




# Class Schema Discovery from Semi-Structured Data

Everaldo Costa Neto   [ Universidade Federal de Pernambuco and Instituto Federal da Bahia - campus Euclides da Cunha | [ecsn@cin.ufpe.br](mailto:ecsn@cin.ufpe.br) ]

Johny Moreira  [ Universidade Federal de Pernambuco | [jms5@cin.ufpe.br](mailto:jms5@cin.ufpe.br) ]

Luciano Barbosa  [ Universidade Federal de Pernambuco | [luciano@cin.ufpe.br](mailto:luciano@cin.ufpe.br) ]

Ana Carolina Salgado  [ Universidade Federal de Pernambuco | [acs@cin.ufpe.br](mailto:acs@cin.ufpe.br) ]

 Centro de Informática, Universidade Federal de Pernambuco, Recife, PE, Brazil.

Received: 26 February 2023 • Published: 20 October 2023

**Abstract** A wide range of applications has used semi-structured data. A characteristic of this type of data is its flexible structure, i.e., it does not rely on schema-based constraints to define its entities. Usually entities of a same kind (i.e, class) do not present the same attribute set. However, some data processing and management applications rely on a data schema to perform their tasks. In this context, the lack of structure is a challenge for these applications to use this data. In this paper, we propose CoFFee, an approach to class schema discovery. Given a set of heterogeneous entity schemata, found within a class, CoFFee provides a summarized set with core attributes. To this end, CoFFee applies a strategy combining attributes co-occurrence and frequency. It models a set of entity schemata as a graph and uses centrality metrics to capture the co-occurrence between attributes. We evaluated CoFFee using data from 12 classes extracted from DBpedia and e-Commerce datasets. We benchmarked it against two other state-of-the-art approaches. The results show that: i) CoFFee effectively provides a summarized schema, minimizing non-relevant attributes without compromising the data retrieval rate; and ii) CoFFee produces a summarized schema of good quality, outperforming the baselines by an average of 19% of F1 score.

**Keywords:** Schema Discovery, Entity Classes, Semi-structured Data, Class Attributes

## 1 Introduction

Semi-structured data, such as RDF and JSON have been widely used by different applications, e.g., applications for structured queries [Adolphs *et al.*, 2011], data integration [Hassanzadeh *et al.*, 2013], and information extraction [Moreira and Barbosa, 2021]. The lack of schema is the major difficulty when trying to consume these data. In this context, dataset schema-related information leverages its use by these applications. For example, to the query formulation task, writing a query requires prior knowledge of the structure of a dataset. Thus, schema-related information describing classes, attributes, and resources contained in the dataset helps the execution of this task.

Despite being a W3C recommendation<sup>1</sup>, many datasets do not provide or have incomplete schema-related information. To this end, schema discovery approaches have been proposed in the literature in order to identify a data schema from a dataset [Christodoulou *et al.*, 2015; Kellou-Menouer and Kedad, 2015; Bouhamoum *et al.*, 2020]. Kellou-Menouer *et al.* [2021] published a survey identifying and classifying the main approaches to schema discovery according to the target problem.

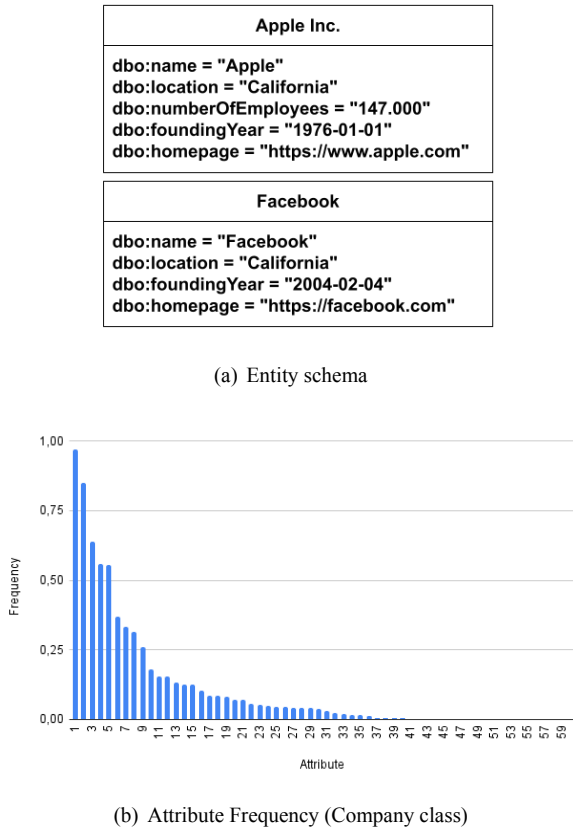
Previous approaches such Christodoulou *et al.* [2015] and Kellou-Menouer and Kedad [2015] have tried to infer a schema for a dataset by identifying the entity classes contained in it. After identifying the classes, it is necessary to define the classes schema. Usually, the set of attributes describing the instances of a class are the ones that will be compos-

ing the class schema. The approaches aforementioned consider the union of the attributes of all its instances. However, this naive method can present some inconsistencies. First, entities of the same class might not necessarily follow a pre-defined schema, and may have different attribute set. Second, the set of all attributes can be large, and the attributes are not equally relevant.

To illustrate this situation, consider the *company* class extracted from DBpedia. Figure 1a presents a snippet of the Apple Inc. and Facebook schemas. Both are companies, but Apple Inc. is described by the `numberOfEmployees` attribute while Facebook is not. This example shows the heterogeneity among the schemas in a same class. The union of the attributes of all its instances is equivalent to 60 attributes, which are not equally relevant. Figure 1b shows the frequency distribution of the attributes of the entities in the *company* class. Note that 37 attributes (61%) occur in less than 5% of instances, while only 5 attributes (8%) occur in more than 50% of instances. In other words, the union strategy may include attributes not relevant to describe the set of instances within a class.

Thus, to fill this gap it is necessary to find a way to define a concise representation, i.e., a summarized schema, for an entity class. A summarized schema is useful for applications that need a well-defined schema to perform their tasks. In this sense, our goal is to explore a set of heterogeneous entity schemas  $S$  to find a class schema  $S_C$ , which contains the most relevant attributes for class  $C$ . Other papers have proposed some related approaches [Wu and Weld, 2007; Moreira and Barbosa, 2021; Issa *et al.*, 2019; Wang *et al.*, 2015];

<sup>1</sup><https://www.w3.org/TR/dwbp/#StructuralMetadata>



**Figure 1.** Examples: (a) Snippet of the Apple Inc. and Facebook schemas (b) Frequency Distribution

however, the core of these solutions is based only on the frequency of the attributes.

**Proposal.** Our intuition is that less frequent attributes which co-occur with the frequent ones are also important to compose a class schema. Aligned to the most frequent attributes the less frequent ones can also introduce some relevance to the context and provide a more complete schema. Thus, we propose **CoFFee**, a free-parameters approach that balances *co-occurrence* and *frequency* of attributes. CoFFee models the entity schemas as a graph and uses centrality metrics (degree centrality and closeness) to capture the notion of co-occurrence between attributes. In addition, we propose a novel score that calculates the relevance of an attribute for a set of entity schemas, combining the centrality and frequency values. We use this score to rank and select a set of core attributes for the class.

**Evaluation.** We evaluate CoFFee on twelve distinct entity classes extracted from DBpedia and e-Commerce datasets. We carried out a comparative analysis with two state-of-the-art approaches most correlated with our proposal. The main results show that: (i) CoFFee provides a summarized schema for a class by filtering out non-relevant attributes; and (ii) our approach has a greater recall compared to baselines, achieving a balance between co-occurrence and frequency.

**Contributions.** We consider the main contributions of this work: (i) a class schema discovery approach that, given a set of entity schemas, provides a summarized schema containing the most significant attributes for a class; (ii) a novel score that calculates the relevance of an attribute combining co-occurrence and frequency; and (iii) a parameter-free heuristic

to select a set of core attributes based on their relevance.

This work extends Costa-Neto *et al.* [2022] by expanding the previous experiments. More specifically, we include six new classes in the experimental evaluation and augment the discussion of the results. Furthermore, we present a new experiment aiming to analyze the influence of frequency and co-occurrence metrics on the attribute relevance score and, consequently, on the results produced by CoFFee. Finally, we add a new section where we present an overview of the schema discovery problem and point out some examples of applications that can benefit from the presented solution.

