A lightweight approach to extract interschema properties from structured, semi-structured and unstructured sources in a big data scenario

International Journal of Information Technology & Decision Making
Vol. 19, No. 3 (2020) 849–889
World Scientific Publishing Company
DOI: 10.1142/S0219622020500182

A lightweight approach to extract interschema properties from structured, semi-structured and unstructured sources in a big data scenario

Abstract

The knowledge of interschema properties (e.g., synonymies, homonymies, hyponymies, subschema similarities) plays a key role for allowing decision making in sources characterized by disparate formats. In the past, a wide amount and variety of approaches to derive interschema properties from structured and semi-structured data have been proposed. However, currently, it is esteemed that more than 80% of data sources are unstructured. Furthermore, the number of sources generally involved in an interaction is much higher than in the past. As a consequence, the necessity arises of new approaches to address the interschema property derivation issue in this new scenario. In this paper, we aim at providing a contribution in this setting by proposing an approach capable of uniformly extracting interschema properties from a huge number of structured, semi-structured and unstructured sources.

Keywords: Unstructured sources; Interschema Property Derivation; Structuring Unstructured Data; Big Data

1 Introduction

In the last few years, we are assisting to a real revolution in the information system scenario. In fact, the number and the size of available data sources have dramatically increased. Furthermore, most of them (i.e., more than 80%) are unstructured [18, 17]. These facts are rapidly changing the scientific and technological "coordinates" of the information system research field [9, 35, 33]. As a consequence of this phenomenon, even issues successfully addressed in the past must be re-considered and re-investigated. One of these issues is certainly the derivation of interschema properties (i.e., *intensional* relationships between concepts represented in different data sources [53], like synonymies, homonymies, hyponymies, overlappings, subschema similarities). This topic has been widely studied in the past [61, 10]; however, the proposed approaches generally considered structured or, at most, semi-structured sources. Furthermore, the number of involved sources, for which most of past approaches were targeted to, was very small, if compared with a typical current source interaction and cooperation scenario.

Interschema property derivation is not just one of the many topics to re-investigate in information systems cooperation field. Actually, it represents the basis of most of the other issues: for instance, the knowledge of interschema properties is necessary for source integration, the construction of data warehouses and data lakes, data analytics, and so forth.

In this paper, we aim at providing a contribution in this setting. Indeed, we propose a novel approach to uniformly perform the extraction of interschema properties from structured, semi-structured and unstructured sources. Our approach has been specifically conceived having in mind two peculiarities that should characterize it, namely: *(i)* the capability of handling unstructured sources; *(ii)* the lightweightness, making it capable of managing a huge number of data sources.

As for the capability of handling unstructured sources, our approach is provided with a preliminary step capable of "structuring" unstructured sources, i.e., of (at least partially) deriving a structure for them. This is possible because it assumes that each unstructured source (e.g., a video, an audio, an image, a text) has associated a list of keywords describing it. The "structuring" process is based exactly on these keywords. This is another main contribution of this paper, which, generally speaking, allows the unstructured sources to be uniformly handled as the structured and the semi-structured ones. With regard to this aspect, some clarifications of what we intend with the terms "structured" and "semi-structured" sources are in order. In particular, we use these terms as they are generally adopted in databases and information systems research field. Here, a structured source consists of some concepts, each having a precise set of attributes and relationships with other concepts of the source. A semi-structured source has similar characteristics, but the set of attributes and relationships characterizing a given concept is handled in a more flexible fashion. Indeed, given a property p or a relationship r of a concept c, some instances of c might have exactly one instance of r and/or one instance of p; other instances of c might have more instances of r and/or more instances of p; finally, other ones might have no instances of r and/or no instances of c. A classical example of structured sources is a relational database (that can be conceptually represented by means of an E/R diagram). A classical example of a semi-structured source is an XML document (that can be conceptually represented by means of a DOM).

Unstructured sources are videos, audios, images or texts. They do not generally have a conceptual representation showing their concepts, along with the corresponding properties and relationships. However, they are generally provided with a set of keywords, denoting the main concepts they are representing. The purpose of our approach for "structuring" unstructured sources is exactly the derivation of the relationships existing among the concepts represented by the keywords associated with unstructured sources. If we are capable of performing this task, unstructured sources can be handled similarly to structured and semi-structured ones. Furthermore, their analysis and management could benefit from the wide amount of results found in the past for structured and semi-structured sources. Finally, the integration, the cooperation and the simultaneous querying of structured, semi-structured and unstructured sources are possible.

Our approach also differs from other ones previously presented in related research fields and that could be in principle extended to address the problem we are considering in this paper. Think, for instance, of ontologies. We could link each available keyword to an ontology and use this last one as the "infrastructure" through which establishing the relationships among the keywords, once these last have been linked to it. This approach is certainly valid, but it needs a support ontology. As a consequence, it can be employed only in those application fields for which an ontology exists and only if all the involved information sources can be mapped onto a unique ontology. If only some of the involved unstructured sources can be referred to an ontology and/or some of them can be mapped onto another ontology and/or, finally, some of them cannot be referred to any ontology, this way of proceeding cannot be adopted. From this point of view, our approach is more general because it can be applied in all cases, independently of the presence of none, one or more ontologies, which the unstructured sources can be referred to. It only needs a thesaurus. If there exists a specific thesaurus for the scenario which the unstructured sources into examination belongs to, then it uses this theasurus. Otherwise, it can still work with a general-purpose thesaurus, like BabelNet [50]. Clearly, if the unstructured sources are specific of a certain field, the availability of a specific thesaurus can help to obtain a better accuracy. However, if this kind of thesaurus is not available, a general-purpose one is sufficient to proceed even if, in this case, accuracy could be lower.

As for the lightweightness of our approach, we observe that, in a big data scenario, such as the one currently characterizing the information system field, a new proposed approach must take scalability into a primary consideration [40, 39]. As a matter of fact, the sources interacting in every task are always very numerous and large (think, for instance, of a data lake constructed to support data analytics in an organization) and the time allowed for each transaction is very limited (think, for instance, of streaming applications). As a consequence, even approaches considered very scalable in the past (such as DIKE [55], MOMIS [7], and Cupid [41]) are not adequate anymore. In our opinion, the tests performed to evaluate our approach and described in Section 6 confirm that it is really capable of satisfying the lightweightness requirement without sacrificing, if not to a very small extent, result accuracy.

