A Method for Generating Comparison Tables From the Semantic Web

Arnaud Giacometti, Béatrice Markhoff, Arnaud Soulet

International Journal of Data Warehousing and Mining
Volume 18 • Issue 2

A Method for Generating Comparison Tables From the Semantic Web

Arnaud Giacometti, Université de Tours, France
Béatrice Markhoff, Université de Tours, France
Arnaud Soulet, Université de Tours, France*

ABSTRACT

This paper presents Versus, which is the first automatic method for generating comparison tables from knowledge bases of the Semantic Web. For this purpose, it introduces the contextual reference level to evaluate whether a feature is relevant to compare a set of entities. This measure relies on contexts that are sets of entities similar to the compared entities. Its principle is to favor the features whose values for the compared entities are reference (or frequent) in these contexts. The proposal efficiently evaluates the contextual reference level from a public SPARQL endpoint limited by a fair-use policy. Using a new benchmark based on Wikidata, the experiments show the interest of the contextual reference level for identifying the features deemed relevant by users with high precision and recall. In addition, the proposed optimizations significantly reduce the number of required queries for properties as well as for inverse relations. Interestingly, this experimental study also shows that the inverse relations bring out a large number of numerical comparison features.

KEYWORDS

Arnaud Giacometti, Béatrice Markhoff, Comparison Table, LIFAT, Semantic Web, Soulet Arnaud, SPARQL Endpoint, Université de Tours

1. INTRODUCTION

A comparison table (see Table 1) is a double-entry table with entities to compare in columns and comparison features in rows. The comparison table is a particularly useful tool for decision making by isolating the common points and major differences between compared entities. Therefore, this analytical technique is popular in science to compare works, in culture to compare art works or in commerce to compare products or services. For instance, the SocialCompare website[1] uses crowdsourcing to build a varied spectrum of comparison tables. The participants build a list of features for each entity and then, they construct tables by manually selecting the compared entities and the comparison features. The need to compare entities goes far beyond that. DBpedia, that is one of the largest hubs of the Semantic Web, was *established around producing a queryable knowledge graph derived from Wikipedia content that's able to answer questions like "What have Innsbruck and Leipzig in common?"*[2].

DOI: 10.4018/IJDWM.298008 *Corresponding Author

Despite the intensive use of comparison tables in real life, to the best of our knowledge, there is no method to automate the choice of the set of features for a given set of entities to compare. Automating the construction of comparison tables has several advantages. On the one hand, it makes it possible to create objective comparison tables, based on publicly available data. On the other hand, it also makes it possible to build comparison tables for fields where this type of analysis is not carried out due to a lack of expertise.

In this paper, the aim is to automate the process of generating a comparison table for a set of entities by querying a knowledge base (KB). For instance, starting from Ada Lovelace and Alan Turing, an end user wants to obtain a comparison table like the one presented by Table 1, built from Wikidata (the last column is the value of crl, a measure explained later). Beyond persons, the goal is to compare any type of entities, such as places (countries, cities), objects (tapestries, statues), institutions (universities, political parties), events (tournaments, festivals) and so on. Unfortunately, there is no theoretical framework for the design of comparison tables *to determine if a feature is interesting for comparing entities*. This task is non-trivial: according to the experiments carried out, in 17% of the cases a human evaluator does not know whether a feature is interesting or not for comparing the entities presented to him/her (see Section 7 for details). In Table 1, it seems natural to use gender to compare two persons. Besides, specifying that Turing was a member of the Royal Society is interesting only because it is two English scientists who are compared. Thus, the main challenge is to formalize the notion of interesting comparison feature. In addition, it is important to benefit from the huge knowledge bases available on the Semantic Web such as DBpedia (Auer & al., 2007), YAGO (Suchanek & al., 2007) or Wikidata (Vrandečić & Krötzsch, 2014) but this raises a problem of robustness and efficiency. Indeed, these knowledge bases are relatively reliable but they suffer from incompleteness (Razniewski & al., 2016; Zaveri & al., 2016). For this reason, it would be desirable that a feature considered interesting at a given moment remains so despite the subsequent addition of facts. For instance, in Table 1, completing Ada Lovelace's religion should not affect the fact that "religion" is an interesting comparison feature. Furthermore, rather than downloading and centralizing data, it is more relevant to directly query public SPARQL endpoints to build the comparison tables. This has the advantage of guaranteeing an optimal level of values freshness. Nevertheless, the fair-use policy of these public endpoints, which cut off queries that are too expensive, raises optimization needs (Soulet & Suchanek, 2019).

This is the first method for generating a comparison table for a given set of entities, without any information other than the knowledge base. The proposed entity-centric approach leads to the following contributions:

- It defines a new interestingness measure, called *contextual reference level (crl)*, in order to judge if a feature is relevant for comparing a given set of entities. Its principle is to favor the features for which the given entities have values frequently used by other sets of similar entities, called *contexts*.

Table 1. A comparison table of Ada Lovelace and Alan Turing as running example

FeaturesEntities	Ada Lovelace	Alan Turing	*crl*
sex or gender	female	male	*0.908*
spoken language	English	English	*0.472*
member of		Royal Society	*0.205*
field of work	mathematics, computing	mathematics, logic, cryptanalysis, cryptography, computer science	*0.110*
manner of death	natural causes	suicide	*0.100*
religion	?	atheism	*0.015*

- It designs and implements Versus, an algorithm to select the contexts and how to efficiently evaluate the contextual reference level of a feature while minimizing the number of queries to the knowledge base. The idea is to estimate bounds and to interrupt the computation as soon as its interest or its absence of interest is guaranteed.
- It evaluates Versus on a publicly available benchmark, named *Comparison Feature Benchmark* (CFB), that the authors developed to assess the quality of comparison features. CFB relies on comparison tables built from Wikidata and manually evaluated. On this benchmark, the contextual reference level leads, with equal precision, to better recall and better accuracy than the state-of-the-art baselines including automatic facet generation. In addition, the optimized evaluation is significantly faster than a basic implementation, as it requires fewer queries.

This paper is an extended version of Giacometti & al. (2021), with the proofs of crl's and Versus' properties (Sections 4 and 5), and considering not only the entities properties (outgoing links) but also their inverse properties (incoming links) as comparison features. As inverse properties require a way to show a lot of values, a new display is proposed to keep readable the comparison tables (see Section 6). Moreover, all results of the experimental study (in Section 7) are updated.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 formalizes the problem. Sections 4 and 5 introduce the contextual reference level and the Versus algorithm. Section 6 explains the strategy for dealing with inverse properties. The experiments in Section 7 evaluate the approach qualitatively and quantitatively.

2. RELATED WORK

To the best of our knowledge, there is no work to build a comparison table of a set of entities. It could be possible to resort to machine learning methods that learn to rank Resource Description Framework (RDF) properties (Dessi & Atzori, 2016). Unfortunately, it would be difficult to gather feedback specific to the problem of comparison table building, to build a training dataset. Besides, as the ranking of the properties depends on the compared entities (for example, "located in" is relevant for only 84.3% of comparison tables in the used benchmark), this would require the construction of a training dataset of considerable size to cover all cases.

Most techniques that compare two entities in a knowledge base rely on a similarity measure (Anyanwu & al., 2005; d'Amato & al., 2009). Such measures are relevant for estimating the resemblance between two entities, but they do not explicitly give the comparison features (Tversky, 1977). In this direction, Petrova & al. (2017) builds relation paths in knowledge bases between two entities to identify all similarities and differences. Unfortunately, no interestingness measure filters out irrelevant paths leading to too many attributes (including irrelevant ones like identifiers). The tasks closest to ours are the infobox template generation (Saez & Hogan, 2018; Wu & Weld, 2008) and the facet extraction (Feddoul & al., 2019; Hahn & al., 2010; Oren & al., 2006; Tzitzikas & al., 2017). First, an infobox is a set of attribute-value pairs describing an entity. The choice of attributes is based on a template defined for each class (grouping a set of entities). For instance, persons[3] are described by their name, birth date, nationality, highlights and so on. Many templates have been produced collaboratively by Wikipedia contributors, but methods have also been proposed to automatically refine these templates for more specific classes (Saez & Hogan, 2018; Wu & Weld, 2008). More recently, Saez & Hogan (2018) proposed an unsupervised metric-based method that favors frequent attributes with popular values with respect to the PageRank. Unfortunately, as for (Petrova & al., 2017), most of the attributes of the infoboxes describe *one* entity in a singular way and therefore cannot be used to compare *several* entities. For instance, image or notable works are not features that can be shared by two persons.