The remainder of the paper is organized as follows. Section 2 formalizes the main concepts. Section 3 presents a brief discussion on the research problem, here addressed by the schema discovery solution. Section 4 discusses some related work and compares it with our paper. In Section 5 we define CoFFee, describing how each step works. Section 6 describes the experiments performed and the results achieved. Finally, in Section 7, we present the final considerations and guide the next steps of the work.

## 2 Definitions

In this section, we present the definitions of some concepts to help understand the problem we tackle in this paper.

**Definition 2.1 (Entity).** An entity  $e$  is a real-world object described by a set of attributes.

**Definition 2.2 (Class).** A class  $C$  is formed by a set of entities that describe the same concept. An entity is seen as an instance of a class. For example, *Apple Inc.* is an instance of the *company* class.

**Definition 2.3 (Entity schema).** An entity schema  $s(e) = \{a_1, \dots, a_n\}$  consists of a set of attributes that describe an entity  $e$ , e.g., for *Apple Inc.* their entity schema is  $s(\text{Apple\_Inc.}) = \{\text{homepage}, \text{location}, \dots, \text{foundingYear}, \text{numberOfEmployees}\}$ .

**Definition 2.4 (Class schema).** A class schema  $S_C = \{a_1, \dots, a_m\}$  consists of a set of core attributes most relevant for represent a set of instances of  $C$ . In this work, the relevance of an attribute is measured by a relevance score combining co-occurrence and frequency metrics.

Based on these definitions, we define our research problem as follows:

**Definition 2.5 (Problem definition).** Given a set of entity schemas  $S = \{s_1, \dots, s_n\}$ , such that each  $s_i \in S$  is an entity schema within the same class  $C$ , we aim to find  $S_C$ .

## 3 Background

In the last two decades the research in Schema Discovery has gained more visibility due to the large adoption of semi-structured data formats, such as XML (Extensible Markup Language), JSON (JavaScript Object Notation), and RDF (Resource Description Framework), as described by Gómez *et al.* [2018].

Semi-structured datasets have a flexible structure, i.e., they do not rely on schema-based constraints to define their entities [Poyraz, 2022]. If, on one hand, this flexibility facilitates the publication of data, on the other hand, the understanding and consumption of these datasets become difficult [Bouhamoum *et al.*, 2022].

Some data processing and management applications rely on a schema to perform their tasks. For example, applications for information extraction based on the slot-filling task depend on a schema to guide the extraction process [Moreira and Barbosa, 2021]. Other applications, such as Q&A systems [Adolphs *et al.*, 2011; Han *et al.*, 2011] and Data Integration [Dong and Srivastava, 2015] require a data schema to intermediate their end tasks, e.g., map a question written in natural language to a structured query and schema mapping, respectively.

Overall, the schema discovery task is based on exploring a dataset to obtain a high-level representation of its structure. Usually, this representation is given by classes and their attributes [Bouhamoum *et al.*, 2020]. In this direction, some approaches for schema discovery were proposed. Kellou-Menouer *et al.* [2021] organized these approaches according to their objectives. Among them, approaches to implicit schema discovery stand out.

According to Kellou-Menouer *et al.* [2021], these approaches aim to identify implicit entity classes in the dataset, in which each class represents a set of similar entities, i.e., entities of the same kind. In this context, a complementary step is the class-level schema discovery. Commonly, entities within the same class are described by a distinct set of attributes. Therefore, it is necessary to deal with the heterogeneity between different representations of similar entities to present a single, consistent, and less complex schema. In this direction, approaches to class schema discovery seek to provide a summarized schema for an entity class. In the next section, we discuss some related work that addresses this task.

## 4 Related Work

In this paper, we propose a class schema discovery approach. In other words, we want to summarize a set of diverse entity schemata found within a given class. Here we discuss papers that similarly deal with this problem.

Wu and Weld [2007] and Moreira and Barbosa [2021] address this problem for the Information Extraction context. Both define a class schema to guide the extraction process. To do this, they calculate the frequency that an attribute appears in the set of schema and select the attributes whose frequency is above a defined threshold. In their experiments, Moreira and Barbosa [2021] defined the schema of some classes, e.g., Country, Artist and University, considering the attributes that appear in at least 60% of the entities of each class.

Weise *et al.* [2016] proposed LD-VOWL, a tool for extracting and visualizing schema information for Linked Data. The authors use the class-centring perspective to extract schema information for a data source. In other words, SPARQL queries are submitted over the instances of a class to reveal

their schema. Specifically, a query identifies the  $k$  most frequent attributes, and the class schema is defined from this result.

Issa *et al.* [2019] proposed LOD-COM, a tool to reveal the conceptual schema of RDF datasets. The authors use an item mining-based approach to find frequent attribute patterns from a set of entities within the same class. The implementation of this approach considers the FP-growth algorithm. Thus, a parameter (support vector) is required to find frequent attribute patterns. As output, the tool returns a class diagram containing the classes and their relationships, the attributes, and an associated completeness value (i.e., percentage of entities that have that attribute). The authors performed a case study in 4 DBpedia classes (Film, Settlement, Organization, and Scientist) to illustrate how the tool works. They varied the parameter values between 0.9 and 0.1, showing that lower thresholds produce more complex schemas with the highest number of selected attributes.

Queiroz-Sousa *et al.* [2013] propose a method for summarizing ontologies. In this context, an ontology can represent a data source schema or describe a knowledge domain. This method considers centrality measure to find the most relevant concepts in a given ontology from user-defined parameters, e.g., summary size and threshold of relevance.

Wang *et al.* [2015] proposed a framework to manage JSON records. The framework supports some tasks, including Schema Consuming. The challenge of this task is to present a summarized schema for a set of heterogeneous JSON records of the same type (or class). To do this, the authors proposed Skeleton. The strategy is parameter-free and based on a gain and cost function. This function projects weights so that the class schema is inclined towards attributes occurring in equivalent schemas. In the experiment that evaluates the effectiveness of Skeleton, the authors used datasets extracted from some sources, such as DBpedia and Freebase, with entities of three different classes (types): Drug, Movie, and Company.

Kellou-Menouer and Kedad [2015] and Christodoulou *et al.* [2015] use a naive strategy to define the class schema. They consider the union of all attributes that occur in instances of class. Kellou-Menouer and Kedad [2015] name the class schema as type profile. A type profile is a description of a class, composed of the union of all attributes, as well as a probability value that indicates the frequency with which an attribute appears in the entities. They use the type profile to find overlapping between classes (generalization/specialization). In the experiments, the authors considered some synthetic and real RDF datasets. For instance, a DBpedia dataset with entities from the classes Politician, SoccerPlayer, Museum, Movie, Book, and Country. There are other naive approaches, e.g., common attribute set (intersect of attributes present in the schema set). As discussed earlier, these naive strategies are not useful in contexts where the set of schemata is heterogeneous.

The main weakness in Wu and Weld [2007]; Moreira and Barbosa [2021]; Weise *et al.* [2016]; Issa *et al.* [2019] is the choice of parameter. The frequency distribution varies by class and the set of instances. Thus, it is necessary to have prior knowledge of the distribution and organization of the data to define a suitable value for the parameter. In

the opposite direction, the approach proposed in this paper is parameter-free, being useful in case users have no prior knowledge of the data. Similar to our, Queiroz-Sousa *et al.* [2013] uses centrality measure, however its method depends on user-defined parameters.

An advantage of the approaches of Kellou-Menouer and Kedad [2015]; Christodoulou *et al.* [2015] is that they are parameter-free. However, the union of all attributes can generate an extensive class schema with non-relevant attributes, since they are not equally relevant. The approach proposed in Wang *et al.* [2015] considers the equivalence between the schemata to select the attributes. This strategy may fail to consider relevant attributes in scenarios with a less heterogeneous schema. In a different way, we propose an approach that combines co-occurrence and frequency. This combination contributes to increasing the recall of relevant attributes and minimizing attributes non-relevant to a set of schemata.

## 5 Solution: CoFFee

In this section, we detail **CoFFee**, an approach for class schema discovery that aims to find a set of core attributes to describe a class.

Returning to the example presented in Section 1, suppose we are interested in finding the schema of the class *company*. As seen earlier, the attributes of this class have a long-tail distribution, e.g., only 8% of attributes (5 of 60) have a frequency greater than 50%. Analyzing a less frequent attribute, e.g., `dbo:numberOfEmployees` (frequency = 37%), we verify that it has a high co-occurrence value with the most frequent attributes in the schema set, such as `dbo:name` and `dbo:foundingYear`. In this direction, the core of our approach is to combine these two aspects to find a high-quality summarized schema for a class. Figure 2 illustrates the pipeline executed to achieve our goal. Each step is detailed below.