Summarizing, the main contribution of this paper is an overall procedure capable of extracting interschema properties from structured, semi-structured and unstructured sources. Our procedure is lightweight because it has been specifically conceived to operate on big data. This feature is deeply investigated in the paper, where we analyze its computational requirements and compare them with the one of similar approaches conceived to work on smaller (only) structured and semi-structured data sources. In spite of its lightweightness, the accuracy of our procedure is very satisfying, as witnessed by the quantitative evaluations presented in the paper. An important component of our approach, which could also be extrapolated to other contexts, is the technique for “structuring” unstructured sources whose distinctive peculiarities have been described above.

The rest of this paper is organized as follows: in Section 2, we examine related literature. In Section 3, we introduce a source representation model that we exploit in our tasks. In Section 4, we show our approach for the construction of a “structured representation” of unstructured data sources. In Section 5, we present our interschema property derivation approach. In Section 6, we present some experiments that we performed to test our approach. Finally, in Section 7, we draw our conclusions and have a look at some possible future developments of this research.

2 Related Literature

2.1 Schema matching for structured and semi-structured sources

Schema matching is one of the most investigated topics in past database research. The first schema matching approaches proposed by researchers were manual and operated only on structured databases.

Subsequently, researchers proposed semi-automatic or automatic schema matching approaches capable of handling both structured and semi-structured data sources. With the advent of big data, unstructured sources are becoming more and more frequent and important.

Schema matching approaches were thought to consider several kinds of heterogeneity; the most relevant of them are lexicographic, structural and semantic ones. The first deals with names and terms; the second considers type formats, data representation models and structural relationships among concepts; the third regards the meaning of involved data.

Let us see, now, in more detail, an overview of several approaches to perform schema matching from the beginning to the present day.

In [14], an approach to transform structured documents by leveraging schema graph matching is proposed. In particular, an XML schema to map each structured document is defined; for this purpose, some XSLT scripts are automatically generated. In [41], Cupid, a system for deriving interschema properties among heterogeneous sources, is proposed. Cupid leverages two different matchings, namely the *structure* and the *linguistic* ones. In [7], MOMIS, a system supporting querying and information source integration in a semi-automatic fashion, is presented. MOMIS implements a clustering procedure for the extraction of interschema properties. DIKE and XIKE [55, 19, 54], as well as the approaches described in [16, 20], also belong to this generation. An overview of this generation of schema matching approaches can be found in [61, 10].

More recent approaches, which significantly differ from the classical ones, are based on probabilistic methods, applied to networks of schemas [26]. They allow the definition of network-level integrity constraints for matching, as well as the analysis of query/click logs [21, 49], specifying the class of desired user-based schema matching.

In [3], an XML-based schema matching approach conceived to operate on large-scale schemas is presented. This approach leverages Prufer sequences. It performs a two-step activity; during the former step it parses XML schemas in schema trees; during the latter one, it exploits Label Prufer Sequences (LPS) to capture schema tree semantic information. In [51], SMART, a Schema Matching Analyzer and Reconciliation Tool, designed for the detection and the subsequent reconciliation of matching inconsistencies, is proposed. SMART is semi-automatic because it requires the intervention of an expert for the validation of results. In [44], the authors propose an approach to determine the semantic similarity of terms using the knowledge present in the search history logs from Google. For this purpose, they exploit four techniques that evaluate: *(i)* frequent co-occurrences of terms in search patterns; *(ii)* relationships between search patterns; *(iii)* outlier coincidence on search patterns; *(iv)* forecasting comparisons. In [5], a framework for the management of a data lake through the corresponding metadata is proposed. This framework leverages schema matching techniques to identify similarities between the attributes of different datasets. These techniques consider both schemas (specifically, attribute types and dependencies) and instances (specifically, attribute values) [10]. The framework integrates different schema matching approaches proposed in the last years, like graph matching, usage-based matching, document content similarity detection and document link similarity detection. [45] proposes an instance-based approach to find 1-1 schema matches. It combines the semantics provided by Google and regular expressions. It does not work well in a scenario where sources are very heterogeneous and data are stored in their raw way. Another instance-based approach is presented in [27]. It faces the heterogeneity of the different schemas by leveraging an ad-hoc mapping language.

Most schema matching approaches based on similarities often filter out unnecessary matchings and information [59] in such a way as to operate easier and faster.

As we have seen in this overview, schema matching has been widely investigated in the past for very heterogeneous scenarios, and very different approaches have been adopted to reach disparate goals. In this "mare magnum" of approaches, ours is characterized by the following features: *(i)* it has been specifically conceived to handle also unstructured sources; *(ii)* it has been designed to be scalable and, therefore, it is lightweight; *(iii)* it is automatic; *(iv)* in spite of these two last features, it presents a good accuracy, as we will see in Section 6.

2.2 Approaches to represent unstructured sources

The representation mechanisms of unstructured sources (basically texts) are mainly based on two strategies, namely analysis of contents and analysis of references [66]. The former infers a representation of a document from the corresponding content, whereas the latter focuses on relationships among documents. Clearly, our interest is mainly on the former strategy, because its objective is similar to the one of our approach.

The most basic approach to represent texts leverages Bags of Words (BOW) [6, 65]. In this case, machine learning techniques are used to identify the set of words that mostly characterizes a text [34, 38]. Some more sophisticated strategies are based on the extraction of sentences [22]. In this case, a text is mapped onto semantic spaces, such as WordNet or Wikipedia. Another strategy is Explicit Semantic Analysis (ESA) [23], which mixes BOW and document references. In ESA, the relatedness between documents is computed by extracting similarities between the concepts identified within them, thanks to the cross-references expressed therein.

An important model in the BOW context is word2vec [46, 47]. This model is based on neural networks. It constructs a vector space and associates each word of the text into examination with a vector in this space in such a way that words sharing common contexts have close corresponding vectors in the vector space. The word2vec model was extended to the doc2vec one [36], which exploits similarities and contextual information of each word to reduce the dimensionality of the vector space. Other approaches reach the same objective (i.e., dimensionality reduction) by means of Latent Semantic Analysis [30], which exploits matrix decomposition methods.

