CoFFee: A Co-occurrence and Frequency-Based Approach to Schema Mining

Everaldo Costa Neto [1,2], **Johny Moreira** [1], **Luciano Barbosa** [1], **Ana Carolina Salgado** [1]

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE)

[2]Instituto Federal da Bahia (IFBA) – *Campus* Euclides da Cunha

{ecsn, jms5, luciano, acs}@cin.ufpe.br

Abstract. *A wide range of applications use semi-structured data. A characteristic of these data is that they are heterogeneous and do not follow a predefined schema, i.e., schema-less. The lack of structure makes it difficult to use this data since many applications depend on it to perform their tasks. Thus, we propose CoFFee, a schema mining approach that, given a set of heterogeneous schemas, provides a summarized schema containing a set of core attributes. To this end, CoFFee uses a strategy that combines co-occurrence and frequency of attributes. It models a set of entity schemas as a graph and uses centrality metrics to capture the co-occurrence between attributes. We evaluated CoFFee using data extracted from six DBpedia classes and compared it with two state-of-the-art approaches. The results achieved show that CoFFee produces a summarized schema of good quality, outperforming the baselines by an average of 22% of the F1 score.*

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. Also, on an information extraction task, as shown in Lange et al. (2010) and Moreira and Barbosa (2021), a schema is required to guide the data extraction process from these applications.

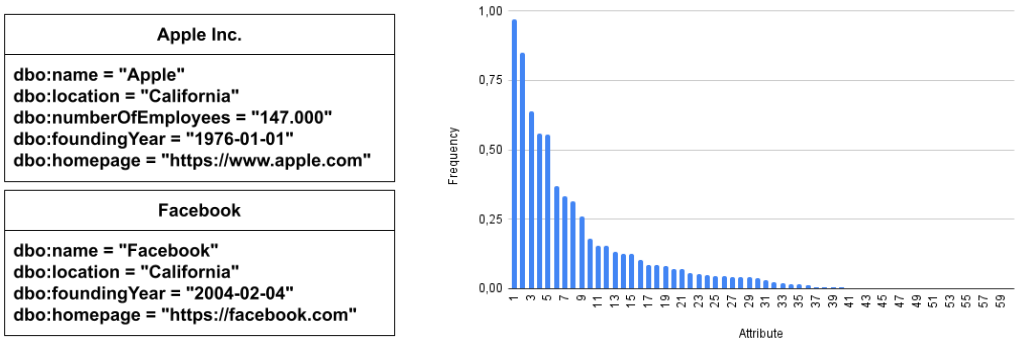
Despite being a W3C recommendation[1], many datasets do not provide or have incomplete schema-related information. To this end, schema discovery approaches have been proposed in the literature in order to identify a data schema from a dataset [Kellou-Menouer et al. 2021]. Kellou-Menouer et. al. (2021)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

Previous approaches such as Christodoulou et al. (2015) and Kellou-Menouer and Kedad (2015) have tried to infer a schema for a dataset by discovering the entity

[1]https://www.w3.org/TR/dwbp/#StructuralMetadata

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 composing 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 predefined 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 of a class.



(a) Entity schema (b) Attribute Frequency (Company class)

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

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 mine 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]; 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 six distinct entity classes extracted from DBpedia. 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 is efficient to provide 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 schema mining 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.

The remainder of the paper is organized as follows. Section 2 formalizes the main concepts. Section 3 discusses some related work and compares it with our paper. In Section 4 we define CoFFee, describing how each step works. Section 5 describes the experiments performed and the results achieved. Finally, in Section 6, we present the final considerations and guide the next steps of the work.

2. Background

In this section, we present 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 $s(e) = \{a_1, ..., a_n\}$ consists of a set of attributes that describe an entity e, e.g., for *Apple Inc.* their entity schema is $s(Apple\_Inc.) = \{homepage, location, ..., foundingYear, numberOfEmployees\}$.