### 5.1 Attribute graph creation

We model a set of entity schemas as a bipartite graph  $BG = \{E, A, EA\}$ , where  $E$  is a set of entities,  $A$  is a set of attribute, and  $EA$  is a set of edges between an entity and a attribute. Our goal is to capture the co-occurrence relationship between the attributes by generating an *attribute graph*, from  $BG$ .

**Definition 5.1** (Attribute graph). An attribute graph  $AG = \{A, ES\}$  is a graph where  $A$  is a set of attributes, and  $ES$  is a set of edges, in which there is an edge between two attributes  $a_k$  and  $a_j$  if they occur in the same entity schema.

We assume that attributes belonging to a set of entity schemas have been submitted to a schema alignment step, i.e., attributes that are homonyms and synonyms have been identified and aligned [Dong and Srivastava, 2015]<sup>2</sup>. Figure 3(a) illustrates an example of a bipartite graph created from a set of entity schemas. Blue rectangles represent an

entity, while green ellipses represent an attribute. The edges between an entity and attribute indicate that an entity  $e_i \in E$  is described by an attribute  $a_j \in A$ . Figure 3(b) illustrates an attribute graph resulting from the bipartite graph shown in Figure 3(a).

### 5.2 Metric calculation

From  $AG$ , we use two centrality metrics to capture the relationship between attributes: *degree* and *closeness centrality* [Zhang and Luo, 2017]. These metrics aim to identify the central nodes of the graph. Each metric expresses a dimension of centrality observed from the graph. The values for each metric are normalized and are in the range of 0 to 1. These centrality measures are defined below.

**Definition 5.2** (Degree centrality). It expresses the number of edges assigned to a node. The centrality degree of an attribute (node)  $a_k$  is calculated as follows:

$$DC(a_k) = \frac{m_i}{(N-1)} \quad (1)$$

Where  $m_i$  number of edges assigned to  $a_k$ , and  $N$  is the number of attributes in  $AG$ .

**Definition 5.3** (Closeness centrality). It denotes how close a node is to all nodes of the graph. This measure is the reciprocal of the sum of the distances from a node to the other nodes. The closeness centrality of an attribute  $a_k$  is calculated as follows:

$$Clo(a_k) = \left( \frac{\sum_{j=1}^N d(a_k, a_j)}{(N-1)} \right)^{-1} \quad (2)$$

Where  $d(a_k, a_j)$  is the shortest distance between  $a_k$  and  $a_j$  in  $AG$ .

We chose these metrics to capture the notion of co-occurrence, focusing on two main aspects: *linkage* and *influence*. For example, an attribute  $a_k$  with a high centrality degree indicates that there is a high number of attributes co-occurring with it. On the other hand, an attribute  $a_k$  with a high value of closeness indicates its high influence on other attributes, i.e., the attribute is close to attributes in the center of the graph. The idea is to capture with which attributes  $a_k$  co-occur. If it occurs with core attributes, its degree of closeness is greater.

We also calculate the frequency of an attribute  $a_k$  on a set of entity schemas  $S$ . The frequency is calculated as follows:

$$F(a_k) = \frac{n_k}{|S|} \quad (3)$$

Where  $n_k$  is the number of times  $a_k$  occurs in  $S$ .

### 5.3 Attribute relevance calculation

We propose a novel score to calculate the relevance of an attribute  $a_k$  concerning  $S$ . We use this score to define the

<sup>2</sup>In this paper, we run the experiments on DBpedia datasets that already solve this issue.

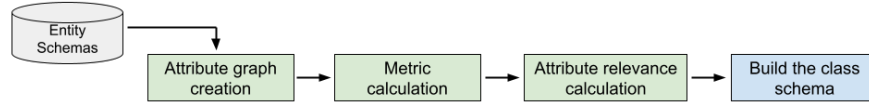


Figure 2. CoFFee's pipeline

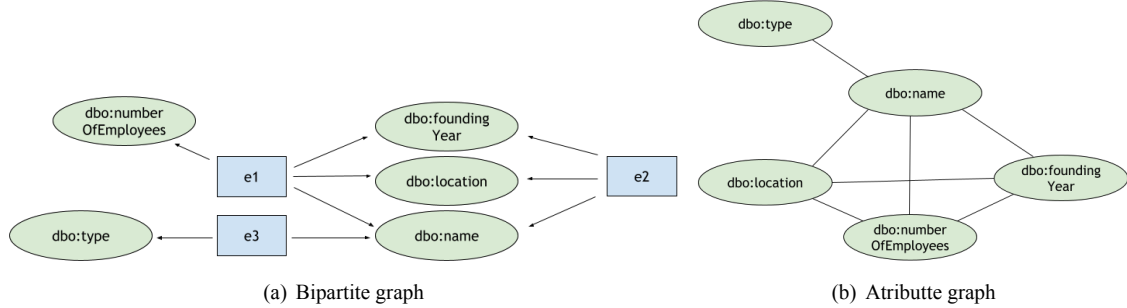


Figure 3. Example of graphs used by CoFFee. In (a) the bipartite graph created from the set of entity schemas, and (b) the attribute graph created from the relationships between the attributes of the set of entity schemas.

class schema. We combine the degree and closeness centrality metrics with the frequency. This score helps to capture less frequent attributes that keep relevant interconnections to core attributes. The attribute relevance is calculated as follows:

$$R(a_i) = DC(a_k) * w_{dc} + Clo(a_k) * w_{clo} + F(a_k) * w_f \quad (4)$$

The weights for each metric are defined proportionally. In our experiments we set:  $w_{dc} = 0.25$ ,  $w_{clo} = 0.25$ ,  $w_f = 0.5$ .

## 5.4 Build the class schema

In this step, our goal is to find  $S_C$  (see Definition 2.4).  $S_C$  is composed of the highest qualified attribute set to describe a set of entity schemas  $S$ . The quality of  $S_C$  is measured according to Equation 5. This measure considers the gain and cost of  $S_C$  (defined below) concerning  $S$ .

$$q(S_C) = \sum_{i=1}^N \alpha_i G(S_i, S_C) - \sum_{i=1}^N \beta_i C(S_i, S_C) \quad (5)$$

$$G(S_i, S_C) = \frac{|S_i \cap S_C|}{|S_i|} \quad (6)$$

$$C(S_i, S_C) = 1 - \frac{|S_i \cap S_C|}{|S_C|} \quad (7)$$

where,  $G(S_i, S_C)$  (Equation 6) is the gain of  $S_C$  in  $S_i$ , i.e., the percentage of attributes in  $S_i$  present in  $S_C$ , and  $C(S_i, S_C)$  (Equation 7) is the cost of  $S_C$  in  $S_i$ , i.e., the percentage of attributes of  $S_C$  that are not present in  $S_i$ . The weights  $\alpha_i$  e  $\beta_i$  indicate the importance of each  $S_i \in S$  in the gain and cost, respectively, such that  $\sum_{i=1}^N \alpha_i = \sum_{i=1}^N \beta_i = 1$ .

This quality metric was proposed by Wang *et al.* [2015]. However, we adapted the calculation of the weights. Thus,  $\alpha_i$  and  $\beta_i$  are calculated as follows:  $\alpha_i = \frac{r(S_i)}{\sum_{i=1}^N r(S_i)}$  and  $\beta_i = \frac{\frac{1}{r(S_i)}}{\sum_{i=1}^N \frac{1}{r(S_i)}}$ , where  $r(S_i) = \sum_{a_k \in S_i} R(a_k)$  is the sum of the attribute relevance values present in  $S_i$ . In short, the weights

allow the selection of the most relevant attributes to compose  $S_C$ . The assumption here is that the most relevant attributes are better at representing  $S$ .

Here, the main challenge is to find  $S_C$  that maximizes  $q(S_C)$ . Due to the size of  $A$ , it can be impractical to test all possible attributes combination. For example, considering the *company* class, where  $|A| = 60$ , there are  $2^{60}$  possible combinations. Thus, we propose a heuristic to find a set  $S_C$  that maximizes  $q(S_C)$  considering the attribute relevance.

Algorithm 1 details the process to find  $S_C$ . It receives as input a set of entity schemas  $S$  and a set of attributes ordered by their relevance  $R$  (Equation 4). It defines  $S_C$  as top- $j$  attributes in  $R$ , where  $j$  ranges from 1 to  $|R|$  (line 3). Thus, the quality for  $S_C$  is calculated using Equation 5 (line 5). The algorithm repeats this process until all attributes contained in  $R$  are added to  $S_C$ . For example, in the first iteration,  $S_C$  contains the most relevant attribute, while in the second iteration, it is equivalent to the two most relevant attributes, and so on. The assumption is that the quality value decreases as fewer relevant attributes are added to  $S_C$ . After executing lines 3-10, the algorithm checks which set of attributes maximized the quality and defines them as  $S_C$  to represent the class schema (line 11).

