

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

[41] Md. Hanif Seddiqui and Masaki Aono. An efficient and scalable algorithm for segmented alignment of ontologies of arbitrary size. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 7(4):344–356, 2009.

[42] Feng Shi, Juanzi Li, Jie Tang, Guotong Xie, and Hanyu Li. Actively learning ontology matching via user interaction. In Abraham Bernstein, David R. Karger, Tom Heath, Lee Feigenbaum, Diana Maynard, Enrico Motta, and Krishnaprasad Thirunarayan, editors, *Proceedings of the 8th International Semantic Web Conference*, pages 585–600, Berlin, Heidelberg, 2009. Springer-Verlag.

[43] Kristian Slabbekoorn, Laura Hollink, and Geert-Jan Houben. Domain-aware ontology matching. In Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist, editors, *Proceedings of the 11th International Conference on The Semantic Web*, pages 542–558, Berlin, Heidelberg, 2012. Springer-Verlag.

[44] Fabian M. Suchanek, Serge Abiteboul, and Pierre Senellart. PARIS: Probabilistic alignment of relations, instances, and schema. *Proceedings of the VLDB Endowment*, 5(3):157–168, November 2011.

[45] Danai Symeonidou, Vincent Armant, Nathalie Pernelle, and Fatiha Saïs. SAKey: Scalable almost key discovery in rdf data. In Peter Mika, Tania Tudorache, Abraham Bernstein, Chris Welty, Craig Knoblock, Denny Vrandečić, Paul Groth, Natasha Noy, Krzysztof Janowicz, and Carole Goble, editors, *Proceedings of the 13th International Semantic Web Conference 2014*, pages 33–49. Springer-Verlag, 2014.

[46] Johanna Völker and Mathias Niepert. Statistical schema induction. In Grigoris Antoniou, Marko Grobelnik, Elena Simperl, Bijan Parsia, Dimitris Plexousakis, Pieter De Leenheer, and Jeff Pan, editors, *Proceedings of the 8th Extended Semantic Web Conference*, pages 124–138, Berlin, Heidelberg, 2011. Springer-Verlag.

[47] 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)*, pages 289–293. The Association for Computer Linguistics, 2013.

[48] Ziqi Zhang, Anna Lisa Gentile, Eva Blomqvist, Isabelle Augenstein, and Fabio Ciravegna. Statistical knowledge patterns: Identifying synonymous relations in large linked datasets. In Harith Alani, Lalana Kagal, Achille Fokoue, Paul Groth, Chris Biemann, Josiane Xavier Parreira, Lora Aroyo, Natasha Noy, Chris Welty, and Krzysztof Janowicz, editors, *Proceedings of the 12th International Semantic Web Conference and the 1st Australasian Semantic Web Conference*, Berlin, Heidelberg, 2013. Springer-Verlag.

[49] Ziqi Zhang, Anna Lisa Gentile, and Fabio Ciravegna. Recent advances in methods of lexical semantic relatedness - a survey. *Natural Language Engineering*, 19(4):411–479, 2012.

[50] Lihua Zhao and Ryutaro Ichise. Mid-ontology learning from linked data. In Jeff Pan, Huajun Chen, Hong-Gee Kim, Juanzi Li, Zhe Wu, Ian Horrocks, Riichiro Mizoguchi, and Zhaohui Wu, editors, *Proceedings of the 1st Joint International Conference on The Semantic Web*, pages 112–127, Berlin, Heidelberg, 2011. Springer-Verlag.

[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*, pages 56–63, New York, USA, 2012. ACM.

[52] Lihua Zhao and Ryutaro Ichise. Instance-based ontological knowledge acquisition. In Philipp Cimiano, Oscar Corcho, Valentina Presutti, Laura Hollink, and Sebastian Rudolph, editors, *Proceedings of the 10th Extended Semantic Web Conference*, pages 155–169, Berlin, Heidelberg, 2013. Springer-Verlag.

Appendix

A. Precision and recall obtained with different knowledge confidence models kc

Figure 14 complements Figure 13a by comparing the balance between precision and recall for different variants of the proposed method using the dev and $test$ sets. We use different shapes to represent different kc models and different colours (black, white and grey) to represent different parameter settings for each kc model. It is clear that without kc, the proposed method tends to favour high-recall but perhaps lose too much precision. All kc models 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 by trading off 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 the method 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 the lgt model (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 ~~kc~~.

B. Exploration during the development of the proposed similarity measure

In this section we present some earlier analysis that helped us during the development of the method. These analysis helped us to identify useful features for evaluating relation equivalence, as well as unsuccessful features which we abandoned in the final form of the proposed method. We analyzed the components of the proposed similarity measure, i.e., triple agreement ta and subject agreement 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 in the proposed method. 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.