Second, faceted search consists of restricting a collection of entities by selecting only those with a certain value for a given attribute, called facet (Tzitzikas & al., 2017). A relevant facet has

frequently shared values among the observed entities. Typically, facets are temporal (publication date, birth date), spatial (conference location, birthplace), personal (author, friend), material (subject, color) or energetic (activity, action) attributes (Tzitzikas & al., 2017). There are a few automatic facet extraction methods. For a given class, Hahn & al. (2010) extracts from the infobox templates the attributes whose values are frequently observed. Similarly, Oren & al. (2006) measures the quality of an attribute by favoring frequently used attributes whose values are few and uniformly distributed. Recently, Feddoul & al. (2019) proposes very similar measures to extract facets but a preprocessing method groups the quantitative values (which the authors do not consider in this paper) and a postprocessing method filters out the redundant facets. These methods mainly derive attributes for a limited number of classes containing a lot of entities. Evaluating very similar entities (like Ada Lovelace and Alan Turing) requires considering smaller groups of much more specific entities (e.g., persons employed by the University of Cambridge). Therefore, the main limitation of automatic facet extraction methods is to miss some very specific but very relevant features. Finally, unlike the facets used for navigation, it does not matter if a comparison feature has a lot of values in the knowledge base with an unbalanced distribution.

3. PROBLEM STATEMENT

A knowledge base on a set of relations $\mathcal{R}$ and a set of constants $\mathcal{E}$ (representing entities and values) is a set of facts $\mathcal{K} \subseteq \mathcal{R} \times \mathcal{E} \times \mathcal{E}$. The facts are written in the form $r\left(s,o\right) \in \mathcal{K}$, where r is the relation, s is the subject and o is the object. For instance, religion (Turing, atheism) indicates that Alan Turing was an atheist[4]. Given a relation r, $r^{-1}\left(s,o\right) \in \mathcal{K}$ means that $r\left(o,s\right) \in \mathcal{K}$ where r^{-1} is the inverse property of r. Besides, $r_{\mathcal{K}}\left(s\right)$ (or more simply, $r\left(s\right)$ when the knowledge base $\mathcal{K}$ is clear) is the set of objects associated to the subject s for the relation r in $\mathcal{K}$. For instance, field of work (Turing) returns the set {mathematics, logic, computer science, cryptanalysis, cryptography} in Wikidata.

The notion of comparison table is formalized as follows:

Definition 1 (Comparison table): Given a knowledge base $\mathcal{K}$, the comparison table of a set of entities $E \subseteq \mathcal{E}$ by a set of features $F \subseteq \mathcal{R}$ is a table with $\left|F\right|$ rows and $\left|E\right|$ columns where each cell intersecting a feature f and an entity e contains the values $f\left(e\right) = \left\{o \in \mathcal{E} : f\left(e,o\right) \in \mathcal{K}\right\}$.

Definition 1 limits the comparison features to the relations of the compared entities. With this definition, to use relation paths (Feddoul & al., 2019; Petrova & al., 2017) of greater length (such as "the country of the birthplace"), it would be necessary to enrich the knowledge base. Table 1 illustrates Definition 1 with the comparison table of the set of entities $E = \left\{\text{Lovelace}, \text{Turing}\right\}$ by the set of features $F = \left\{\text{sex or gender}, \text{ spoken language}, \ldots\right\}$. The cell at the intersection of field of work and Turing contains the values field of work (Turing).

An interestingness measure $m : \mathcal{R} \times 2^{\mathcal{E}} \times 2^{\left(\mathcal{R} \times \mathcal{E} \times \mathcal{E}\right)} \rightarrow \left[0,1\right]$ evaluates the interest $m\left(f,E,\mathcal{K}\right)$ of using the relation f as a feature for comparing the entities of E in the knowledge base $\mathcal{K}$.

Definition 2 (Interesting feature): Given a KB $\mathcal{K}$, a set of entities $E \subseteq \mathcal{E}$, an interestingness measure $m : \mathcal{R} \times 2^{\mathcal{E}} \times 2^{\left(\mathcal{R} \times \mathcal{E} \times \mathcal{E}\right)} \rightarrow \left[0,1\right]$ and a threshold $\gamma \in \left[0,1\right]$, an interesting feature $f \in \mathcal{R}$ (for m and γ) satisfies $m\left(f,E,\mathcal{K}\right) \geq \gamma$.

Given a KB $\mathcal{K}$, a set of entities E, an interestingness measure m and a threshold γ, the goal is to extract all the interesting features $F = \left\{f \in \mathcal{R} : m\left(f, E, \mathcal{K}\right) \geq \gamma\right\}$ to build a readable comparison table of E by F.

For this purpose, this paper addresses three challenges. The first challenge consists in defining an interestingness measure that estimates the relevance of a feature from a knowledge base (see Section 4). The second challenge is to evaluate this measure by minimizing the number of SPARQL queries (see Section 5). The third challenge is to display cleverly the cells to make it readable (see Section 6).

4. CONTEXTUAL REFERENCE LEVEL OF A FEATURE

4.1 Definition

Intuitively, to understand and to be able to interpret a comparison table, a feature is interesting if the values describing the compared entities are known by the user. In psychology, it is well known that the user needs at least one reference value to compare two values (Tversky, 1977). In particular, if these values are too rare (or even only characterize one compared entity), the user of the table is unlikely to know them because he has never been confronted with them. Sometimes such values provide valuable information about the entity, but they do not help to compare the entities with each other. For instance, the place of burial of Ada Lovelace is Hucknall Church St Mary Magdalene while that of Alan Turing is Woking Crematorium. There is no particular conclusion to draw from this difference (except perhaps that Alan Turing was atheist unlike Ada Lovelace, but the feature religion is much better suited to underline this point). Of course, this notion of *reference value* is dependent on the compared entities: even if there are only few people who are members of the Royal Society, this feature makes sense to compare two persons employed by the University of Cambridge. This is the key idea of the interestingness measure defined later on: it evaluates the relevance of a feature according to entities that are similar to the compared entities, for instance those "being employed by Cambridge" or those "speaking English" for Ada Lovelace and Alan Turing. Definition 3 formalizes this idea by introducing the notion of *context*:

Definition 3 (Context): Given a set of entities $E \subseteq \mathcal{E}$ and a relation-object couple $\left(r, o\right) \in \mathcal{R} \times \mathcal{E}$ such that $E \subseteq r^{-1}\left(o\right)$, the context $\mathcal{C}$ for E stemming from $\left(r, o\right)$ is the set of entities $r^{-1}\left(o\right) \backslash E$. $\mathbb{C}_E$ denotes the set of all contexts for E.

Intuitively, a context $\mathcal{C}$ is a set of entities that are similar but different from the entities of E with respect to a relation-object couple $\left(r, o\right)$ shared by all the entities of E. For the comparison table provided by Table 1, an example of context is the set of entities having English as spoken language (here, the relation-object couple is (spoken language, English)). Naturally, the classes of the knowledge base are conducive to contexts. For example, all persons (i.e., entities with couple (instance of, human)) could constitute a context for Lovelace and Turing.