---

### Algorithm 1 Build the schema class

---

**Require:**  $S$ : Set of entity schemas;  $R$ : Set of attributes ordered by relevance (Eq. 4)

**Ensure:**  $S_C$ : Set of core attributes of the class

```

1:  $q_{max} \leftarrow 0$ 
2:  $k \leftarrow 0$ 
3: for  $j \leftarrow 1$  to  $|R|$  do
4:    $S_C \leftarrow$  pick top- $j$  in  $R$ 
5:    $q \leftarrow q(S_C)$ 
6:   if  $q \geq q_{max}$  then
7:      $q_{max} \leftarrow q$ 
8:      $k \leftarrow j$ 
9:   end if
10: end for
11:  $S_C \leftarrow$  pick top- $k$  in  $R$ 

```

▷ Eq. 5

---

## 6 Experiments

In this section, we present the experimental evaluation of our method and discuss the achieved results.

### 6.1 Dataset

We evaluated our approach over two DBpedia datasets (version 12/2021): *mappingbased-objects*<sup>3</sup> and *mappingbased-literals*<sup>4</sup>. We consider data from eleven classes: Film, Artist, Company, Scientist, University, Book, Actor, Aircraft, RacingDriver, Airport, and ShoppingMall. We choose these classes since evaluated baselines in this context also explore some of them. We identify the instances of each class through the *rdf:type* predicate contained in the *instance types* dataset<sup>5</sup>. We select these classes considering the diversity of the entities contained in each one of them, e.g., Places (Airport, ShoppingMall), Organisation (Company, University), Person (Artist, Actor, RacingDriver). Furthermore, DBpedia data is used by many applications.

We also use a public dataset with product specifications (monitor) extracted from the e-Bay product catalog<sup>6</sup>. A pre-processing step was required due to the noise coming from the web scraping. Hence, we considered the attributes with frequency above 5% and we align attributes with the same semantics, e.g., *manufacturer* and *brand*.

Table 1 presents statistics of the data. The **Entity Schemata** column indicates the number of entities (and schemas) belonging to each class. The **Attributes** column shows the number of distinct attributes contained in the entities' schemas. The **Distinct** column indicates the percentage of distinct schemas in the class, i.e., the degree of heterogeneity among entity schemata.

We consider different scenarios in terms of the number of entities, attributes, and level of heterogeneity. Specifically, we observed a correlation between these last two variables. Note that the classes with the highest level of heterogeneity are those with the highest number of distinct attributes, e.g., Company, Scientist, University, and Actor. This occurs because of the large number of attribute combinations in the entities' schemata. We also use classes with a medium (e.g., Artist and Book) and low (e.g. RacingDriver and Airport) heterogeneity level. The idea is to analyze how the evaluated approaches behave in these scenarios.

### 6.2 Baselines

We compare the performance of our approach against Skeleton [Wang et al., 2015] and LOD-CM [Issa et al., 2019] since these solutions are highly aligned with the objective of this paper. We briefly discuss the intuition behind each approach below.

<sup>3</sup>[https://databus.dbpedia.org/dbpedia/mappings/mappingbased-objects/2021.12.01/mappingbased-objects\\_lang=en.ttl.bz2](https://databus.dbpedia.org/dbpedia/mappings/mappingbased-objects/2021.12.01/mappingbased-objects_lang=en.ttl.bz2)

<sup>4</sup>[https://databus.dbpedia.org/dbpedia/mappings/mappingbased-literals/2021.12.01/mappingbased-literals\\_lang=en.ttl.bz2](https://databus.dbpedia.org/dbpedia/mappings/mappingbased-literals/2021.12.01/mappingbased-literals_lang=en.ttl.bz2)

<sup>5</sup>[https://databus.dbpedia.org/dbpedia/mappings/instance-types/2022.03.01/instance-types\\_lang=en\\_transitive.ttl.bz2](https://databus.dbpedia.org/dbpedia/mappings/instance-types/2022.03.01/instance-types_lang=en_transitive.ttl.bz2)

<sup>6</sup>[http://bit.ly/monitor\\_specs\\_di2kg\\_benchmark](http://bit.ly/monitor_specs_di2kg_benchmark)

Class	Entity Schemata	Attributes	Distinct (%)
Film	142,933	34	10
Artist	23,921	46	11
Company	65,400	60	37
Scientist	39,617	56	30
University	24,229	48	41
Book	46,388	34	18
Actor	3,516	37	30
Aircraft	12,301	20	5
RacingDriver	2,965	35	3
Airport	15,419	20	5
ShoppingMall	3,244	35	6
Monitor	4,191	38	31

**Table 1.** Dataset statistics

- **Skeleton.** It is a parameter-free approach that aims to present a summarized representation, i.e., a set of core attributes, for a set of schemas. It considers equivalence between schemas, and the class schema is inclined towards attributes that occur in equivalent schemas.
- **LOD-CM.** It uses the FP-growth algorithm to find patterns (i.e., a set of attributes) that frequently co-occur above a user-defined threshold. The class schema is the set of attributes contained in the set of patterns identified by the algorithm. We vary the parameter considering the values 0.5, 0.3, and 0.1.

### 6.3 Experimental setup

We performed three experiments, which are described below.

- **Experiment 1.** In this experiment, we aim to analyze the effectiveness of the class schema generated by the approaches, i.e., we check if the approaches provide a summarized schema without losing information that is relevant to the class. To this end, we use two metrics proposed in Wang et al. [2015]: Retrieval Rate (RR) and Relative Size (RS). In other words, RR measures the gain of information obtained using the class schema, whereas RS measures the size of the class schema concerning the universal attribute set. The metrics are calculated according to Equations 8 and 9, respectively.

$$RR = \frac{\sum_{i=1}^N \frac{|S_i \cap S_C|}{|S_i|}}{|S|} \quad (8)$$

$$RS = \frac{|S_C|}{|A|} \quad (9)$$

Where,  $S$  is a set of entity schemata,  $S_C$  is the class schema, and  $A$  is the set of distinct attributes in  $S$ .

- **Experiment 2.** In this experiment, we evaluate the influence that the weights have on the computation of attribute relevance (see Section 5.3) and, consequently, on the class schema generated by CoFFee. To this end, we define five scenarios varying the weights assigned to the metrics. Table 2 presents the setup in each of them.

Scenario	Frequency	Centrality	Closeness
1	0.90	0.05	0.05
2	0.70	0.15	0.15
3	0.50	0.25	0.25
4	0.30	0.35	0.35
5	0.10	0.45	0.45

**Table 2.** Distribution of weights for calculating attribute. relevance

- **Experiment 3.** In this experiment, we analyze the quality of the class schema in comparison to a reference schema.

For DBpedia classes, we consider the set of attributes belonging to the infobox template most used by its instances as reference schema. Infoboxes are one of the resources used by DBpedia to extract structured information from Wikipedia [Moreira *et al.*, 2021], and infobox templates are created by a crowdsourcing effort and are a reasonable approximation of the class schema. DBpedia provides an ontology, but it is not interesting to use it for this comparison due to its size. For example, the Scientist class has 239 attributes and aggregates attributes from its superclasses (Person, Agent, and Thing). However, Scientists instances do not use most of these attributes, e.g., the `olympicGamesWins` attribute, which belongs to the Person class. For these reasons, we believe that the infobox template provides a closer reference schema for the instances of a DBpedia class.

For the Monitor class, we consider the set of attributes the users can use for search refinement from e-Bay website. We believe that these attributes provide a summary representation of this class of products.

Table 3 presents information about the reference schema used in this experiment. The **Template** column indicates the used template’s name, and the **Attribute** column shows the number of attributes contained in the template. It is important to note that we excluded some attributes defined as metadata, such as: `image`, `alt`, and `caption`. We use Precision (P), Recall (R), and F-measure (F1) metrics to calculate schema quality. These metrics are calculated according to Equations 10, 11, and 12, respectively.

$$P = \frac{TP}{TP + FP} \quad (10)$$

$$R = \frac{TP}{TP + FN} \quad (11)$$

$$F1 = \frac{2 * P * R}{P + R} \quad (12)$$

Where  $TP$  (True Positive) is the number of selected attributes that belong to the reference schema;  $FP$  (False Positive) is the number of attributes that were selected but that do not belong to the reference schema; and  $FN$  (False Negative) is the number of attributes that belong to the reference schema but have not been selected.