Definition 2.4 (Class schema)**.** A class schema $S_C = \{a_1, ..., a_m\}$ consists of a set of meaningful attributes that represent a set of instances of C.

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

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

3. Related Work

In this paper, we propose a schema mining approach. In other words, we want to summarize a set of diverse entity schemas 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 Information Extracting 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. 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 instances of a class. The implementation of this approach considers the FP-growth algorithm. Thus, a parameter (support vector) is required to find frequent attribute patterns. 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. 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. 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 schemas is heterogeneous.

The main weakness in Wu and Weld (2007), Moreira and Barbosa (2021), Weise et al. (2016), and 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) and 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 schemas 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 schemas.

4. Proposal: CoFFee

In this section, we detail **CoFFee**, an approach for schema mining 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.

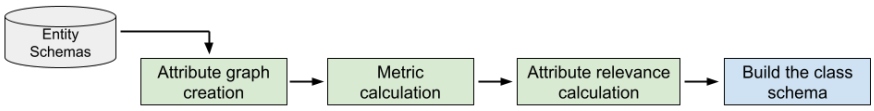
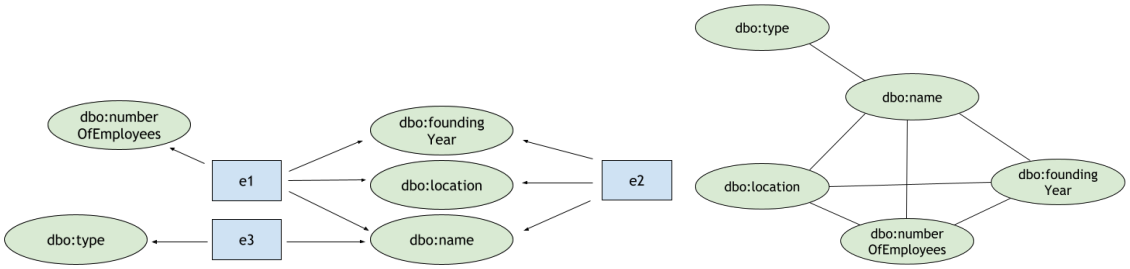


Figure 2. CoFFee's pipeline



(a) Bipartite graph (b) Atributte graph

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.

4.1. 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 4.1 (Atributte 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 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][2]. 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).

[2]In this paper, we run the experiments on DBpedia datasets that already solve this issue.

4.2. 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 4.2 (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:

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

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

Definition 4.3 (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:

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

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 to 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-occurs. 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:

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

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

4.3. 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 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:

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

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

4.4. 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.

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

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

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

where, $G(S_i, C_S)$ (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 α_i e β_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, α_i and β_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 testing 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 less 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

```
q_max ← 0
k ← 0
for j ← 1 to |R| do
    S_C ← pick top-j in R
    q ← q(S_C)                                   ▷ Eq. 5
    if q >= q_max then
        q_max ← q
        k ← j
    end if
end for
S_C ← pick top-k in R
```

5. Experimental Evaluation

In this section, we present the experimental validation of our method and discuss the achieved results. Experimental data and source code are available on github[3].

5.1. Dataset

We validate our approach over two DBpedia datasets (version 12/2021): *mappingbased-objects*[4] and *mappingbased-literals*[5]. We consider data from six classes: Film, Artist, Company, Scientist, University and Book. We choose these classes since the baselines this evaluation already explore them. We identify the instances of each class through the *rdf:type* predicate contained in the *instance types* dataset[6]. Table 1 presents statistics of the data. The **Entity Schemas** 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 schemas.

Class	Entity Schemas	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

Table 1. Dataset statistics

5.2. 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.

[3] https://github.com/ecsn/coffee

[4] https://databus.dbpedia.org/dbpedia/mappings/mappingbased-objects/2021.12.01/mappingbased-objects_lang=en.ttl.bz2

[5] https://databus.dbpedia.org/dbpedia/mappings/mappingbased-literals/2021.12.01/mappingbased-literals_lang=en.ttl.bz2

[6] https://databus.dbpedia.org/dbpedia/mappings/instance-types/2022.03.01/instance-types_lang=en_transitive.ttl.bz2

- **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 co-occur frequently above a user-defined threshold. The class schema is the set of attributes contained in the set of patterns identified by the algorithm. In our experiments, we set the parameter 0.5.