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 $arg_o(r)$ in data. One approach is to extract classes of their objects and expect a dominant class for all objects of this relation. Thus we started

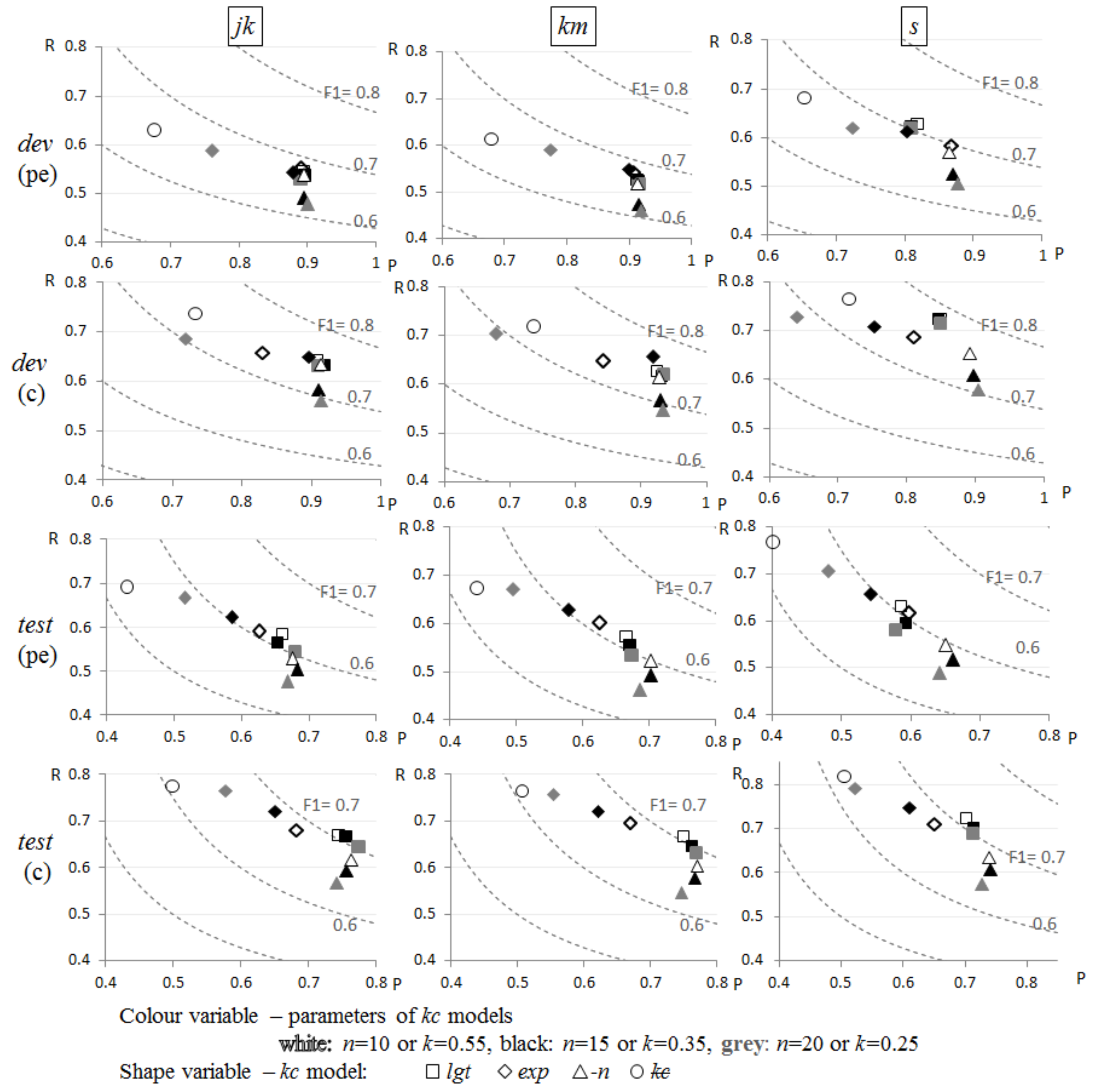

Fig. 14. Balance between precision and recall for the proposed method and its variant forms. pe - pair equivalence, c - clustering. The dotted lines are F1 references.

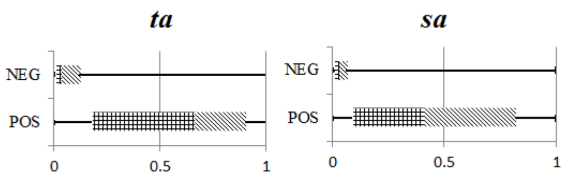

Fig. 15. Distribution of ta and sa scores for true postive and true negative examples in dev.

by querying the DBpedia SPARQL endpoint with the following queries:

```
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 $arg(r)$ 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 Zhao et al. [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:

$$mra(r_1, r_2) = \begin{cases} 0, & \text{if } RG_{r_1,r_2} = \emptyset \\ max\{frac(rg^i_{r_1}) + frac(rg^i_{r_2})\}, & \text{otherwise} \end{cases} \tag{17}$$

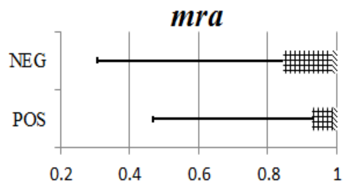

Fig. 16. Distribution of mra scores for true postive and true negative examples in dev.

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 the proposed method. 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.