## 6.4 Results

In this section, we discuss the experiments results.

Class	Template	Attributes
Film	Infobox_film	21
Artist	Infobox_artist	29
Company	Infobox_company*	19
Scientist	Infobox_scientist	40
University	Infobox_university	51
Book	Infobox_book	28
Actor	Infobox_person	12
Aircraft	Infobox_aircraft	39
RacingDriver	Infobox_F1_driver	23
Airport	Infobox_airport	27
ShoppingMall	Infobox_shopping_mall	17
Monitor	eBayMonitor	16

**Table 3.** Schema reference information (\*short version)

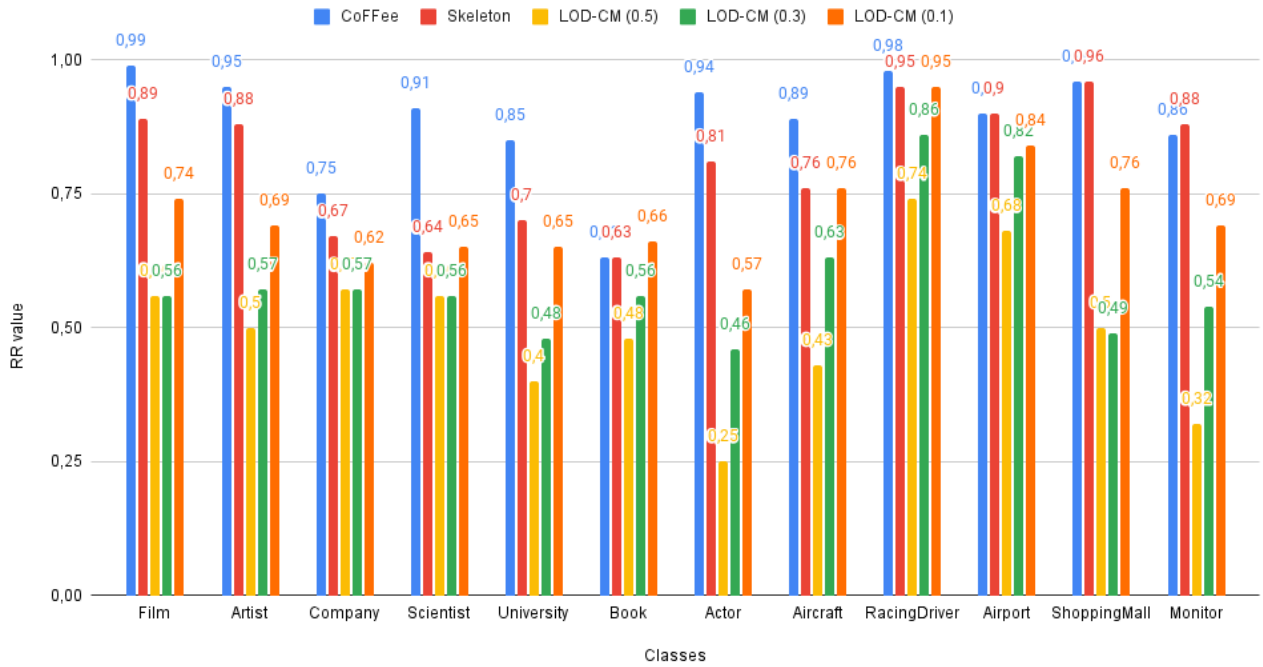
Class	Universal attribute set	
	RR = 1	RS = 1
Film	0.01	0.50
Artist	0.05	0.74
Company	0.25	0.85
Scientist	0.09	0.77
University	0.15	0.78
Book	0.37	0.83
Actor	0.06	0.66
Aircraft	0.11	0.70
RacingDriver	0.02	0.80
Airport	0.10	0.55
ShoppingMall	0.04	0.83
Monitor	0.14	0.61
<b>AVG</b>	<b>0.12</b>	<b>0.72</b>

**Table 4.** Difference between Universal x CoFFee approaches for RR and RS metrics.

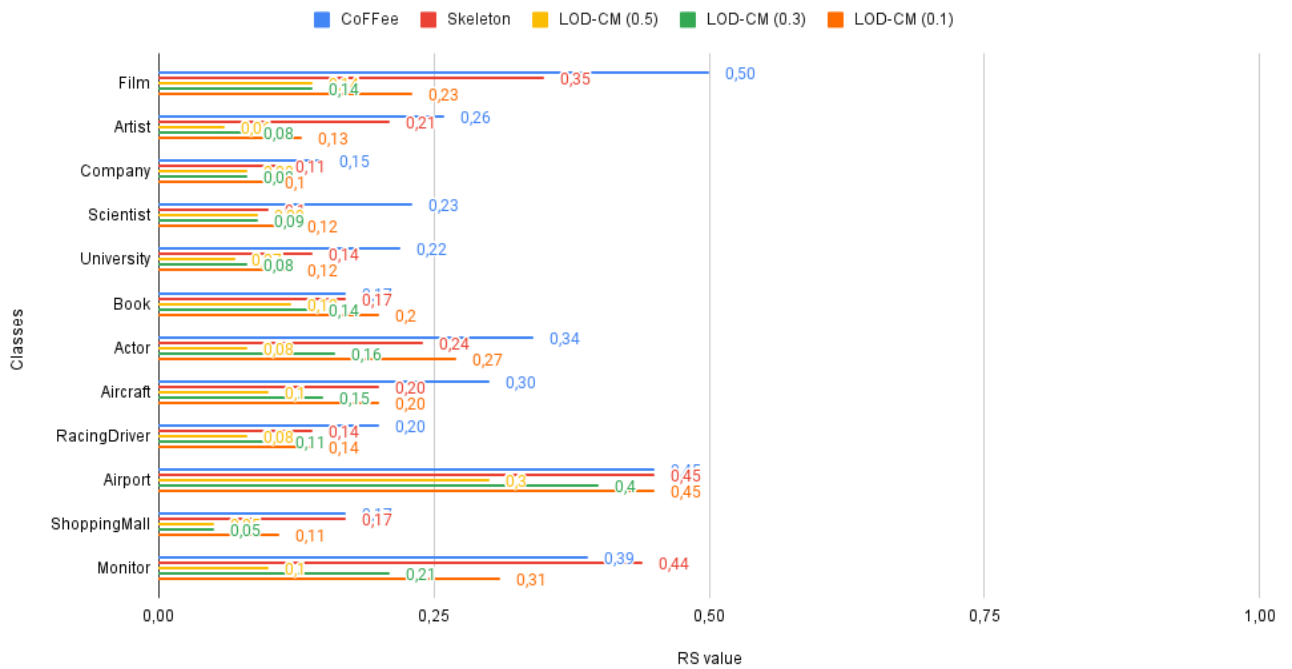
### 6.4.1 Experiment 1

Figure 4 shows the performance of the approaches concerning the retrieval rate (RR) and relative size (RS) metrics. For comparison, we consider the universal attribute set (i.e., the union of all attributes of all instances of a class) as a baseline. The value of RR and RS for this universal schema are equal to 1. Our goal is to provide a summarized class schema without losing relevant information. For that, we minimize the RS index while keeping the RR value as close as possible to 1.

When comparing CoFFee with the universal attribute set, the RR index varies between 0.63 (Book) and 0.99 (Film). Also, the index stays above 0.80 in 10 out of 12 evaluated classes. Meanwhile, the RS index falls between 0.5 (Film) and 0.15 (Company) - see Figure 4. Table 4 presents the difference obtained between CoFFee’s performance and the baseline (Universal). Note that the difference in performance measured by the RR index is smaller than the RS index. In summary, the number of attributes selected by CoFFee is 72% lower than the baseline on average, see the RS index. Nonetheless, the RR index is significant. CoFFee averaged 88% for RR, achieving a difference of just 0.12 (see RR index) against the baseline. In other words, the set of attributes selected by CoFFee offers a more summarized description of the class instances while preserving the recall.



(a) RR index



(b) RS index

Figure 4. Effectiveness of approaches to summarize the class schema.



Looking at the metrics for Skeleton, we note that this approach also provides a summarized schema keeping the RR index relatively high. Specifically, comparing CoFFee x Skeleton, we noticed an increase in the RR index (gain). In other words, our approach selects more relevant attributes. Consequently, by increasing the number of selected attributes, the RS index grows (cost). However, the difference between RR and RS is positive, indicating that CoFFee selects attributes that contribute to leveraging the retrieval rate.