5.3. Experimental setup

Below we summarize the setups of two experiments we perform.

Experiment 1 aims at analysing 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]: **Retrival Rate (RR)** = $\frac{\sum_{i=1}^{N} \frac{S_i \cap S_C}{|S_i|}}{|S|}$ and **Relative Size (RS)** = $|S_C|/|A|$. In other words, RR measures the gain of information obtained using the class schema, while RS measures the size of the class schema concerning the universal attribute set.

Experiment 2 analyzes the quality of the class schema in comparison to a reference schema. 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 class. Table 2 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*.

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

Table 2. Schema reference information (*short version)

We use **Precision (P)** = $\frac{TP}{TP+FP}$, **Recall (R)** = $\frac{TP}{TP+FN}$, and **F-measure (F1)** = $\frac{2*P*R}{P+R}$ metrics to calculate schema quality. Where TP is the number of selected attributes that belong to the reference schema; FP is the number of attributes that were selected but that do not belong to the reference schema; and FN is the number of attributes that belong to the reference schema but have not been selected.

5.4. Results

In this section, we discuss the experiments results.

5.4.1. Experiment 1

	CoFFee		Skeleton		LOD-CM	
Class	**RR**	**RS**	**RR**	**RS**	**RR**	**RS**
Film	0.99	0.5	0.89	0.35	0.56	0.14
Artist	0.95	0.26	0.88	0.21	0.50	0.06
Company	0.75	0.15	0.67	0.11	0.57	0.08
Scientist	0.91	0.23	0.64	0.10	0.56	0.09
University	0.85	0.22	0.70	0.14	0.40	0.07
Book	0.63	0.17	0.63	0.17	0.48	0.12

Table 3. Effectiveness of approaches to summarize the class schema.

Table 3 shows the performance of the approaches concerning the retrieval rate (RR) and relative size (RS) indices. 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 4 of 6 classes evaluated. Meanwhile, the RS index falls between 0.5 (Film) and 0.15 (Company). In other words, the set of attributes selected by CoFFee offers a more summarized description of the class instances while preserving the recall. Looking at the metrics for Skeleton, the RR index is high for classes in which the schemas are less heterogeneous (e.g., Film and Artist), i.e., a lower percentage of distinct schemas (see Table 1), while the RR index is lower in classes with heterogeneous schemas (e.g., Company and Scientist).

Attribute	**CoFFee**	**Skeleton**	**LOD-CM**
dbo:name	✓	✓	✓
dbo:foundingYear	✓	✓	✓
dbo:industry	✓	✓	✓
dbo:type	✓	✓	✓
dbo:homepage	✓	✓	✓
dbo:location	✓	✓	
dbo:product	✓	✓	
dbo:numberOfEmployees	✓		
dbo:keyPerson	✓		

Figure 4. Attributes selected by the approaches (Class: Company)

Figure 4 shows the Company class attributes selected by each approach. Comparing CoFFee and Skeleton, 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 schemas. Despite the

dbo:numberOfEmployees attribute does not appear often in equivalent schemas, it does co-occur with core attributes such as *dbo:name*, *dbo:foundingYear* and *dbo:industry* in some schemas. Skeleton was built to select attributes that occur in equivalent schemas, unlike our approach that considers co-occurrence and frequency of attributes.

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. In other words, this approach fails to consider relevant attributes. In addition, the parameter defined by LOD-CM can influence the results since the frequency distribution is different for each class. Moreover, manually setting it is a 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. From these experiments we observe that CoFFee's heuristic for class attributes selection (Section 4.4) proved to be efficient.