Word-based methods are currently flanked by concept-based ones. As an example, [64, 63] introduce the idea of Bag of Concepts, in place of Bag of Words. In this approach, concepts are generated by disregarding semantic similarities between words. Semantic similarities have been considered only recently [31].

Another relevant set of approaches use ontologies or, in general, external sources of semantics, to generate conceptual representations of documents by matching document terms with ontology concepts (see, for instance, [11, 28, 69, 2]). The performance of these approaches is strongly related to the quality of the adopted external sources. As a consequence, in these approaches, very specific domains can strongly benefit from the availability of dedicated ontologies.

The approaches examined above generally consider only texts; they do not operate with other forms of unstructured sources, such as videos. Furthermore, they terminate with the derivation of keywords or key concepts representing a source. In fact, none of them tries to go a step over, i.e., to define a certain "structure" for an unstructured source, which is one of the objectives of this paper.

An attempt to define a "structure" for an unstructured source can be found in [42]. This approach generates a rowset with n attributes, i.e., a tabular representation from unstructured data. A single rowset is a set of tuples and is equivalent to a relation in relational databases; logical associations may exist between rowsets, but these are not explicitly defined. The schema of a rowset may be defined on read. Transformation functions, possibly based on fuzzy logic, are used to properly read the complex unstructured data and map them on the rowset schema. These functions are also exploited to address the data variety issue, by means of an interface for the dataset extraction, which is unified and valid for all the sources. Different transformation functions can be used to map different unstructured data onto the same schema. The content of a rowset depends on the membership function associated with a fuzzy logic and on the possible constraints regarding it. However, data extraction is only one of the steps defined in [42], which develops a general data processing system based on an Extract, Process, and Store (EPS) paradigm.

From the above description, it appears evident that the approach of [42] shares several features with ours; in particular, the purpose of structuring unstructured data is common to both of them. However, the two approaches also present several differences. Indeed, for the structuring task, the approach of [42] strongly depends on user defined transformation functions and on rowset schemas (which are not automatically inferred from the sources). Now, the definition of both the functions and the schema may be difficult for complex sources. Furthermore, mapping more sources on the same schema requires a manual integration step, which, again, may be a difficult task when the number of involved sources is high. On the other hand, querying obtained data sources is particularly effective with the use of fuzzy techniques and the declarative U-SQL query language characterizing the approach of [42]. On the contrary, in our proposal, to perform the structuring of unstructured sources, we leverage network analysis, as well as lexical and string similarities, for automatically deriving a general and uniform schema of different unstructured sources. In fact, as we will see in the following, unstructured sources are "structured" by first representing them as a network, starting from a set of keywords associated with them; then, this structure is enriched by adding arcs that link nodes having lexical or string similarities even if they belong to different sources. As a consequence, it is possible to state that the approach presented in [42] is more effective and flexible in querying data lake contents, but it requires a more complex design phase, with a heavy human intervention, difficult to sustain in presence of numerous data sources. On the contrary, our approach simplifies the structuring phase, because it does not need a preliminary structure to be used as a model, and it does not require a human intervention. On the other side, its querying capabilites are limited to the summarization of unstructured sources provided by the keywords representing them. Therefore, in a certain sense, our approach and the one of [42] can be considered orthogonal.

3 A network-based model for uniformly representing structured, semi-structured and unstructured sources

In this section, we present a network-based model for uniformly representing data sources of different formats. This model will be extensively used in the rest of this paper. In order to understand the peculiarities of our model, we assume to have a set DS of m data sources of interest possibly characterized by different data formats.

$$DS = \{D_1, D_2, \cdots, D_m\}$$

Each data source D_k has associated a rich set $\mathcal{M}_k$ of metadata. We indicate with $\mathcal{M}_{DS}$ the repository of the metadata of all the data sources of DS:

$$\mathcal{M}_{DS} = \{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_m\}$$

Given the source D_k, in order to represent the information content stored in $\mathcal{M}_k$, our model starts from a notation typical of XML, JSON and many other semi-structured data models. According to this notation, Obj_k denotes the set of all the objects stored in $\mathcal{M}_k$. Obj_k consists of the union of three subsets:

$$Obj_k = Att_k \cup Smp_k \cup Cmp_k$$

where:

- Att_k denotes the set of the attributes of $\mathcal{M}_k$;
- Smp_k indicates the set of the simple elements of $\mathcal{M}_k$;
- Cmp_k represents the set of the complex elements of $\mathcal{M}_k$.

Here, the meaning of the terms “attribute”, “simple element” and “complex element” is the one typical of semi-structured data models.

$\mathcal{M}_k$ can be also represented as a graph:

$$\mathcal{M}_k = \langle N_k, A_k \rangle$$

N_k is the set of the nodes of $\mathcal{M}_k$. There is a node n_{k_j} in N_k for each object o_{k_j} of Obj_k. According to the structure of Obj_k, N_k consists of the union of three subsets:

$$N_k = N_k^{Att} \cup N_k^{Smp} \cup N_k^{Cmp}$$

where N_k^{Att} (resp., N_k^{Smp}, N_k^{Cmp}) denotes the set of the nodes corresponding to Att_k (resp., Smp_k, Cmp_k). There is a biuniovocal correspondence between a node of N_k and an object of Obj_k. Therefore, in the following, we will use these two terms interchangeably. Each node has associated a name that identifies it in the schema which the corresponding element or attribute belongs to.

Let x be a complex element of $\mathcal{M}_k$. We denote by Obj_x the set of the objects directly contained in x and by N_x^{Obj} the set of the corresponding nodes. Finally, let x be a simple element of $\mathcal{M}_k$. We indicate by Att_x the set of the attributes directly contained in x and by N_x^{Att} the set of the corresponding nodes.

A_k denotes the set of the arcs of $\mathcal{M}_k$. It consists of three subsets:

$$A_k = A'_k \cup A''_k \cup A'''_k$$

where:

- $A'_k = \{(n_x, n_y, L_{xy}) | n_x \in N_k^{Cmp}, n_y \in N_{n_x}^{Obj}\}$; in other words, there is an arc in A'_k from a complex element of $\mathcal{M}_k$ to each object directly contained in it. L_{xy} represents the label of A'_k.

- $A''_k = \{(n_x, n_y, L_{xy}) | n_x \in N_k^{Smp}, n_y \in N_{n_x}^{Att}\}$; in other words, there is an arc in A''_k from a simple element of $\mathcal{M}_k$ to each attribute directly contained in it. L_{xy} represents the label of A''_k.