Given a set of entities $E \subseteq \mathcal{E}$, a feature $f \in \mathcal{R}$ and a context $\mathcal{C} \in \mathbb{C}_E$, the more the set of values $f\left(e\right)$ for an entity $e \in E$ describes the entities of $\mathcal{C}$, the more this feature f is likely to provide interesting values for a user who wants to compare entities in E. From this intuition, given an entity $e \in E$, the interest of a feature f should increase with the probability of observing the values in $f\left(e\right)$ in the set of values $f\left(s\right)$ of similar entities $s \in \mathcal{C}$: $\Pr\left[f\left(s\right) \cap f\left(e\right) \neq \varnothing \mid s \in \mathcal{C}\right]$. Then, given a set of entities $E = \left\{e_1, \ldots, e_p\right\}$ and a context $\mathcal{C} \in \mathbb{C}_E$, the *contextual reference level* of a feature

f, denoted by $crl_{\mathcal{C}}\left(f,E,\mathcal{K}\right)$, is defined as the probability of observing the values $f\left(e_i\right)$ of at least one entity $e_i \in E$ in the set of values $f\left(s_i\right)$ of similar entities $s_i \in \mathcal{C}$:

$$\begin{aligned} crl_{\mathcal{C}}\left(f,E,\mathcal{K}\right) &= \Pr\left[\left(f\left(s_1\right)\cap f\left(e_1\right)\neq\varnothing\right)\vee\ldots\vee\left(f\left(s_P\right)\cap f\left(e_P\right)\neq\varnothing\right) \mid s_1\in\mathcal{C},\ldots,s_P\in\mathcal{C}\right] \\ &= \Pr\left[\left(\exists e_i\in E\right)\left(f\left(s_i\right)\cap f\left(e_i\right)\neq\varnothing\right) \mid s_i\in\mathcal{C}\right] \end{aligned}$$

It is indeed a probability because if a similar entity s_i shares features with several entities in E, it is counted only once. In practice, entities belong to several relevant contexts. For example, for Ada Lovelace and Alan Turing, there are contexts stemming from four couples (see below for details): (field of work, mathematics), (employer, Univ. of Cambridge), (occupation, computer scientist) and (spoken language, English). For this reason, the definition of $crl_{\mathcal{C}}\left(f,E,\mathcal{K}\right)$ considers a set of contexts $\mathcal{C}=\left\{\mathcal{C}_1,\ldots,\mathcal{C}_K\right\}$ as follows:

Definition 4 (Contextual reference level): Given a set of entities $E=\left\{e_1,\ldots,e_P\right\}\subseteq\mathcal{E}$ and a set of contexts $\mathcal{C}=\left\{\mathcal{C}_1,\ldots,\mathcal{C}_K\right\}\subseteq\mathbb{C}_E$, the contextual reference level of a feature F is defined as:

$$crl_{\mathcal{C}}\left(f,E,\mathcal{K}\right)=\Pr\left[\left(\exists e_i\in E\right)\left(\exists k\in\left[1...K\right]\right)\left(f\left(s_i^k\right)\cap f\left(e_i\right)\neq\varnothing\right) \mid s_i^k\in\mathcal{C}_k\right]$$

It is important to note that the compared entities in E play a very strong role in this definition because they limit the choice of $\mathcal{C}$ in the set of potential contexts $\mathbb{C}_E$. The fourth column of Table 1 indicates the *crl* of each feature computed from Wikidata in the four contexts mentioned above. For instance, 0.908 corresponds to the probability of observing the value female or male (respectively stemming from Ada Lovelace or Alan Turing for e_i) as sex or gender of an entity s_i that is a mathematician or an employee of Cambridge or a computer scientist or an English speaker. With Definition 4, it is possible to directly calculate the *crl* of a feature with a SPARQL query. However, this statistical query would often be too costly not to be interrupted by the fair-use policy of public SPARQL endpoints (Soulet & Suchanek, 2019). Nevertheless, this definition implicitly assumes that the entities s_i^k are identically and independently drawn in the different contexts $\mathcal{C}_k$. With this i.i.d. assumption, the following property rewrites the contextual reference level:

Property 1: Given a set of entities $E\subseteq\mathcal{E}$, a set of contexts $\mathcal{C}\subseteq\mathbb{C}_E$ and a feature $f\in\mathcal{R}$, one has:

$$crl_{\mathcal{C}}\left(f,E,\mathcal{K}\right)=1-\prod_{\mathcal{C}\in\mathcal{C}}\prod_{e\in E}\left(1-\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing) \mid s\in\mathcal{C}\right]\right)=1-\prod_{\mathcal{C}\in\mathcal{C}}\left(1-crl_{\mathcal{C}}\left(f,E,\mathcal{K}\right)\right)$$

Proof: Let $\mathcal{K}$ be a knowledge base, $f\in\mathcal{R}$ be a feature and $E\subseteq\mathcal{E}$ be a set of entities, one has the following equalities:

$$\begin{aligned} crl_{\mathcal{C}}\left(f,E,\mathcal{K}\right) &= \Pr\left[\left(\exists e_i\in E\right)\left(\exists k\in\left[1...K\right]\right)\left(f\left(s_i^k\right)\cap f\left(e_i\right)\neq\varnothing\right) \mid s_i^k\in C_k\right] \\ &= 1-\prod_{C_k\in\mathcal{C}}\left(1-\Pr\left[\left(\exists e_i\in E\right)\left(f\left(s_i^k\right)\cap f\left(e_i\right)\neq\varnothing\right) \mid s_i^k\in C_k\right]\right) \\ &= 1-\prod_{C_k\in\mathcal{C}}\left(1-\left(1-\prod_{e_i\in E}\left(1-\Pr\left[f\left(s_i^k\right)\cap f\left(e_i\right)\neq\varnothing \mid s_i^k\in C_k\right]\right)\right)\right) \\ &= 1-\prod_{C_k\in\mathcal{C}}\prod_{e_i\in E}\left(1-\Pr\left[f\left(s_i^k\right)\cap f\left(e_i\right)\neq\varnothing \mid s_i^k\in C_k\right]\right) \end{aligned}$$

The Morgan's law is used for the first two rows. Then, simplifying the expression proves the soundness of Property 1.

Interestingly, each probability $\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing\middle|s\in\mathcal{C}\right]$ can easily be calculated independently by a low-cost SPARQL query. The experimental section shows that in practice, the error rate of this kind of query is under 0.5%. In addition, considering Property 1, it is easy to see that the *crl* increases with the probability $\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing\mid s\in\mathcal{C}\right]$ and that its range is $\left[0,1\right]$. The *crl* of the feature f is zero when no entity among those of the contexts $\mathcal{C}$ has a common value with the entities of E. Conversely, $crl_{\mathcal{C}}\left(f,E,\mathcal{K}\right)$ is equal to 1 as soon as a value in $f\left(e\right)$ is shared by all the entities of at least one context $\mathcal{C}$.

4.2 Quality Criteria Analysis

Properties 2-4 present three quality criteria that a well-behaved interestingness measure for evaluating features should satisfy. First, the following property proves that the contextual reference level is monotone with respect to contexts:

Property 2: Given a KB $\mathcal{K}$, a feature f and a set of entities E, one has $crl_{\mathcal{C}}\left(f,E,\mathcal{K}\right)\leq crl_{\mathcal{C}'}\left(f,E,\mathcal{K}\right)$ if the two sets of contexts satisfy $\mathcal{C}\subseteq\mathcal{C}'\subseteq\mathbb{C}_{E}$.

Proof: Let $\mathcal{K}$ be a knowledge base, $f\in\mathcal{R}$ be a feature and $E\subseteq\mathcal{E}$ be a set of entities. Let $\mathcal{C}\subseteq\mathbb{C}_{E}$ and $\mathcal{C}'\subseteq\mathbb{C}_{E}$ be two sets of contexts such that $\mathcal{C}\subseteq\mathcal{C}'$, one has the following equalities:

$$\begin{aligned}
crl_{\mathcal{C}'}\left(f,E,\mathcal{K}\right) &= 1-\prod_{C\in\mathcal{C}'}\prod_{e\in E}\left(1-\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing\mid s\in C\right]\right)\\
&= 1-\prod_{C\in\mathcal{C}}\prod_{e\in E}\left(1-\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing\mid s\in C\right]\right)\\
&\quad\times\prod_{C\in\left(\mathcal{C}'\setminus\mathcal{C}\right)}\prod_{e\in E}\left(1-\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing\mid s\in C\right]\right)\\
&\geq 1-\prod_{C\in\mathcal{C}}\prod_{e\in E}\left(1-\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing\mid s\in C\right]\right)\\
&\geq crl_{\mathcal{C}}\left(f,E,\mathcal{K}\right)
\end{aligned}$$

Interestingly, the addition of a new context favors the emergence of new interesting features (e.g., if a new relation is added to the knowledge base). However, this may also raise problems of redundancy between contexts (see below). The following property goes further by showing that contextual reference level is also robust against incompleteness for the feature f:

Property 3: Given two KBs $\mathcal{K}$ and $\mathcal{K}'$, a set of contexts $\mathcal{C}\subseteq\mathbb{C}_{E}$ and a feature F such that $f_{\mathcal{K}}\left(e\right)\subseteq f_{\mathcal{K}'}\left(e\right)$ for each $e\in E$, one has $crl_{\mathcal{C}}\left(f,E,\mathcal{K}\right)\leq crl_{\mathcal{C}}\left(f,E,\mathcal{K}'\right)$.