To understand and explore the particularity of these approaches, we take the *company* class as an example. Figure 5 shows the Company class attributes selected by CoFFee and Skeleton. Comparing, the former considers attributes `dbo:numberOfEmployees` and `dbo:keyPerson`, while the latter does not. Although the `dbo:numberOfEmployees` attribute has a similar frequency (0.32) to the `dbo:product` attribute (0.33), Skeleton does not select the attribute because it was not frequent in equivalent schemata. Despite the `dbo:numberOfEmployees` attribute does not appear often in equivalent schemata, it does co-occur with core attributes such as `dbo:name`, `dbo:foundingYear` and `dbo:industry` in some schemata. Skeleton was built to select attributes that occur in equivalent schemas, unlike our approach that considers co-occurrence and frequency of attributes.

Attribute	CoFFee	Skeleton
<code>dbo:name</code>	✓	✓
<code>dbo:foundingYear</code>	✓	✓
<code>dbo:industry</code>	✓	✓
<code>dbo:type</code>	✓	✓
<code>dbo:homepage</code>	✓	✓
<code>dbo:location</code>	✓	✓
<code>dbo:product</code>	✓	✓
<code>dbo:numberOfEmployees</code>	✓	
<code>dbo:keyPerson</code>	✓	

Figure 5. Attributes selected by CoFFee and Skeleton (Class: Company)

We noticed that Skeleton usually gets an RR index high for classes in which the schemata are less heterogeneous (e.g., *RacingDriver* and *Airport*), i.e., a lower percentage of distinct schemata (see Table 1), while the RR index is lower in classes with heterogeneous schemata (e.g., *Company* and *Scientist*). In other words, in a more homogeneous scenario, Skeleton can select a greater number of attributes since its heuristic is based on schema equivalence. It is important to highlight that CoFFee presents a stable behavior in both scenarios since the core of its approach depends on the relationship of attributes and not on the equivalence between the entities' schemata.

In the monitor class, CoFFee selects fewer attributes than Skeleton (15 and 17 attributes, respectively). We observe that CoFFee ignored some less frequent attributes, which were considered by Skeleton because they appear in equivalent schemata, e.g., `miscellaneous` (this attribute appears in only 9% of monitor specifications). Despite that, the RR index of these approaches is close (0.86 x 0.88). Looking

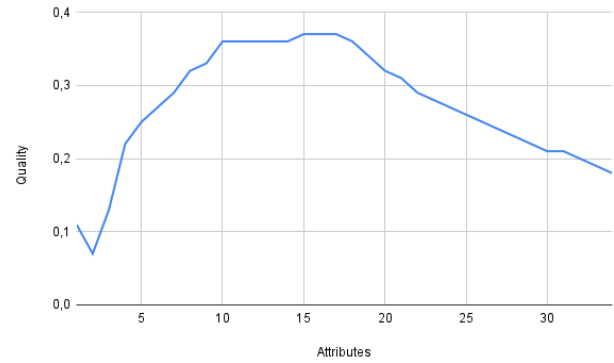


Figure 6. Quality of the class schema ordered by the relevance of the attributes. (Class: Film)

at the RS index (0.39 x 0.44), we observe that the trade-off between these two metrics is not positive. This shows that CoFFee balances gain and cost, identifying attributes that can increase the retrieval rate.

LOD-CM is the approach that provides a more summarized schema, i.e., it has a low RS index, but the RR index value is also low (see Figure 4). In other words, this approach fails to consider relevant attributes. LOD-CM depends on a parameter to find a set of attributes that co-occur under this threshold. In our experimental evaluation, we define three values (0.5, 0.3, and 0.1). In this case, a low threshold implies an increased coverage of attributes. The selection of an attribute is conditioned on the existence of a pattern that satisfies the defined threshold. In this sense, this approach can limit the number of selected attributes depending on the value assigned to the parameter and the frequency distribution of the attributes, especially in scenarios where this distribution is long-tail. Manually setting it is challenging when the user has no prior knowledge of the dataset. It is important to note that the CoFFee and Skeleton approaches are parameter-free.

CoFFee applies a heuristic to build the class schema based on the quality metric that balances gain and cost, weighted by the attribute relevance score (Equation 5). In this sense, we evaluate the behavior of this heuristic. Figure 6 shows how the quality varies as attributes are added to the class schema. CoFFee considers the 17 most relevant attributes to compose the schema of the class *Film*. Figure 6 shows the schema's quality decreasing as we add less relevant attributes to it. Comparing CoFFee to the Universal schema, we observe that the class schema size is reduced by 50%, while the RR index remains close to 1. In summary, CoFFee showed to be efficient to provide a concise formation in comparison with the universal attribute set, minimizing non-relevant attributes without compromising the recall of the information retrieved by the class.

Since CoFFee can be used with data processing and retrieval solutions, we measure the execution time (seconds) it spends to discover the schema of each class. The results are shown in Figure 7, in which each bar is the average of 5 executions by class.

There is a correlation between the execution time and the number of entities in each class. Since our heuristic to build the class schema (Algorithm 1) uses an iterative process, considering the number of attributes, we compute the Spearman

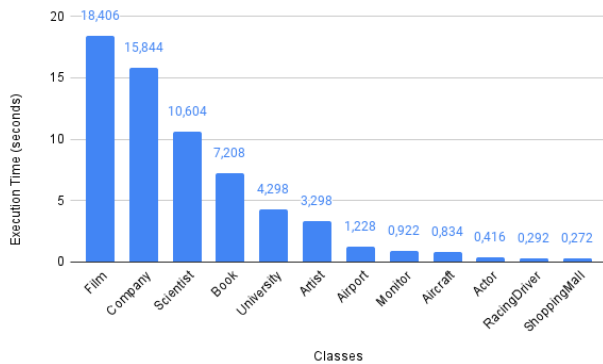


Figure 7. CoFFee's execution time

correlation [Spearman, 1961] between the execution time and the number of distinct attributes in each class. The test result accepted the null hypothesis, showing that these variables are unrelated. In other words, the observations show that the time spent by CoFFee to discover the schema of each class is influenced by the number of entities and not by the number of distinct attributes contained in the entities of a class. For example, see the results for the Film and Book classes.

#### 6.4.2 Experiment 2

We observe CoFFee's performance in the schema summarization task considering the RR and RS values obtained with the class schema generated in the scenarios of Section 6.3. For comparison purposes, we adopt scenario 3 (the default configuration used by CoFFee) as a baseline. In this experiment, we only consider classes where the RR value is less than 0.95 in scenario 3.

Figure 8 presents the values obtained in each scenario considering the RR (a) and RS (b) metrics. In general, we observe that changing the weights of the metrics can influence the calculation of the attribute's relevance score and, consequently, the output produced by CoFFee. We notice that these changes are more sensitive in scenarios 1 and 5, i.e., when the defined weights are far from the default values. To understand this behavior, we focus our discussion on four classes where these differences were significant: Company, Book, Aircraft, and Airport.

We notice a trend in part of the observed classes: the RR value increases as the frequency weight decreases, e.g., Book and Aircraft. This increases the number of attributes selected and, consequently, the retrieval rate. However, it is essential to note that this trend is not a pattern. In this case, the frequency distribution exerts an influencing factor. For example, for the Aircraft class in scenario 3, CoFFee selected six attributes, while in scenario 5, it selected ten attributes. When comparing the relevance score of the attributes in each scenario, we notice that the 4th most relevant attribute changed (`dbo:origin` in scenario 3 and `dbo:unitCost` in scenario 5). Although the `dbo:origin` attribute has a higher frequency than the `dbo:unitCost` attribute, the latter co-occur more frequently with the most frequent attributes of the class. In scenario 5, the weights for the co-occurrence metrics are higher. For this reason, the `dbo:unitCost` at-

tribute was selected.

By increasing the number of selected attributes, the size of the class schema grows and reflects on the value of the RS metric, as can be seen in Figure 8b. As the challenge is to find the trade-off between RR and RS, we compare the difference between these values against scenario 3 (default setup). We find that there is no significant gain in most cases. For example, in the *Airport* class, in scenario 3, it is possible to obtain a retrieval rate of 0.89 with a cost of 30% of the attributes, while in scenario 5, the retrieval rate is 0.99 using 50% of the attributes. In other words, the gain in retrieval rate is proportionally less than the cost.