- $A'''_k = \{(n_x, n_y, L_{xy}) | n_x \in N_k, n_y \in N_k, D_k \text{ is unstructured}, \sigma(n_x, n_y)=\texttt{true}\}$. Here, $\sigma(n_x, n_y)$ is a function that receives two nodes and returns `true` if there exists a similarity between n_x and n_y. For instance, $\sigma(n_x, n_y)$ could return `true` if the concepts represented by n_x and n_y are semantically similar or if the names identifying n_x and n_y in the corresponding schema present a high string similarity. L_{xy} represents the label of A'''_k.

As for the label L_{xy} associated with each arc, in the current version of this model, it is `NULL` for the arcs of A'_k and A''_k. However, we do not exclude that, in the future, enrichments of our model might lead us to use this field for storing some knowledge. Instead, L_{xy} has an important meaning for the arcs of A'''_k. In fact, as will be clear in Section 5, it is used to denote the strength of the correlation between n_x and n_y.

From an abstract point of view, there is a "fil rouge" linking the three subsets of A_k, which leads to the concept of homophily in Social Network Analysis. Indeed, A'_k, A''_k and A'''_k are the three possible ways to represent the links between a concept and its "direct homophiles", i.e., the other concepts that can contribute to define (at least partially) its meaning.

4 Structuring an unstructured source

Our network-based model for uniformly representing and handling data sources with disparate formats is perfectly fitted for semi-structured sources. Indeed, it is sufficient:

- deriving the metadata of the source (if not yet provided) by applying one of the several techniques and tools defined for this purpose w.r.t. the various kinds of format;

- defining a complex element to represent the source as a whole;

- introducing a complex element, a simple element and an attribute for each complex element, simple element and attribute present in the metaschema of the source;

- defining an arc of A'_k from the source to the root of the document;
- introducing an arc of A'_k or A''_k for each relationship existing between the objects composing the source metadata.

Clearly, our model is sufficiently powerful to represent structured data. Indeed, it is sufficient:

- deriving the E/R schema of the source (if not yet provided) by performing a classical database reverse engineering activity;
- defining a complex element to represent the source as a whole;
- introducing a complex element for each entity of the E/R schema and an attribute for each attribute of the schema;
- defining an arc of A'_k from the complex element corresponding to the source to each complex element associated with an entity of the E/R schema;
- introducing an arc of A''_k from an entity to each of its attributes;
- defining an arc of A'_k for each one-to-many relationship of the E/R schema; this arc is from the entity participating to the relationship with a maximum cardinality equal to 1 to the entity participating with a maximum cardinality equal to N;
- representing a many-to-many relationship without attributes as a pair of one-to-many relationships and, then, modeling them accordingly;
- representing a many-to-many relationship R with attributes that connects two entities E_1 and E_2 as an entity having the same attributes as R and linked to E_1 and E_2 by means of two one-to-many relationships; the new entity and the new relationships are then suitably modelled by applying the rules defined in the previous cases.

The highest modeling difficulty regards unstructured data because it is worth avoiding a flat representation consisting of a simple element for each keyword provided to denote the source content. As a matter of fact, this flat representation would make the reconciliation, and the next integration, of an unstructured source with the other semi-structured and structured sources of DS very difficult. This is a very challenging issue to address. In the following, we propose our approach to "structure" unstructured sources. As pointed out in the Introduction, this is one of the main contributions of this paper. It is in itself a major issue in the current information systems scenario and, at the same time, plays a key role to provide our interschema property derivation approach with the capability of operating on sources with disparate formats.

Our approach assumes that each unstructured source into consideration (e.g., a video, an audio, an image, a text) is provided with a list of keywords describing it[1]. They will play a key role, as will be clarified in the following. We observe that this assumption is not particularly strong or out

[1] Here, we assume that the list is ordered and the order is the one in which the keywords appear in the list.

of place. As a matter of fact, in the reality, most video, image or audio providers associate a list of keywords (sometimes, in the form of tags) with the contents they deliver. As for text, representing keywords can be also easily derived through suitable techniques, like TF-IDF [43].

Our approach consists of four phases, namely: *(1)* creation of nodes; *(2)* management of lexical similarities; *(3)* management of string similarities; *(4)* management of (temporary) duplicated arcs. We describe these phases below.

- **Phase 1: Creation of nodes.** During this phase, our approach creates a complex node representing the source as a whole and a simple node for each keyword[2]. Furthermore, it adds an arc of A'_k from the node associated with the source to any node corresponding to a keyword. Initially, there is no arc between two keywords. To determine the arcs to add, the next phases are necessary.

- **Phase 2: Management of lexical similarities.** During this phase, our approach handles lexical similarities. For this purpose, it leverages a suitable thesaurus. Taking the current trends into account, this thesaurus should be a multimedia one; for this purpose, in our experiments, we have adopted BabelNet [50]. In particular, our approach adds an arc of A'''_k from the node n_{k_1}, corresponding to the keyword k_1, to the node n_{k_2}, corresponding to the keyword k_2, and vice versa, if k_1 and k_2 have at least one common lemma[3] in the thesaurus. Furthermore, it transforms the nodes n_{k_1} and n_{k_2} from simple to complex. The new arcs have a label corresponding to the number of common lemmas for k_1 and k_2 in the thesaurus.

- **Phase 3: Management of string similarities.** During this phase, our approach derives string similarities and states that there exists a similarity between two keywords k_1 and k_2 if the string similarity degree $kd(k_1, k_2)$, computed by applying a suitable string similarity metric on k_1 and k_2, is "sufficiently high" (see below). In this case, it adds an arc of A'''_k from n_{k_1} to n_{k_2}, and vice versa. Both the two arcs have $kd(k_1, k_2)$ as their label. We have chosen N-Grams [32] as string similarity metric because we have experimentally seen that it provides the best results in our context. In particular, we have selected bi-grams as the best trade-off between accuracy and costs. In fact, mono-grams would require a lower cost but they would also return a lower accuracy than bi-grams. By contrast, tri-grams would guarantee a very high accuracy but at the expense of the computational cost, which would be excessive. Again, if n_{k_1} and n_{k_2} are simple nodes, our approach transforms them into complex ones.