Proof: Let $\mathcal{C}\subseteq\mathbb{C}_{E}$ be a set of contexts, $f\in\mathcal{R}$ be a feature and $\mathcal{K}$ and $\mathcal{K}'$ be two knowledge bases such that $f_{\mathcal{K}}\left(e\right)\subseteq f_{\mathcal{K}'}\left(e\right)$ for every entity $e\in E$, one has the following equalities:

$$\begin{aligned} crl_{\mathcal{C}}\left(f,E,\mathcal{K}'\right) &= 1-\prod_{C\in\mathcal{C}}\prod_{e\in E}\left(1-\Pr\left[f_{\mathcal{K}'}\left(s\right)\cap f_{\mathcal{K}'}\left(e\right)\neq\varnothing \mid s\in C\right]\right) \\ &\geq 1-\prod_{C\in\mathcal{C}}\prod_{e\in E}\left(1-\Pr\left[f_{\mathcal{K}}\left(s\right)\cap f_{\mathcal{K}}\left(e\right)\neq\varnothing \mid s\in C\right]\right) \\ &\geq crl_{\mathcal{C}}\left(f,E,\mathcal{K}\right) \end{aligned}$$

The passage from the first equation to the second one is due to the fact that $\left(f_{\mathcal{K}}\left(s\right)\cap f_{\mathcal{K}}\left(e\right)\right)\subseteq\left(f_{\mathcal{K}'}\left(s\right)\cap f_{\mathcal{K}'}\left(e\right)\right)$. Consequently, $\Pr\left[f_{\mathcal{K}'}\left(s\right)\cap f_{\mathcal{K}'}\left(e\right)\neq\varnothing \mid s\in C\right]$ is greater or equal to $\Pr\left[f_{\mathcal{K}}\left(s\right)\cap f_{\mathcal{K}}\left(e\right)\neq\varnothing \mid s\in C\right]$.

This property underlines that the value of *crl* is always underestimated when some facts are missing. If new facts are added in the knowledge base, then the contextual reference level of a feature can only increase (if the context $\mathcal{C}$ remains unchanged). For this reason, the extracted features will remain interesting for crl if the knowledge base is completed. In Table 1, the feature religion was selected despite the lack of value for Ada Lovelace. Whatever the value could be stated, this feature would remain interesting for crl.

Finally, the next property proves that the contextual reference level of a feature f is zero when it is an identifier (i.e., an injective function $f\left(x\right)=f\left(y\right)\Rightarrow x=y$):

Property 4: Given a set of entities E and a set of contexts $\mathcal{C}$, one has $crl_{\mathcal{C}}\left(f,E,\mathcal{K}\right)=0$ for any feature F that is an identifier.

Proof: Let $\mathcal{K}$ be a knowledge base, $E\subseteq\mathcal{E}$ be a set of entities and $\mathcal{C}\subseteq\mathbb{C}_{E}$ be a set of contexts. Let $f\in\mathcal{R}$ be an identifier, i.e., $f\left(e_1\right)\cap f\left(e_2\right)\neq\varnothing\Rightarrow e_1=e_2$. For any entity $e\in E$ and any context $C\in C$, one has $e\notin C$ (see Definition 3 in the submission). Consequently, $f\left(s\right)\cap f\left(e\right)=\varnothing$ for $s\in\mathcal{C}$ and $\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing \mid s\in C\right]$ equals zero.

Interestingly, an identifier F is not relevant for comparing entities because by definition all values of F uniquely identifies an entity. For instance, for a set of countries, the property GeoNames ID is not an interesting feature w.r.t.

5. VERSUS: A METHOD FOR EXTRACTING INTERESTING FEATURES

5.1 Overview

The overall idea is to analyze each relation f that describes at least one entity in E to determine whether it is an interesting feature in $\mathcal{K}$: $crl_{\mathcal{C}}\left(f,E,\mathcal{K}\right)\geq\gamma$. Algorithm 1 sketches this process. First, the set F that will contain all the interesting features is initialized with the empty set (Line 1) and the set of all the candidate relations $\mathcal{R}_{E}$ gathers the relations that describe at least one entity in E (Line 2). After, each relation in $\mathcal{R}_{E}$ is separately processed (Lines 3-6). Line 4 selects the set of contexts $\mathcal{C}\subseteq\mathbb{C}_{E}$ without considering the relation f (see Algorithm 2). This set of contexts is immediately used by Algorithm 3 in order to decide whether the relation f is an interesting feature for the entities in E. If f is really interesting for crl, it is added to the set of interesting features F. Finally, this set is returned at Line 7.

The rest of this section details Lines 4 and 5 based respectively on Algorithms 2 and 3 Context selection gives the method for selecting the set of contexts. Of course, this choice is decisive in the calculation of the contextual reference level. Next sections also present an efficient algorithm for

Algorithm 1. Versus: extracting the set of interesting features w.r.t. crl

Input: A knowledge base $\mathcal{K}$, a set of entities $E \subseteq \mathcal{E}$ and a threshold γ
Output: The set of interesting features $F \subseteq \mathcal{R}$
1: $F := \emptyset$
2: $\mathcal{R}_E := \{r \in \mathcal{R} : e \in E \wedge r(e,s) \in \mathcal{K}\}$
3: **for all** $f \in \mathcal{R}_E$ **do**
4: Select the set of contexts$\mathcal{C}$for the entities E and the feature f with Algorithm 2
5: **if** $crl_{\mathcal{C}}(f,E,\mathcal{K}) \geq \gamma$ (using Algorithm 3) **then** $F := F \cup \{f\}$
6: **end for**
7: **return** F

evaluating the contextual reference level. Indeed, the naive evaluation of the contextual reference level is expensive, as for each feature, it requires to calculate $\left|\mathcal{C} \times E\right|$ queries for the numerators and $\left|\mathcal{C}\right|$ queries for the denominators (see Definition 4).

5.2 Context Selection

This step aims to select a small number of relevant contexts among all the contexts of $\mathbb{C}_E$ that may be redundant. Indeed, in the case where a large number of contexts in $\mathcal{C}$ are correlated, the contextual reference level might be abnormally overestimated (see Property 2). For example, since all employees of the University of Cambridge are necessarily humans, the context stemming from (instance of, human) does not provide additional information, but it increases the contextual reference level. It is however important to keep a set of contexts that cover all the specificities of the entities similar to E: $\cap\mathbb{C}_E$. For example, the context stemming from (occupation, computer scientist) is important because it distinguishes Ada Lovelace and Alan Turing from mathematicians at the University of Cambridge who have not contributed in computer science. In this way, one of the smallest sets of contexts $\mathcal{C}^* \subseteq \mathbb{C}_E$ is chosen so that it characterizes the same set of entities as $\mathbb{C}_E$ by intersecting: $\mathcal{C}^* \in \text{argmin}_{\mathcal{C} \subseteq \mathbb{C}_E} \left\{\left|\mathcal{C}\right| : \cap\mathcal{C} = \cap\mathbb{C}_E\right\}$. The exact resolution of this problem is NP-hard and it would require a large number of knowledge base queries. Therefore, Algorithm 2 proposes a heuristic algorithm, which eliminates superfluous contexts from the smallest one to the largest one.