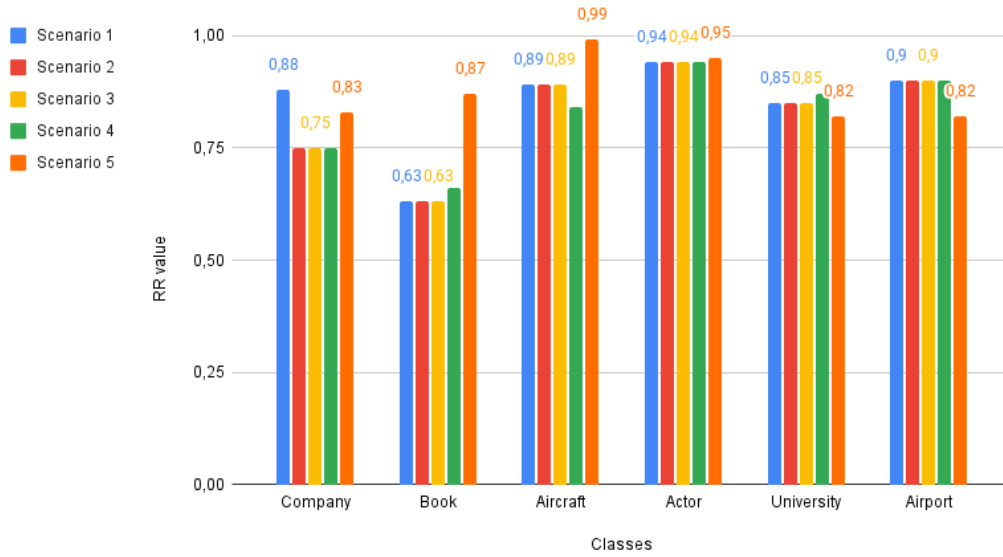
Scenarios 1 and 3 present an opposite situation to the one discussed previously. However, we noticed that this did not happen in the evaluated classes. Except for the *company* class, there was no difference in the retrieval rate. Specifically, in scenario 1 of the class *company*, CoFFee selected 16 attributes, while in scenario 3 it selected 9. We noticed that this non-standard behavior is directly related to the frequency distribution of the attributes. In this case, a small portion of attributes have a frequency very close to each other and therefore was considered part of the class schema since, in this scenario, the frequency metric has a significant weight concerning the co-occurrence metrics. On the other hand, in scenario 3, this same attribute portion is discarded due to its low co-occurrence with the other attributes.

In summary, we can conclude from the analysis carried out in this experiment that the default setup of CoFFee has proved adequate for the classes we evaluate. Most of the time, there is no prior knowledge of the dataset during the schema discovery process. In this case, relying on the default setup, CoFFee is viable since the difference between gain (RR) and cost (RS) remains stable. Based on this, we can state that the balance between frequency and co-occurrence weights allows for finding the trade-off between RR and RS. However, it is important to emphasize that this experiment provides an intuition for potential users of our approach. Applications may have different goals, e.g., some may want a schema with higher coverage of attributes even if this results in a complex schema (with a higher number of attributes), while others with a balanced schema size, prioritizing those attributes that are more descriptive for class. In this sense, the weights for calculating the relevance of the attribute can be calibrated so that CoFFee can cover both cases.

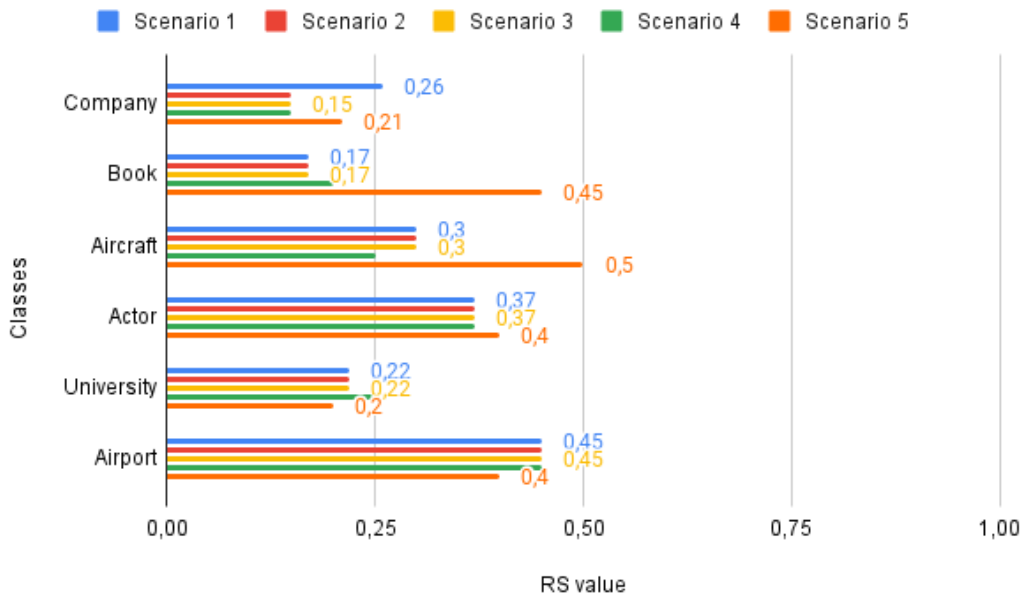
#### 6.4.3 Experiment 3

Table 5 benchmarks the proposed approach with the baselines regarding the reference schema. All approaches had a high precision (close to 1) in most of the evaluated classes, i.e., the attributes selected were present in the reference schema, with few exceptions. For example, 16 of 17 attributes selected by CoFFee in the Film class were present in the reference schema. The exception was the `dbo:imdbId` attribute. This attribute belongs to the class but is not being considered for new instances<sup>7</sup>. For this reason, the attribute is not present in the reference schema. A similar case also occurs in the Artist, Actor and Airport classes.

<sup>7</sup>According to infobox template: [https://en.wikipedia.org/wiki/Template:Infobox\\_film](https://en.wikipedia.org/wiki/Template:Infobox_film)



(a) RR index



(b) RS index

Figure 8. CoFFee’s performance in the schema summarization task in several scenarios.

Class	Approach	P	R	F1
Film	CoFFee	0.94	0.76	<b>0.84</b>
	Skeleton	1.00	0.57	0.72
	LOD-CM (0.5)	1.00	0.24	0.38
	LOD-CM (0.3)	1.00	0.24	0.38
	LOD-CM (0.1)	1.00	0.38	0.55
Artist	CoFFee	0.92	0.38	<b>0.54</b>
	Skeleton	0.90	0.31	0.46
	LOD-CM (0.5)	1.00	0.10	0.19
	LOD-CM (0.3)	1.00	0.13	0.24
	LOD-CM (0.1)	1.00	0.20	0.34
Company	CoFFee	1.00	0.48	<b>0.64</b>
	Skeleton	1.00	0.37	0.54
	LOD-CM (0.5)	1.00	0.16	0.27
	LOD-CM (0.3)	1.00	0.16	0.27
	LOD-CM (0.1)	1.00	0.31	0.48
Scientist	CoFFee	1.00	0.33	<b>0.49</b>
	Skeleton	1.00	0.15	0.26
	LOD-CM (0.5)	1.00	0.12	0.22
	LOD-CM (0.3)	1.00	0.12	0.22
	LOD-CM (0.1)	1.00	0.17	0.29
University	CoFFee	1.00	0.22	<b>0.36</b>
	Skeleton	1.00	0.13	0.24
	LOD-CM (0.5)	1.00	0.06	0.11
	LOD-CM (0.3)	1.00	0.07	0.14
	LOD-CM (0.1)	1.00	0.11	0.21
Book	CoFFee	1.00	0.21	0.35
	Skeleton	1.00	0.21	0.35
	LOD-CM (0.5)	1.00	0.14	0.25
	LOD-CM (0.3)	1.00	0.17	0.30
	LOD-CM (0.1)	1.00	0.25	<b>0.40</b>
Actor	CoFFee	0.85	1.00	0.92
	Skeleton	0.77	0.58	0.66
	LOD-CM (0.5)	1.00	0.25	0.40
	LOD-CM (0.3)	1.00	0.50	0.66
	LOD-CM (0.1)	1.00	0.83	<b>0.90</b>
Aircraft	CoFFee	1.00	0.16	<b>0.27</b>
	Skeleton	1.00	0.10	0.19
	LOD-CM (0.5)	1.00	0.05	0.10
	LOD-CM (0.3)	1.00	0.07	0.14
	LOD-CM (0.1)	1.00	0.10	0.19
RacingDrive	CoFFee	1.00	0.30	<b>0.46</b>
	Skeleton	1.00	0.21	0.35
	LOD-CM (0.5)	1.00	0.13	0.23
	LOD-CM (0.3)	1.00	0.17	0.29
	LOD-CM (0.1)	1.00	0.21	0.35
Airport	CoFFee	0.88	0.29	<b>0.44</b>
	Skeleton	0.88	0.29	<b>0.44</b>
	LOD-CM (0.5)	0.83	0.18	0.30
	LOD-CM (0.3)	0.87	0.25	0.39
	LOD-CM (0.1)	0.77	0.26	0.38
ShoppingMall	CoFFee	1.00	0.35	<b>0.52</b>
	Skeleton	1.00	0.35	<b>0.52</b>
	LOD-CM (0.5)	1.00	0.11	0.21
	LOD-CM (0.3)	1.00	0.11	0.21
	LOD-CM (0.1)	1.00	0.23	0.38
Monitor	CoFFee	0.73	0.68	<b>0.71</b>
	Skeleton	0.58	0.62	0.60
	LOD-CM (0.5)	0.75	0.19	0.30
	LOD-CM (0.3)	0.75	0.38	0.50
	LOD-CM (0.1)	0.66	0.50	0.57

Table 5. Class schema quality compared to the reference schema.