  Now, we illustrate in detail what "sufficiently high" means and how our approach operates. Let $KeySim$ be the set of the string similarities for each pair of keywords of the source into consideration. Each record in $KeySim$ has the form $\langle k_i, k_j, kd(k_i, k_j)\rangle$. Our approach first computes the maximum keyword similarity degree kd_{max} present in $KeySim$. Then, it examines each keyword similarity registered therein. Let $\langle k_1, k_2, kd(k_1, k_2)\rangle$ be one of these similarities. If

[2] Here and in the following, to make the presentation smoother, we use the term "complex node" to indicate a node belonging to N_k^{Cmp} and the term "simple node" to denote a node of N_k^{Smp}. Furthermore, we use the term "source" (resp., "keyword") to denote both the source (resp., a keyword) and the corresponding node associated with it.

[3] In this paper, we use the term "lemma" according to the meaning it has in BabelNet [50]. Here, given a term, its lemmas are other objects (terms, emoticons, etc.) that contribute to specify its meaning.

$((kd(k_1,k_2) \geq th_k \cdot kd_{max})$ and $(kd(k_1,k_2) \geq th_{kmin}))$, which implies that the keyword similarity degree between k_1 and k_2 is among the highest ones in $KeySim$ and that, in any case, it is higher than or equal to a minimum threshold, then it concludes that there exists a similarity between n_{k_1} and n_{k_2}. We have experimentally set $th_k = 0.70$ and $th_{kmin} = 0.50$.

Observe that the choice to consider string similarities, in particular the one to adopt N-Grams as the technique for detecting string similarities, makes our approach robust against mispelling errors possibly present in the keywords. In fact, as shown in [25], N-Grams is well suited to handle also this kind of error.

- **Phase 4: Management of (temporary) duplicated arcs.** This phase is devoted to handle the possible simultaneous presence of both lexical and string similarities for the same pair of keywords. Indeed, it may occur that, for a pair of nodes n_{k_1} and n_{k_2}, there are two arcs from n_{k_1} to n_{k_2} belonging to A_k''' and generated by both lexical and string similarities, and two arcs from n_{k_2} to n_{k_1}. In this case, the two arcs from n_{k_1} to n_{k_2} corresponding to these two forms of similarities, must be merged in only one arc, which has associated a label denoting both the number of common lemmas between k_1 and k_2 in BabelNet and the value of $kd(k_1,k_2)$. The same happens for the two arcs from n_{k_2} to n_{k_1}.

From this description, it emerges that, at the end of the four phases, given two nodes n_{k_1} and n_{k_2}, four cases may exist, namely:

1. There is no arc from n_{k_1} to n_{k_2}.
2. A pair of arcs derived from a lexical similarity links them. In this case, the two arcs actually coincide (also in their labels); therefore, one of them can be removed. Note that the choice of the arc to be removed has deep implications in the definition of the topology of the corresponding network. Indeed, one of the two nodes involved (i.e., the source node of the maintained arc) will be certainly a complex node, whereas the other one may be a simple node (if no other arc starts from it) or a complex node (if at least another arc, different from the removed one, starts from it). In turn, the topology of the network has implications in the nature and the quality of the interschema properties that can be extracted, as will be clear in Section 5. Therefore, it is appropriate that the choice of the arc to be removed is not random and that a clear rule guiding it is defined. The rule that we chose for our approach is the following: given a pair of arcs between two nodes n_{k_1}, corresponding to the keyword k_1, and n_{k_2}, corresponding to the keyword k_2, with k_1 preceding k_2 in the list of keywords associated with the source D_k, the arc from n_{k_1} to n_{k_2} is maintained and the one from n_{k_2} to n_{k_1} is removed.
3. A pair of arcs derived from a string similarity links them. As in the previous case, the two arcs coincide and one of them is removed. The policy adopted to determine the arc to remove is the same as the one followed in the previous case.
4. A pair of arcs derived from Phase 4 links them. As in the previous case, the two arcs coincide and one of them is removed.

Actually, arc labels introduced above are not necessary in our approach for the extraction of semantic relationships described in Section 5. However, we have decided to maintain them in our model because we aim at providing an approach to "structure" unstructured sources that is general and that may be adopted in several future applications, some of which could benefit from this information.

Moreover, we point out that, in the prototype implementing our approach, in order to increase its efficiency, we directly added only one arc, namely (n_{k_1}, n_{k_2}), during Phases 2, 3 and 4, instead of adding two arcs and of removing one of them at the end of the four phases.

4.1 Example

In this section, we propose an example of how our approach to construct a "structured" representation of an unstructured source operates. In particular, the unstructured source into consideration is a video, which talks about environment and pollution. As we said before, for each unstructured source, our approach begins from a list of keywords representing its content. In order to keep our description simple and clear, in this example, we assume that our video has a limited number of keywords, namely the ones shown in Figure 1.