Given a knowledge base $\mathcal{K}$, a set of entities E and a feature f, Algorithm 2 returns a set of contexts $\mathcal{C}$. Line 1 builds the set of contexts $\mathbb{C}_E$ except it excludes the context stemming from the feature f (i.e., $r \neq f$). The contexts are then sorted from the smallest to the largest (Line 2) to favor the removal of overly general contexts. The loop (Lines 3-5) iterates over each context $\mathcal{C}_i$ starting with the smallest one. Line 4 tests whether the intersection of contexts without $\mathcal{C}_i$ gives the same set of entities as with $\mathcal{C}_i$. If this is the case, it means that this context does

Algorithm 2. Selecting a set of contexts

Input: A knowledge base $\mathcal{K}$, a set of entities $E \subseteq \mathcal{E}$ and a feature $f \in \mathcal{R}$
Output: A set of contexts $\mathcal{C} \subseteq \mathbb{C}_E$
1: $\mathcal{C} := \{r^{-1}(o) \backslash E : r \in (\mathcal{R} \backslash \{f\}) \wedge (\forall e \in E)(r(e,o) \in \mathcal{K})\}$
2: Sort the contexts of$\mathcal{C}$by ascending cardinality
3: **for all** context $\mathcal{C}_i \in \langle C_1, \ldots, C_n \rangle$ **do**
4: **if** $\bigcap(\mathcal{C} \setminus \mathcal{C}_i) = \bigcap\mathcal{C}$ **then** $\mathcal{C} := \mathcal{C} \setminus \mathcal{C}_i$
5: **end for**
6: **return** $\mathcal{C}$

not provide any specificity and it is discarded from $\mathcal{C}$. Once the loop is completed, Line 6 returns the set of non-redundant contexts.

Table 2 presents the relation-object couples (r, o) from which contexts are computed considering Ada Lovelace and Alan Turing. After having been sorted by ascending cardinality in Wikidata (i.e., $\left|r^{-1}(o) \backslash E\right|$), the two redundant contexts were eliminated by Lines 3-7 of Algorithm 2. For example, the restriction "instance of human" does not delete any entity among those belonging to all other contexts. It is important to note that the interest of an approach centered on entities is to consider contexts that do not depend only on classes (i.e., there are other relations than instance of). However, the number of contexts in $\mathcal{C}$ remains reasonable in practice (9 at most in all experiments, see Figure 1). Most often, the iteration of Lines 3-5 removes few contexts (1.61 in average), but in some cases, many redundant contexts are eliminated (for example, 168 in the most extreme case).

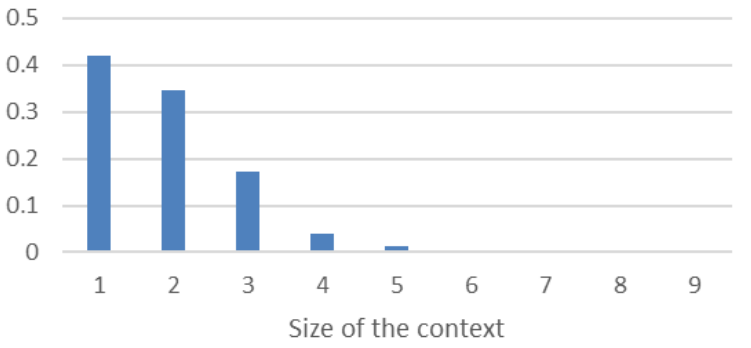
5.3 Efficient Evaluation of the Contextual Reference Level

Rather than calculating the exact contextual reference level of a feature, the idea is to do a partial calculation of this value in order to only determine whether $crl_c(f, E, \mathcal{K})$ is greater than γ. It is easy to see that the complement to 1 of the contextual reference level (i.e., $1 - crl_c(f, E, \mathcal{K})$)

Table 2. Relation-object couples common to Ada Lovelace and Alan Turing

Relation r	Object o	$\left\|r^{-1}(o) \backslash E\right\|$
field of work	mathematics	2,018
employer	Univ. of Cambridge	3,129
occupation	computer scientist	7,943
described by source	Obalky knih.cz	47,563
spoken language	English	165,714
instance of	human	6,389,426

Figure 1. Distribution of comparison tables w.r.t their context size



decreases with each multiplication by a factor of the form $\left(1-\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing\mid s\in\mathcal{C}\right]\right)$. With this observation, it is possible to derive a lower bound for the contextual reference level. In the process of calculation, when this lower bound exceeds the threshold γ, one has the guarantee that $crl_c\left(f,E,\mathcal{K}\right)\geq\gamma$. Conversely, it is possible to derive an upper bound of the contextual reference level by using $\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing,s\in\mathcal{E}\right]$ as an upper bound of the joint probability $\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing,s\in\mathcal{C}\right]$. The following property formalizes these two bounds:

Property 5: Given a knowledge base $\mathcal{K}$, the contextual reference level of a feature f for the entities E is bounded for any $\mathcal{S}\subseteq\mathcal{C}\times E$:

$$\begin{aligned} crl_c\left(f,E,\mathcal{K}\right) &\geq 1-\prod_{(\mathcal{C},e)\in\mathcal{S}}\left(1-\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing\mid s\in\mathcal{C}\right]\right)\\ &\leq 1-[\prod_{(\mathcal{C},e)\in\mathcal{S}}\left(1-\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing\mid s\in\mathcal{C}\right]\right)\\ &\times\prod_{(\mathcal{C},e)\in(\mathcal{C}\times E)\setminus\mathcal{S}}\left(1-\frac{\min\left\{\left|\left\{s\in\mathcal{E}:f\left(s\right)\cap f\left(e\right)\neq\varnothing\right\}\right|,\left|\mathcal{C}\right|\right\}}{\left|\mathcal{C}\right|}\right)\end{aligned}$$

optimistic factor

Proof: Let $\mathcal{K}$ be a knowledge base, $f\in\mathcal{R}$ be a feature and $E\subseteq\mathcal{E}$ be a set of entities. Let $\mathcal{S}\subseteq\mathcal{C}\times E$ be a set, one has:

$$\begin{aligned} crl_c\left(f,E,\mathcal{K}\right) &= 1-\prod_{(C,e)\in\mathcal{S}}\left(1-\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing\mid s\in C\right]\right)\\ &\times\prod_{(C,e)\in(\mathcal{C}\times E)\setminus\mathcal{S}}\left(1-\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing\mid s\in C\right]\right)\end{aligned}$$

As:

$$\prod_{(C,e)\in(\mathcal{C}\times E)\setminus\mathcal{S}}\left(1-\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing\mid s\in C\right]\right)\in\left[0,1\right]$$

it proves the lower bound. Now, let us consider the upper bound. It is easy to see that:

$$\begin{aligned}\frac{\min\left\{\left|\left\{s\in\mathcal{E}:f\left(s\right)\cap f\left(e\right)\neq\varnothing\right\}\right|,\left|C\right|\right\}}{\left|C\right|} &\geq\frac{\left|\left\{s\in C:f\left(s\right)\cap f\left(e\right)\neq\varnothing\right\}\right|}{\left|C\right|}\\ &\geq\Pr\left[f\left(s\right)\cap f\left(e\right)\neq\varnothing\mid s\in C\right]\end{aligned}$$

leading to overestimate the contextual reference level.

Algorithm 3 benefits from these bounds for efficiently evaluating if $crl_c\left(f,E,\mathcal{K}\right) \geq \gamma$. More precisely, Lines 1 and 2 respectively initialize the product p and the optimistic factor o discussed above by considering all the couples in $\mathcal{C}\times E$. The loops of Lines 3 and 4 enumerate the different entities $e \in E$ and the different contexts $\mathcal{C} \in \mathcal{C}$. At each iteration, Line 5 refines the calculation of p taking into account the probability $\Pr\left[f\left(s\right)\cap f\left(e\right) \neq \varnothing \mid s \in \mathcal{C}\right]$ while Line 7 updates o. If the current contextual reference level is higher than the threshold γ, Line 6 returns $true$ because $1-p$ is a pessimistic approximation of the final contextual reference level. Conversely, Line 8 returns $false$ when the upper bound $1-p\times o$ is lower than γ.