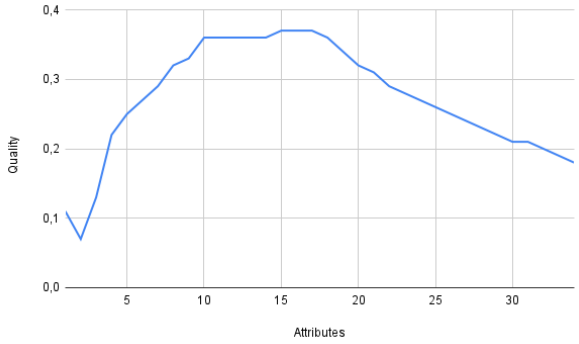


Figure 5. Quality of the class schema ordered by the relevance of the attributes. (Class: Film)

Figure 5 shows how the quality (Equation 5) 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 5 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.

5.4.2. Experiment 2

Table 4 benchmarks the proposed approach with the baselines regarding the reference schema. All approaches had a high precision (close to 1) in 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[7]. For this reason, the attribute is not present in the reference schema. A similar case also occurs in the Artist class.

[7]According to infobox template: `https://en.wikipedia.org/wiki/Template:Infobox_film`

	CoFFee			Skeleton			LOD-CM		
Class	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
Film	0.94	0.76	0.84	1.00	0.57	0.72	1.00	0.24	0.38
Artist	0.92	0.38	0.54	0.90	0.31	0.46	1.00	0.10	0.19
Company	1.00	0.47	0.64	1.00	0.37	0.54	1.00	0.16	0.27
Scientist	1.00	0.33	0.49	1.00	0.15	0.26	1.00	0.12	0.22
University	1.00	0.22	0.36	1.00	0.13	0.24	1.00	0.06	0.11
Book	1.00	0.21	0.35	1.00	0.21	0.35	1.00	0.14	0.25
AVG	0.97	0.39	0.56	0.98	0.29	0.44	1.00	0.13	0.24

Table 4. Class schema quality compared to the reference schema.

CoFFee outperforms the baselines in the evaluated classes. It achieves an upper average difference in F1 of 0.12 for Skeleton and 0.32 for LOD-CM. The reason for this is that CoFFee achieves a high recall value. 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 class, where CoFFee selects 13 attributes, while Skeleton and LOD-CM 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.

6. Conclusion

In this paper, we address the schema mining 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 efficient in selecting the most relevant attributes by combining co-occurrence and frequency. We performed experiments with data extracted from six DBpedia classes 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. As future directions, we intend to create a tool that provides structural metadata from the results obtained by CoFFee to describe the content and leverage the use of datasets that do not have these types of metadata.

References

[Adolphs et al. 2011] 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.

[Christodoulou et al. 2015] 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.

[Dong and Srivastava 2015] Dong, X. L. and Srivastava, D. (2015). *Schema Alignment*, pages 31–61. Springer International Publishing, Cham.

[Hassanzadeh et al. 2013] 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 et al. 2019] 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 et al. 2021] 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 and Kedad 2015] Kellou-Menouer, K. and Kedad, Z. (2015). Schema discovery in rdf data sources. In *International Conference on Conceptual Modeling*, pages 481–495. Springer.

[Lange et al. 2010] Lange, D., Böhm, C., and Naumann, F. (2010). Extracting structured information from wikipedia articles to populate infoboxes. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, page 1661–1664, New York, NY, USA. Association for Computing Machinery.

[Moreira and Barbosa 2021] Moreira, J. and Barbosa, L. (2021). Deepex: A robust weak supervision system for knowledge base augmentation. *J. Data Semant.*, 10(3-4):309–325.

[Moreira et al. 2021] Moreira, J., Neto, E. C., and Barbosa, L. (2021). Analysis of structured data on wikipedia. *International Journal of Metadata, Semantics and Ontologies*, 15(1):71–86.

[Queiroz-Sousa et al. 2013] 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.

[Wang et al. 2015] 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.

[Weise et al. 2016] 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 and Weld 2007] 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 and Luo 2017] 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.