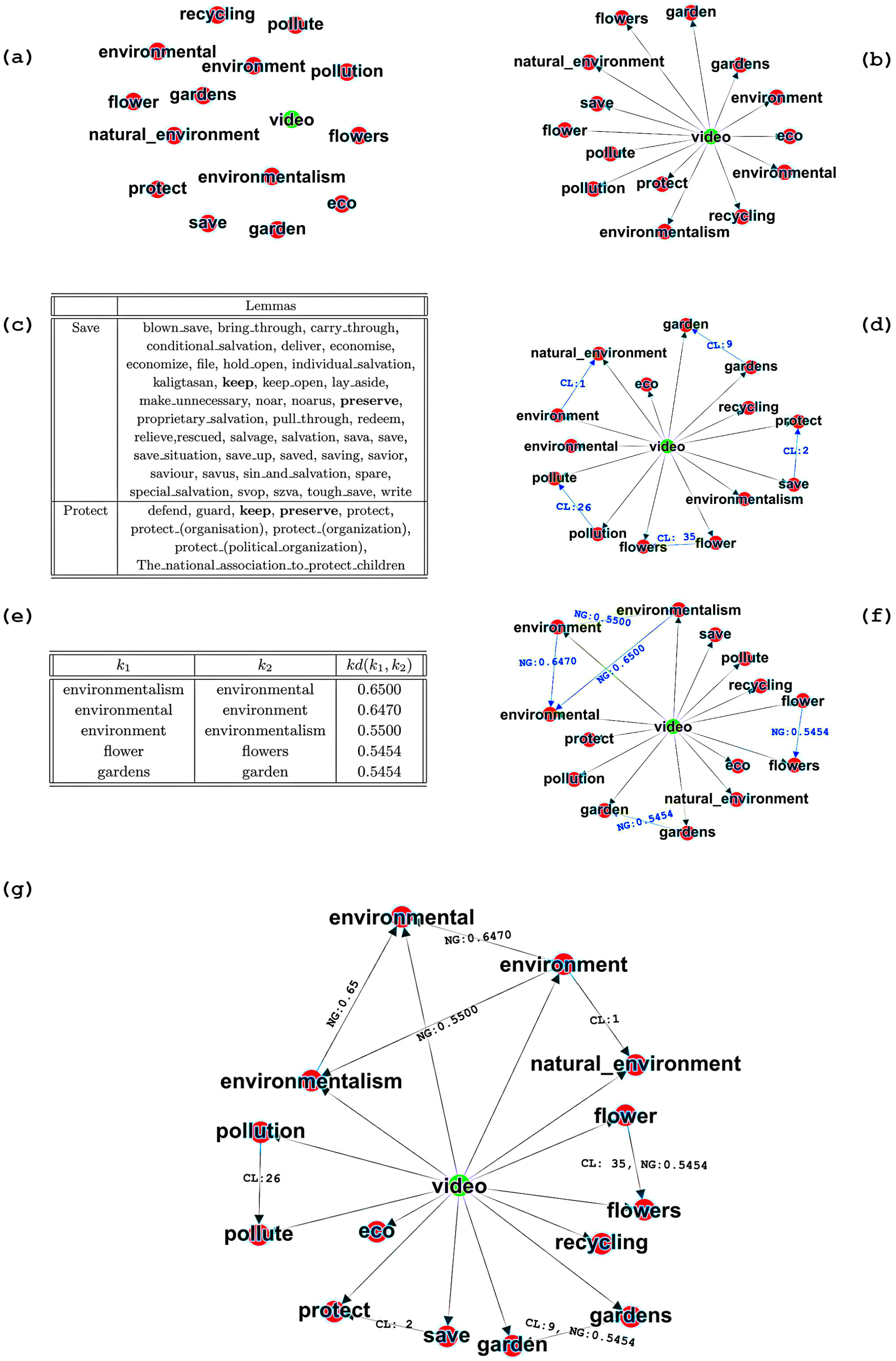
Our approach starts with Phase 1. As we can see in Figure 1(a), during this phase, it constructs a graph having a node for each keyword. A further node is added to represent the video as a whole; nodes representing keywords are colored in red, whereas the other one is colored in green. Following our strategy, in Figure 1(b), we added an arc from the node representing the whole video to each node associated with a keyword.

Now, Phase 2 starts. During this phase, our approach uses a thesaurus. In our example, we leveraged BabelNet. In particular, let k_1 and k_2 be two keywords of Figure 1(a) having at least one common lemma in BabelNet. An arc is added from the node n_{k_1}, associated with k_1, to the node n_{k_2}, associated with k_2, and vice versa. In Figure 1(c), we show two keywords ("Save" and "Protect") and the corresponding lemmas in BabelNet. Common lemmas (i.e., "keep" and "preserve") are in bold. Since "Save" and "Protect" have at least one common lemma, an arc is added between the corresponding nodes in Figure 1(d)[4]. This arc is highlighted in blue. Each arc has a label representing the number of common lemmas between the corresponding keywords in BabelNet.

After having examined lexical similarities, Phase 2 terminates and our approach proceeds with Phase 3, which leverages string similarities. In particular, let k_1 and k_2 be two keywords of Figure 1(a) having a string similarity degree higher than or equal to $th_k \cdot kd_{max}$ and, at the same time, higher than or equal to th_{kmin}. An arc is added from the node n_{k_1}, corresponding to k_1, to the node n_{k_2}, corresponding to k_2. In Figure 1(e), we report the pairs of keywords that satisfy this feature. In Figure 1(f), we added an arc for each pair of keywords of Figure 1(e). Here, to better highlight them, we have omitted the arcs constructed during Phase 2. Again, these arcs are highlighted in blue. Each arc has a label representing the string similarity degree (computed by means of N-Grams) between the corresponding keywords.

Finally, in Figure 1(g), Phase 4 of our approach combines the arcs derived in Phases 2 and 3. In particular, it may happen that, for a pair of keywords (see, for instance, the keywords "garden" and

[4] Here, we have directly added only one arc between "Save" and "Protect", instead of adding two arcs and removing one of them later, after the four phases.



(c)

	Lemmas
Save	blown_save, bring_through, carry_through, conditional_salvation, deliver, economise, economize, file, hold_open, individual_salvation, kaligtasan, **keep**, keep_open, lay_aside, make_unnecessary, noar, noarus, **preserve**, proprietary_salvation, pull_through, redeem, relieve,rescued, salvage, salvation, sava, save, save_situation, save_up, saved, saving, savior, saviour, savus, sin_and_salvation, spare, special_salvation, svop, szva, tough_save, write
Protect	defend, guard, **keep**, **preserve**, protect, protect_(organisation), protect_(organization), protect_(political_organization), The_national_association_to_protect_children

(e)

k_1	k_2	$kd(k_1, k_2)$
environmentalism	environmental	0.6500
environmental	environment	0.6470
environment	environmentalism	0.5500
flower	flowers	0.5454
gardens	garden	0.5454

Figure 1: Graphical representation of our approach to derive a “structure” for an unstructured source

"gardens"), two arcs have been generated, one in Figure 1(d) and one in Figure 1(f). In this case, in Figure 1(g), the two arcs are substituted by only one arc, representing both of them. The label of this arc reports the label of both the original ones.

5 Extracting interschema properties from disparate sources

We are now ready to illustrate our strategy for uniformly extracting interschema properties from structured, semi-structured and unstructured sources. Here, we assume that the content of the sources of interest is represented by means of the model described in Section 3, and that our approach to "structure" unstructured sources, described in Section 4, has been already applied on all unstructured sources.