Let us illustrate Algorithm 3 with the computation of the contextual reference level of two features for Ada Lovelace and Alan Turing with a threshold $\gamma = 0.01$ illustrated by Table 3. The probability of having an entity with a natural death (like Ada Lovelace) among those who studied mathematics is 0.025. From this evaluation, it is certain that manner of death is an interesting feature because its exact contextual reference level exceeds the lower bound $1-\left(1-0.025\right)$ which is higher than the threshold γ. In this case, this avoids the evaluation of 7 queries that would have been necessary for the exact calculation of the contextual reference level. For the feature student of, the optimistic factor after the first evaluation is equal to 0.998. It is therefore sure that the contextual reference level of this feature is at most equal to $1-\left(1-0\right)\times 0.998 = 0.002$ which is lower than the threshold γ.

Algorithm 3. Computing the contextual reference level of a relation

Input: A knowledge base $\mathcal{K}$, a set of entities $E \subseteq \mathcal{E}$, a threshold γ, a set of contexts$\mathcal{C}$and a relation f
Output: Return true if the relation f is interesting i.e., $crl_{\mathcal{C}}(f,E,\mathcal{K}) \geq \gamma$

1: $p := 1$
2: $o := \prod_{(\mathcal{C},e)\in\mathcal{C}\times E}\left(1-\frac{\min\{|\{s\in\mathcal{E}:f(s)\cap f(e)\neq\emptyset\}|,|\mathcal{C}|\}}{|\mathcal{C}|}\right)$
3: **for all** $e \in E$ **do**
4: **for all** $\mathcal{C} \in \mathcal{C}$ **do**
5: $p := p \times (1-(|\{s \in \mathcal{C} : f(s)\cap f(e)\neq\emptyset\}|)/(|\mathcal{C}|))$
6: **if** $1-p \geq \gamma$ **then** $true$
7: $o := o/\left(1-\frac{\min\{|\{s\in\mathcal{E}:f(s)\cap f(e)\neq\emptyset\}|,|\mathcal{C}|\}}{|\mathcal{C}|}\right)$
8: **if** $1-p\times o < \gamma$ **then** $false$
9: **end for**
10: **end for**
11: **return** $false$

Table 3. Computation of crl of two features for Ada Lovelace and Alan Turing

	Entities E	**Contexts** $\mathcal{C}$			
		Field of work	**Employer**	**Occupation**	**Spoken language**
manner of death	Lovelace	**0.025**	0.018	0.017	0.038
	Turing	0.002	0.001	0.002	0.003
student of	Lovelace	**0,000**	0,000	0,000	0,001
	Turing	0,001	0,000	0,000	0,001

Again, the contextual reference level computation can be interrupted (Line 8) avoiding the evaluation of 7 queries.

6. DISPLAYING COMPARISON TABLES WITH INVERSE PROPERTIES

In a Web knowledge base, an entity is described not only by its properties (outgoing links) but also by its incoming links which show its role as object of properties. Versus can easily deal with both properties and inverse properties. In Table 1, the comparison table is particularly readable because each cell contains a small number of values (maximum 5 values, for the fields of work of Alan Turing). Unfortunately, sometimes there are so many values in a cell that you cannot display them all. For example, generating a comparison table taking into account inverse properties of Paris and New York results in several comparison features whose number of values is very large (Table 4).

As can be seen in Table 4, the inverse properties are particularly conducive to the emergence of comparison features with cells containing many values. Of course, it would be possible to show only part of these values. It could be possible to select the k most interesting values according to a criterion to be defined. However, the intuition is that when a feature leads to a large number of values the comparison between the entities is more quantitative than qualitative. For example, the end user does not want to compare 2 by 2 the entities with a work location in Paris or New York, but just the quantity of these work locations for the two cities. The emergence of these numerical features is even particularly interesting for carrying out a compensatory analysis.

7. EXPERIMENTS

After presenting the evaluation benchmark in the next subsection, the experiments carried out aim to answer the following three questions: Does the contextual reference level really isolate the best features? (Q1); What is the gain of the optimized evaluation? (Q2) and What is the impact of inverse properties? (Q3). Versus is implemented in Java using the Jena library to query the public Wikidata SPARQL endpoint. Versus was run on Windows 10 with an Intel core i7 processor and 32 GB of RAM. Due to the few operations performed on the client side, the execution times correspond essentially to the processing time of SPARQL queries on the server side[5]. Although execution times vary with server load and available data, they shape the behavior of the approaches. Note that the source code of Versus, the evaluation tool and the results are available on the website https://lovelace-vs-turing.com and on github.com/asoulet/versus.

7.1 Comparison Feature Benchmark (CFB)

As comparison table generation is a new problem, the authors had to develop a benchmark, named *Comparison Feature Benchmark* (CFB). Its twofold objective is to constitute a reference dataset to compare the speed of different approaches and a gold standard to assess the quality of the discovered

Table 4. Some features of the comparison table of Paris and New York with a lot values

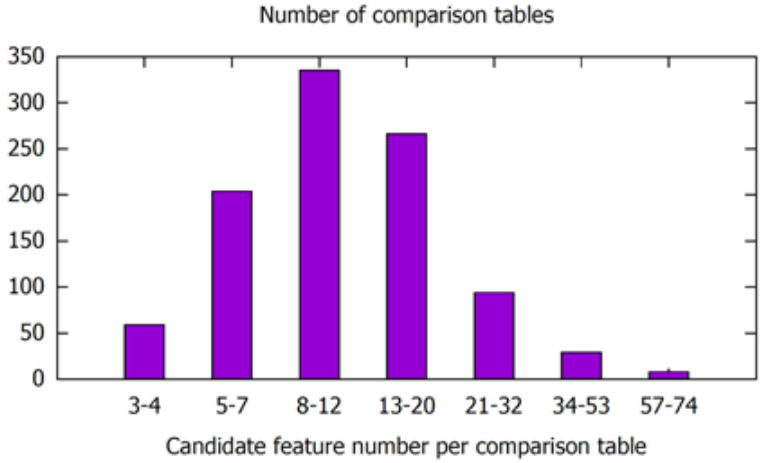
Features	Nb val. for Paris	Nb val. for New York
Inverse of narrative location (P840)	2640	4893
Inverse of work location (P937)	20404	2154
Inverse of residence (P551)	1214	2379
Inverse of filming location (P915)	668	1289
…	…	…

comparison features. This section starts by describing the method to select from Wikidata the sets of entities to be compared with their candidate comparison features. For simplicity, only *pairs* of entities are considered. Then, this section explains how the quality of the candidate comparison features have been manually evaluated.

First, 1,000 types T_i ($i \in [1..1000]$) are randomly drawn from Wikidata that have between 10k and 1k instances. This random sample guarantees to cover a wide variety of entities (person, place, objects, events and so on) in order to best reflect Wikidata diversity. Second, for each type T_i, the two entities e_i^1 and e_i^2 that have the highest degree of incoming facts (i.e., maximizing $\deg(e) = |\{s \in \mathcal{E} : r(s,e) \in \mathcal{K} \wedge e \in T_i\}|$) are selected. This in-degree ranking favors popular entities of the type T_i. For instance, the entities Paris (Q90) and London (Q84) are selected for the type city (Q515). Then, for each pair of entities $E_i = \{e_i^1, e_i^2\}$, the set F_i contains all the relations $r_j \in \mathcal{R}$ that have URI as objects and that are a direct property of Wikidata (by using the prefix https://www.wikidata.org/prop/direct/) and $r_j(e_i^1)$ or $r_j(e_i^2)$ is not empty (note that the inverse properties are not considered in this benchmark). Thus, F_i is the set of candidate comparison features to compare entities in E_i. Finally, for each pair of entities $E_i = \{e_i^1, e_i^2\}$, the benchmark CFB stores all the facts $r_j(e_i^k, o_i^k)$ ($k \in \{1,2\}$) where $r_j \in F_i$ and o_i^k is an object randomly drawn from the values in $r_j(e_i^k)$ (if $r_j(e_i^k)$ is the empty set, then o_i^k is null). For each type T_i ($i \in [1...1000]$) this process builds a comparison table with $|F_i|$ rows and two columns to compare e_i^1 and e_i^2. Figure 2 indicates the number of comparison tables with the number of candidate comparison features. Note that the slices are non-linear.