We observe that the reference schema has more attributes (see Table 3) concerning the output produced by the approaches. However, most of the classes' instances do not use some attributes in the reference schema. Entity schemata are flexible, which means that instances of the same class have a set of distinct attributes and do not necessarily have all the attributes suggested for that class. Consequently, attributes less used by the class instances may exist in the reference schema. In Moreira *et al.* [2021], the authors observed this correlation between the size of attributes suggested in a template *versus* and the number of attributes used by the instances. In this sense, this justifies the values obtained by all approaches for the recall metric.

Regarding the F1 metric, CoFFee outperforms the baselines in most of the evaluated classes. The reason for this is that CoFFee achieves a high recall value. CoFFee obtained an average difference in recall of 0.11 points for Skeleton and 0.29, 0.23 and 0.13 for LOD-CM with parameters 0.5, 0.3 and 0.1, respectively. Unlike the other approaches, we leverage low frequent attributes considering their occurrence with core attributes (more frequent). The biggest difference in these results comes from the Scientist and Actor classes. For example, for Scientist class, CoFFee selects 13 attributes, while Skeleton and LOD-CM (0.5) select 6 and 5, respectively. We consider attributes like: `dbo:knowFor` and `dbo:award`, which are relevant attributes for a Scientist.

Overall, it was possible to verify that CoFFee provides a good quality schema to represent an entity class. The schema generated by CoFFee is in line with the reference schema (high precision). Compared with the other approaches for summarizing schemas, our approach covered the highest number of selected relevant attributes (highest recall and F1).

## 7 Conclusion

In this paper, we address the class-level schema discovery problem. We propose CoFFee, an approach capable of providing a summarized schema to represent the entities of a class. CoFFee deals with heterogeneous schemas and is effective in selecting the most relevant attributes by combining co-occurrence and frequency. We performed experiments with data from twelve classes extracted from DBpedia and e-Commerce datasets and compared CoFFee with two state-of-the-art approaches. Compared to these solutions, our approach increases the recall of attributes and keeps the precision at high rates when looking at a reference schema. The results obtained show that CoFFee is effective to provide a summarized schema without losing relevant information. Furthermore, we show that the default weights defined for the frequency and co-occurrence metrics are adequate to keep the results stable. In future directions, we intend to create a tool that provides schema-related information from the results obtained by CoFFee to describe the content and leverage the use of datasets that do not have this information. Moreover, we intend to adapt CoFFee to support an incremental approach. Datasets can be dynamic and may change over time by adding new entities, as described in Bouhamoum *et al.* [2022].

## Acknowledgements

Not applicable.

## Funding

Not applicable.

## Authors' Contributions

**Everaldo Costa Neto:** Investigation, Conceptualization, Methodology, Experiments and Writing. **Johny Moreira:** Writing and Revision. **Luciano Barbosa** and **Ana Carolina Salgado:** Validation, Writing and Revision.

## Competing interests

The authors have no competing interests to declare.

## Availability of data and materials

The datasets generated and/or analysed during the current study are available on github<sup>8</sup>.

## References

- Adolphs, P., Theobald, M., Schafer, U., Uszkoreit, H., and Weikum, G. (2011). Yago-qa: Answering questions by structured knowledge queries. In *2011 IEEE Fifth International Conference on Semantic Computing*, pages 158–161. IEEE.
- Bouhamoum, R., Kedad, Z., and Lopes, S. (2020). Scalable schema discovery for rdf data. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems XLVI*, pages 91–120. Springer.
- Bouhamoum, R., Kedad, Z., and Lopes, S. (2022). Incremental schema generation for large and evolving rdf sources. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems LI*, pages 28–63. Springer.
- Christodoulou, K., Paton, N. W., and Fernandes, A. A. (2015). Structure inference for linked data sources using clustering. In *Transactions on Large-Scale Data-and Knowledge-Centered Systems XIX*, pages 1–25. Springer.
- Costa-Neto, E., Moreira, J., Barbosa, L., and Salgado, A. C. (2022). Coffee: A co-occurrence and frequency-based approach to schema mining. In *Anais do XXXVII Simpósio Brasileiro de Bancos de Dados*, pages 52–64, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/sbbd.2022.224190.
- Dong, X. L. and Srivastava, D. (2015). *Schema Alignment*, pages 31–61. Springer International Publishing, Cham. DOI: 10.1007/978-3-031-01853-4<sub>2</sub>.
- Gómez, S. N., Etcheverry, L., Marotta, A., and Consens, M. P. (2018). Findings from two decades of research on schema discovery using a systematic literature review. In *AMW*.
- Han, L., Finin, T., and Joshi, A. (2011). Gorelations: An intuitive query system for dbpedia. In *Joint International Semantic Technology Conference*, pages 334–341. Springer.
- Hassanzadeh, O., Pu, K. Q., Yeganeh, S. H., Miller, R. J., Popa, L., Hernández, M. A., and Ho, H. (2013). Discovering linkage points over web data. *Proceedings of the VLDB Endowment*, 6(6):445–456.
- Issa, S., Paris, P.-H., Hamdi, F., and Si-Said Cherfi, S. (2019). Revealing the conceptual schemas of rdf datasets. In Giorgini, P. and Weber, B., editors, *Advanced Information Systems Engineering*, pages 312–327, Cham. Springer International Publishing.
- Kellou-Menouer, K., Kardoulakis, N., Troullinou, G., Kedad, Z., Plexousakis, D., and Kondylakis, H. (2021). A survey on semantic schema discovery. *The VLDB Journal*, pages 1–36.
- Kellou-Menouer, K. and Kedad, Z. (2015). Schema discovery in rdf data sources. In *International Conference on Conceptual Modeling*, pages 481–495. Springer.
- Moreira, J. and Barbosa, L. (2021). Deepex: A robust weak supervision system for knowledge base augmentation. *J. Data Semant.*, 10(3-4):309–325. DOI: 10.1007/s13740-021-00134-x.
- Moreira, J., Neto, E. C., and Barbosa, L. (2021). Analysis of structured data on wikipedia. *International Journal of Meta-data, Semantics and Ontologies*, 15(1):71–86.
- Poyraz, K. (2022). Partial rdf schema retrieval. Master's thesis.
- Queiroz-Sousa, P. O., Salgado, A. C., and Pires, C. E. (2013). A method for building personalized ontology summaries. *Journal of Information and Data Management*, 4(3):236–236.
- Spearman, C. (1961). The proof and measurement of association between two things.
- Wang, L., Zhang, S., Shi, J., Jiao, L., Hassanzadeh, O., Zou, J., and Wangz, C. (2015). Schema management for document stores. *Proc. VLDB Endow.*, 8(9):922–933. DOI: 10.14778/2777598.2777601.
- Weise, M., Lohmann, S., and Haag, F. (2016). Ld-vowl: Extracting and visualizing schema information for linked data. In *2nd international workshop on visualization and interaction for ontologies and linked data*, pages 120–127.
- Wu, F. and Weld, D. S. (2007). Autonomously semantifying wikipedia. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 41–50.
- Zhang, J. and Luo, Y. (2017). Degree centrality, betweenness centrality, and closeness centrality in social network. In *Proceedings of the 2017 2nd International Conference on Modelling, Simulation and Applied Mathematics (MSAM2017)*, volume 132, pages 300–303.

<sup>8</sup><https://github.com/ecsneto/coffee>