Before delving into a detailed description of our approach, a discussion about the role played by source metadata, and about the consequences of this role, is in order. Indeed, as previously pointed out, our approach assumes that some metadata are available for each structured, semi-structured and unstructured source. This assumption is important because both our approach for structuring unstructured sources and our approach for extracting interschema properties use these metadata. It is, then, of outmost importance to analyze the possible issues (and the corresponding solutions) in obtaining good quality metadata, when they are not directly provided with the sources, and the impact that they have on the results returned by our approach.

Metadata generation received much attention in the literature. According to [1], metadata relative to a data source are currently generated by crawlers, by professional metadata creators, or, finally, by source creators. Generating metadata by means of automatic crawlers has great advantages, such as low cost and high efficiency; however, in some cases, the quality of generated metadata could be poor. In this context, it could be extremely useful the support of several mechanisms for controlling the quality of metadata, as well as the aid of metadata professionals, such as cataloguers and indexers; these are people who have had a formal training and are efficient in using metadata. Generally, they produce high-quality metadata. However, it has been observed that, in some cases, even metadata generated by professionals or by source authors may have poor quality and might hamper, rather than aid, the usage of the corresponding sources. This happens because most authors have little previous knowledge on metadata creation [1].

As pointed out in [57], the widespread adoption of several mechanisms for controlling the quality of metadata witnesses a strong awareness of the importance of having high-quality metadata at disposal. However, despite the relevance and the impact of metadata quality are universally recognized in the literature, there is no agreement yet on what metadata quality actually means. This implies, among the other things, the impossibility of defining systematic approaches to its automatic measurement and enhancement [67]. Metadata quality assurance should be verified simultaneously to metadata creation [56]. Indeed, a poor quality of metadata negatively affects the performance of systems using them and the overall user satisfaction. Quality assurance procedures are generally complemented by manual quality review and, if necessary, by the assistance of the technical staff during the process of metadata creation. Other mechanisms, such as metadata creation guidelines (sometimes embedded into the metadata creation system) and metadata generation tools, are on the rise.

The great relevance given to the metadata quality improvement is observed in the study presented in [29]. Here, the authors introduce a quality measure and analyze the metadata quality in the Europeana context over the years. They observe that the metadata quality improves not only in new collections but also in the same collection over the years.

As pointed out in [57], in the metadata generation process, accuracy and consistency are prioritized over completeness, whereas the semantics of metadata elements is perceived to be less important. In principle, this might be an issue for our approach, since it strongly relies on semantics. The authors of [57] also point out that semantic overlaps and ambiguities are by far the two most critical factors. Actually, as our approach exploits thesauruses, string, and semantic similarities to relate keywords, these negative factors are significantly mitigated.

After this important discussion about the metadata of the involved sources, we can start our discussion about the derivation of interschema properties. We recall that, in the current big data scenario, any interschema property extraction strategy must be lightweight. For this reason, in our effort to define a new approach for this task, we avoided highly complex choices, such as the fixpoint computation characterizing DIKE [55, 54] and XIKE [19], or the clustering-based computation characterizing MOMIS [8], or, again, the wide range of parameter computation characterizing Cupid [41]. These choices, as well as most of the other ones present in the past approaches proposed for reconciling and integrating structured and semi-structured data sources (e.g., the construction of a data warehouse) [61, 10], would certainly return very accurate results. However, their speed is incompatible with the one required in many current applications, which must allow the derivation of semantic relationships "on-the-fly" from a very high number of data sources, most of which are unstructured, i.e., in a format not considered by classic approaches. As a consequence, our strategy must necessarily privilege quickness over accuracy even if, clearly, accuracy must be high. In Section 6, we will see if, and how, this issue has been addressed.

Our strategy consists of two phases; the former computes the semantic similarity degree of each pair of objects stored in the metadata of the involved sources. The latter derives semantic relationships between the same objects starting from the results returned by the former.

5.1 Semantic similarity degree computation

Our approach to semantic similarity degree computation consists of three steps, namely:

- basic similarity computation;
- standard similarity computation;
- refined similarity computation.

In the next subsections, we illustrate these three steps in detail.

5.1.1 Basic similarity computation

Basic similarities consider only lexicon (determined with the support of suitable thesauruses, such as BabelNet [50] and WordNet [48], and string similarity metrics, such as N-Grams [32]), and object types.

Let D_1 and D_2 be two sources, let $\mathcal{M}_1$ and $\mathcal{M}_2$ be the corresponding metadata, let $x_1 \in Obj_1$ and $x_2 \in Obj_2$ be two objects belonging to $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively. The basic similarity degree $bs(x_1, x_2)$ between x_1 and x_2 can be computed as:

$$bs(x_1, x_2) = \omega \cdot \sigma_L(x_1, x_2) + (1 - \omega) \cdot \sigma_T(x_1, x_2)$$

In other words, the basic similarity degree between x_1 and x_2 can be computed as a weighted mean of two components. The former, σ_L, returns their lexical similarity, whereas the latter, σ_T, specifies the similarity of their types. ω is a weight belonging to the real interval $[0, 1]$ and used to tune the importance of σ_L w.r.t. σ_T. We have experimentally set ω to 0.90.

σ_L can be directly detected from a thesaurus. In our experiments, we used WordNet in the first beat, because it provides the similarity degree between the two objects, and BabelNet, when WordNet did not provide any result. Since this last thesaurus does not return the similarity degree of two objects that it considers similar, we coupled BabelNet with a suitable string similarity metric (in particular, N-Grams). This last is applied to the objects and the corresponding lemmas returned by BabelNet; obtained results are, then, combined to compute the lacking similarity degree. Furthermore, in very specific application contexts, specialized thesauruses could be used.

σ_T is defined as follows:

$$\sigma_T = \begin{cases} 1 & \textit{if } (x_1 \in Cmp_1 \textit{ and } x_2 \in Cmp_2) \textit{ or } (x_1 \in Smp_1 \textit{ and } x_2 \in Smp_2) \textit{ or} \\ & \quad (x_1 \in Att_1 \textit{ and } x_2 \in Att_2) \\ 0.5 & \textit{if } (x_1 \in Cmp_1 \textit{ and } x_2 \in Smp_2) \textit{ or } (x_1 \in Smp_1 \textit{ and } x_2 \in Cmp_2) \textit{ or} \\ & \quad (x_1 \in Smp_1 \textit{ and } x_2 \in Att_2) \textit{ or } (x_1 \in Att_1 \textit{ and } x_2 \in Smp_2) \\ 0 & \textit{otherwise} \end{cases}$$

5.1.2 Standard similarity computation

Standard similarities take both basic similarities and the neighbors of the involved objects into account.