Second, 1,195 candidate features (out of the 11,852, or about 10%) were drawn at random and evaluated manually by one of the 6 evaluators. Each time, it was asked if the candidate feature was relevant to compare the pair of entities (by selecting in CFB the facts and the evaluator can answer "No", meaning that the feature is not relevant (44.9% of the evaluations), "Yes" (37.9%) or "I don't know" (17.2%). Only 80 evaluations were common, of which 74 agreed. It leads to a Cohen's kappa coefficient of 0.832 that corresponds to an almost perfect agreement (Landis

Figure 2. Distribution of comparison tables



& Koch, 1977). This evaluation was carried out using the tool available at the website https://lovelace-vs-turing.com (see Figure 3).

7.2 Q1: Quality of the Extracted Features

Figures 4 and 5 respectively report the precision-recall results (ignoring "I don't know" evaluations) and the number of comparison features. First, this experiment benefits from the CFB benchmark described in the previous section for comparing the contextual reference level used by Versus (denoted by) with the metric used by automatic facet generation (Oren & al., 2006) (denoted by Facet). For the facet-oriented metric, the type of the two entities (see above) is used to define the collection on which the metric is computed. There are also two baselines: the all method (Petrova & al., 2017) that selects all the candidate features of the benchmark and the infobox method that selects all the candidate features present in at least one of the Wikipedia infoboxes of the entities . Figure 4 reports the precision, recall and accuracy for these methods by varying the minimum threshold for crl and Facet. For the reasons mentioned in related work, the precision of all and infobox, less than 50%, is catastrophic. When the precision of Facet is better than that of the recall of Facet is dramatically low (less than 20 features are extracted). Overall, the contextual reference level is much better than

Figure 3. Evaluation tool used by annotators

Figure 4. Precision, recall and accuracy for crl, Facet, infobox and all

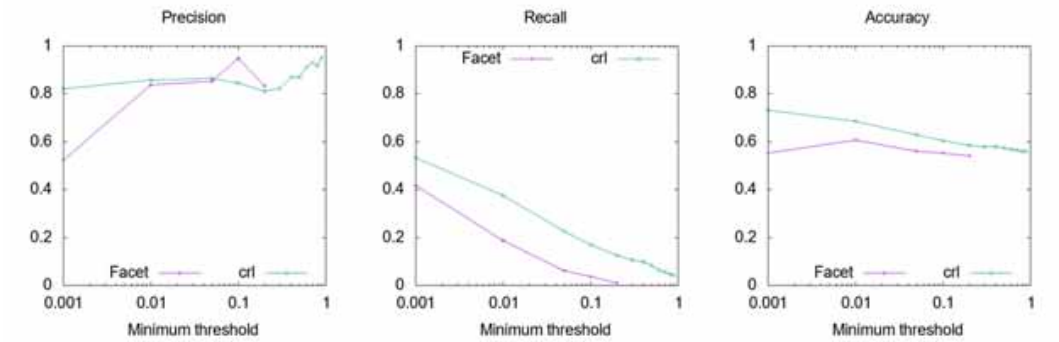
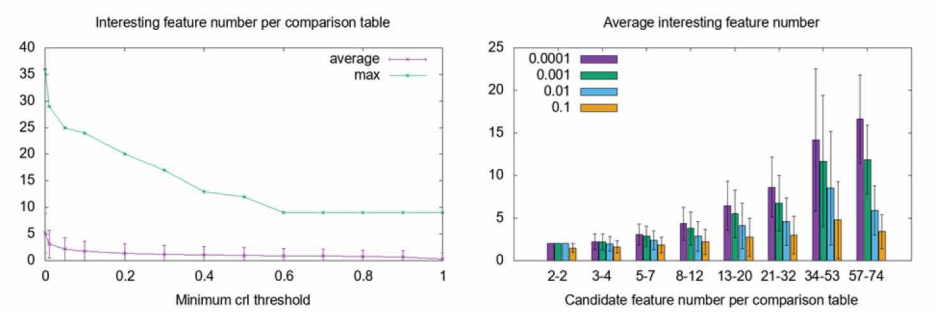


Figure 5. Number of interesting features per comparison table



facet-oriented metric with comparable precision but higher recall and higher accuracy. This result is not surprising because, unlike Facet, Versus brings out features specific to the two compared entities.

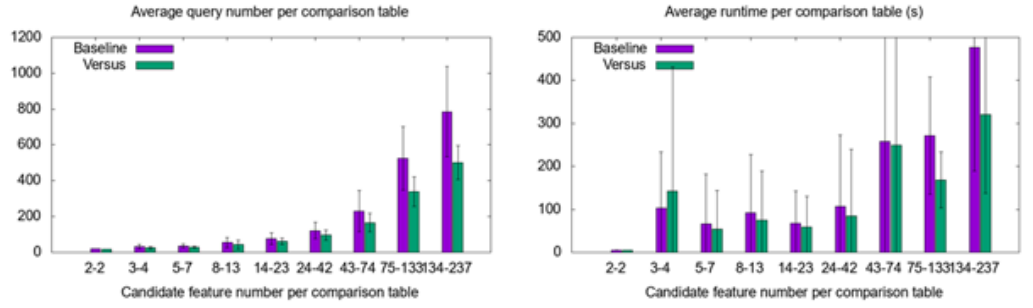
The precision of the contextual reference level, always above 76%, is generally high with regard to the baselines whose precision is less than 50%. Interestingly, this precision increases with the minimum contextual reference level threshold (from 76% for $\gamma = 0.0001$ to 86% for $\gamma = 0.05$). This demonstrates the ability of *crl* to isolate the most relevant features. However, the recall decreases very quickly with the minimum contextual reference level threshold. This is explained by the decrease in the number of interesting features with γ as shown in Figure 5. With $\gamma = 0.1$, the left-hand side graph indicates that a comparison table contains only three features on average. However, the right-hand side graph shows strong disparities depending on the initial number of candidate features describing the entities (note that slices are non-linear). In practice, to have a good compromise, it seems appropriate to set γ with a value less than 0.1.

7.3 Q2: Efficiency of the Method

This section assesses the efficiency gain of the optimized method (Versus benefiting from Property 5) with a baseline where the exact value of the contextual reference level is calculated (baseline based on Property 1). Unlike the previous section where the benchmark prevented the use of inverse properties, this new experiment is now exploiting them. It is important to note that the Facet method (that only requires 8,061 ms per comparison table) is much faster than Versus.

Figure 6 indicates the number of SPARQL queries required to build the 1,000 comparison tables of the benchmark. For $\gamma = 0.01$, the left-hand side figure shows that the number of queries increases linearly with the number of candidate features to be tested. This result is expected because the number

Figure 6. Efficiency of Versus vs baseline w.r.t. query number and execution time



of contexts (between 1 and 7) is relatively independent of the number of features. Overall, Versus is always efficient.

The right-hand side chart in Figure 6 details the average execution time to construct a comparison table according to the number of features. Unlike the number of queries, the execution time of the construction of a comparison table does not increase linearly with the number of features. In addition, the standard deviations are very high. This phenomenon is explained by the fact that not all queries have the same complexity. For instance, it is more expensive to evaluate a query with the person as context than with the country as context because the latter contains fewer entities. In particular, very few queries ( ) have an execution time that exceeds the limit of the fair-use policy of Wikidata and fail as shown by the error rates on Figure 7.

7.4 Q3: Qualitative Analysis of the Features

This section briefly analyzes the impact of adding the inverse properties as potential comparison features. As explained in Section 6, a comparison feature is transformed into a numerical feature if an entity has at least 10 values for this feature.

Of course, the addition of the inverse properties significantly increases the number of comparison features, which is interesting to give the end user additional information to compare the entities. Nevertheless, it should be noted that the number of features resulting from inverse properties is minority (812, that is only 20.4% of the features). This result was expected since it was already observed in other related problems of the literature describing the entities (infoboxes, facet generation).

Of all the comparison features, 560 (or 14.1%) are numerical ones. Without the inverse properties, there were only 3.3% of numerical features. This high increase is a good reason for using a specific display. Moreover, the numerical values are generally tens for non-inverse properties while they are of the order of several thousands for inverse properties, making it impossible to be displayed by list.

8. CONCLUSION

This paper presented a generation method of comparison tables from the Semantic Web. To this end, it introduced a measure that evaluates whether a feature has values for the compared entities which are sufficiently common among other similar entities. One has broken down the computation

Figure 7. Query error rate

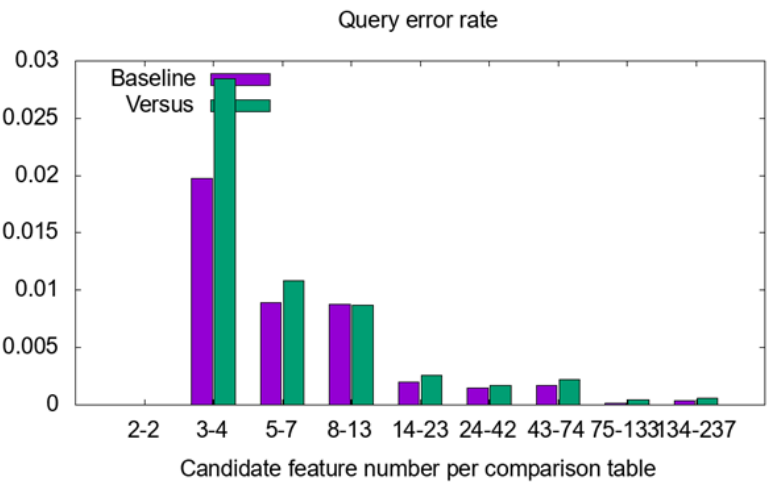
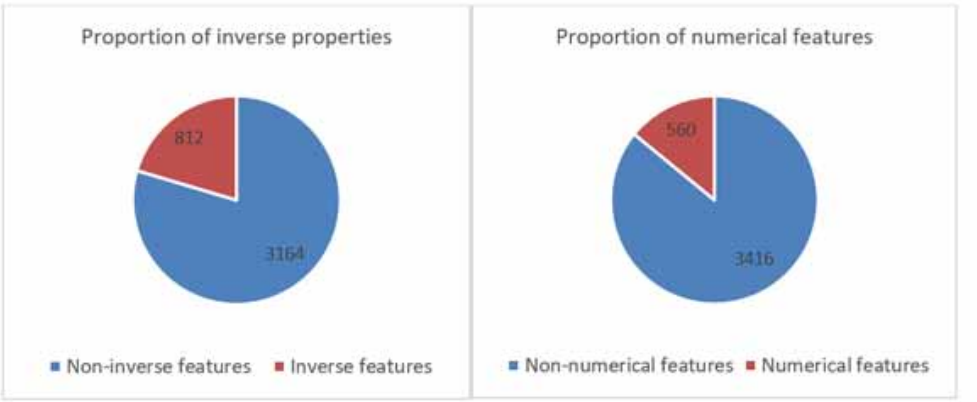


Figure 8. Proportion of numerical features



of the contextual reference level into several low-cost SPARQL queries so that it satisfies the fair-use policy of Wikidata public endpoint. Finally, this computation is also optimized in Versus to reduce this number of queries. Experiments on the built benchmark show the good precision of the contextual reference level for isolating the most relevant features. Interestingly, this entity-centric approach has a higher recall and accuracy than a baseline using facet-oriented metric, which relies on classes. Moreover, thanks to the presented optimization, Versus is faster than a naive approach. Interestingly, the experiments also investigated the impact of inverse properties on the construction of comparison tables. They lead to a non-negligible number of features, which lend themselves well to a display in the form of a count.

In future work, it would be interesting to investigate other kinds of interestingness measures not based on the contextual reference level, but on the contrary, on exceptionality. If such measures are likely to have a weak recall, they could be used in addition to the contextual reference level for extracting unexpected features. Instead of evaluating each feature one by one, it would also be relevant to extract an interesting *set* of features so as to avoid redundancies. This would be essential to combine several endpoints from the Linked Open Data cloud that necessarily contain repeated information.

ACKNOWLEDGMENT

We thank the evaluators for the time they took to annotate the features. This work was partially supported by the grant ANR-18-CE38-0009 (“SESAMES”).

REFERENCES

Anyanwu, K., Maduko, A., & Sheth, A. (2005). SemRank: Ranking complex relationship search results on the semantic web. *Proc. of the 14th international conference on World Wide Web*, 117–127. doi:10.1145/1060745.1060766

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). DBpedia: A nucleus for a web of open data. In *The Semantic Web* (pp. 722–735). Springer. doi:10.1007/978-3-540-76298-0_52

d'Amato, C., Fanizzi, N., & Esposito, F. (2009). A semantic similarity measure for expressive description logics. arXiv preprint arXiv:0911.5043

Dessi, A., & Atzori, M. (2016). A machine-learning approach to ranking RDF properties. *Future Generation Computer Systems*, *54*, 366–377. doi:10.1016/j.future.2015.04.018

Feddoul, L., Schindler, S., & Löffler, F. (2019). Automatic facet generation and selection over knowledge graphs. In *International Conference on Semantic Systems*. Springer. doi:10.1007/978-3-030-33220-4_23

Giacometti, A., Markhoff, B., & Soulet, A. (2021). Comparison Table Generation from Knowledge Bases. In *Extended Semantic Web Conference* (pp. 179–194). Springer. doi:10.1007/978-3-030-77385-4_11

Giacometti, A., Markhoff, B., & Soulet, A. (2021). Versus: générateur de tableaux comparatifs à partir de bases de connaissances. In Extraction et Gestion des connaissances. RNTI.

Hahn, R., Bizer, C., Sahnwaldt, C., Herta, C., Robinson, S., Bürgle, M., Düwiger, H., & Scheel, U. (2010). Faceted Wikipedia search. In *International Conference on Business Information Systems*. Springer.

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, *33*(1), 159–174. doi:10.2307/2529310 PMID:843571

Oren, E., Delbru, R., & Decker, S. (2006). Extending faceted navigation for RDF data. In *International Semantic Web conference*. Springer.

Petrova, A., Sherkhonov, E., Grau, B. C., & Horrocks, I. (2017). Entity comparison in RDF graphs. In *International Semantic Web Conference*. Springer.

Razniewski, S., Suchanek, F., & Nutt, W. (2016). But what do we actually know? *Proc. of the 5th Workshop on Automated Knowledge Base Construction*, 40–44. doi:10.18653/v1/W16-1308

Sáez, T., & Hogan, A. (2018). Automatically generating Wikipedia info-boxes from Wikidata. *Companion Proceedings of the Web Conference 2018*, 1823–1830. doi:10.1145/3184558.3191647

Soulet, A., & Suchanek, F. M. (2019). Anytime large-scale analytics of linked open data. In *International Semantic Web Conference*. Springer. doi:10.1007/978-3-030-30793-6_33

Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). Yago: A core of semantic knowledge. *Proc. of the 16th international conference on World Wide Web*, 697–706. doi:10.1145/1242572.1242667

Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*(4), 327–352. doi:10.1037/0033-295X.84.4.327

Tzitzikas, Y., Manolis, N., & Papadakos, P. (2017). Faceted exploration of rdf/s datasets: A survey. *Journal of Intelligent Information Systems*, *48*(2), 329–364. doi:10.1007/s10844-016-0413-8

Vrandečić, D., & Krötzsch, M. (2014). Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, *57*(10), 78–85. doi:10.1145/2629489

Wu, F., & Weld, D. S. (2008). Automatically refining the Wikipedia infobox ontology. *Proc. of the 17th international conference on World Wide Web*, 635–644. doi:10.1145/1367497.1367583

Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., & Auer, S. (2016). Quality assessment for linked data: A survey. *Semantic Web*, *7*(1), 63–93. doi:10.3233/SW-150175

ENDNOTES

1 https://socialcompare.com/
2 https://downloads.dbpedia.org/repo/lts/publication/strategy/2019.09.09/strategy_databus_initiative.pdf
3 https://en.wikipedia.org/wiki/Template:Infobox_person
4 The Typewriter font denotes the literals from Wikidata that are used as illustrations.
5 https://query.wikidata.org/