Let D_k be a source of the set DS of the sources of interest, let $\mathcal{M}_k = \langle N_k, A_k \rangle$ be the corresponding set of metadata, let Obj_k be the set of the objects of $\mathcal{M}_k$. The set $nbh(x)$ of the neighbors of an object $x \in Obj_k$ is defined as:

$$nbh(x) = \{y | y \in Obj_k, (n_x, n_y) \in A_k\}$$

Let D_1 and D_2 be two sources, let $\mathcal{M}_1$ and $\mathcal{M}_2$ be the corresponding sets of metadata, let $x_1 \in Obj_1$ and $x_2 \in Obj_2$ be two objects belonging to $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively. The standard similarity degree $ss(x_1, x_2)$ between x_1 and x_2 can be computed as follows:

- If both $nbh(x_1) = \emptyset$ and $nbh(x_2) = \emptyset$, then $ss(x_1, x_2) = bs(x_1, x_2)$ [5].
- If either $nbh(x_1) = \emptyset$ and $nbh(x_2) \neq \emptyset$ or $nbh(x_2) = \emptyset$ and $nbh(x_1) \neq \emptyset$, then $ss(x_1, x_2) = f_p \cdot bs(x_1, x_2)$. Here, f_p is a factor, whose possible values belong to the real interval $[0, 1]$, which

[5] For instance, this happens when both x_1 and x_2 are attributes; indeed, the nodes corresponding to attributes do not have outgoing arcs.

"penalizes" the value obtained for basic similarities. Indeed, these are the only similarities that we can compute and, therefore, we must base our standard similarity computation on them. However, we must consider that the sets of neighbors of x_1 and x_2 have different features, because one of them is empty and the other one is not empty, and this fact must be taken into account. We have experimentally set $f_p = 0.85$.

- In all the other cases, i.e., if $x_1 \in (Smp_1 \cup Cmp_1)$ and $x_2 \in (Smp_2 \cup Cmp_2)$, then $ss(x_1, x_2)$ can be computed as follows:

  1. $nbh(x_1)$ and $nbh(x_2)$ are determined.
  2. A bipartite graph, whose nodes are the ones of $nbh(x_1)$ and $nbh(x_2)$, is constructed.
  3. For each pair (p, q), such that $p \in nbh(x_1)$ and $q \in nbh(x_2)$, an arc is added in the bipartite graph; the weight of this arc is set to $bs(p, q)$.
  4. The maximum weight matching is computed on this bipartite graph. Let A_M be the set of the returned arcs. Then:

$$ss(x_1, x_2) = \begin{cases} \frac{2 \cdot \sum_{(p,q) \in A_M} bs(p,q)}{|nbh(x_1)| + |nbh(x_2)|} & \textit{if neither } D_1 \textit{ nor } D_2 \textit{ are unstructured} \\ \\ \frac{2 \cdot \sum_{(p,q) \in A_M} bs(p,q)}{2 \cdot min(|nbh(x_1)|, |nbh(x_2)|)} & \textit{otherwise} \end{cases}$$

In this formula, if neither D_1 nor D_2 are unstructured, $ss(x_1, x_2)$ returns the value of an objective function that takes into account how many nodes of $nbh(x_1)$ and $nbh(x_2)$ are linked by basic similarity relationships and how strong these relationships are. Furthermore, the objective function penalizes the presence of dangling nodes, i.e., nodes of $nbh(x_1)$ or $nbh(x_2)$ that do not participate to the maximum weight matching.

If D_1 and/or D_2 are unstructured, then it is necessary to consider that, even if our approach performed a "structuring" task, its final structure is limited, if compared with the rich structure characterizing the other kinds of source. As a consequence, the sets of neighbors of the nodes belonging to unstructured sources are generally much smaller than the ones characterizing the other kinds of source. Therefore, in this case, using the same objective function adopted when neither D_1 nor D_2 are unstructured would not take this important feature into account, and the overall result would be biased. To address this issue, if D_1 and/or D_2 are unstructured, in the denominator of $ss(x_1, x_2)$ we consider the minimum size between $|nbh(x_1)|$ and $|nbh(x_2)|$, clearly multiplied by 2 to indicate the maximum number of nodes that could be linked by a similarity relationship in this situation.

5.1.3 Refined similarity computation

Refined similarities are based on standard similarities (for simple and complex objects), basic similarities (for attributes) and object neighbors.

Let D_1 and D_2 be two sources, let $\mathcal{M}_1$ and $\mathcal{M}_2$ be the corresponding sets of metadata, let $x_1 \in Obj_1$ and $x_2 \in Obj_2$ be two objects belonging to $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively. The refined similarity degree $rs(x_1, x_2)$ between x_1 and x_2 can be computed as follows:

- If $nbh(x_1) = \emptyset$ and/or $nbh(x_2) = \emptyset$, then $rs(x_1, x_2) = ss(x_1, x_2)$.
- Otherwise, if $x_1 \in (Smp_1 \cup Cmp_1)$ and $x_2 \in (Smp_2 \cup Cmp_2)$, then $rs(x_1, x_2)$ is obtained by applying the same four steps described for $ss(x_1, x_2)$ with the only difference that, in Step 3, the weight of the arc (p, q), such that $p \in nbh(x_1)$ and $q \in nbh(x_2)$, is set to $ss(p, q)$, and no more to $bs(p, q)$. In other words, while standard similarity computation leverages basic similarities, refined similarity computation is based on standard similarities.

Clearly, from a theoretical point of view, it would be possible to perform other refinement steps. In this case, at the i^{th} refinement step, the similarities would be computed starting from the ones obtained at the $(i-1)^{th}$ step, by setting these last ones as the weights of the arcs of the bipartite graph. However, the advantages in accuracy that these further refinement steps could produce do not justify the computational costs introduced by them (see Section 6), especially in an agile and lightweight context, such as the one characterizing the big data scenario.

5.2 Semantic relationship detection

The derivation of semantic relationships among the objects of the sources of DS represents the second phase of our strategy. It takes the refined semantic similarities among the objects of DS as input. The semantic relationships that it can return are the following:

- *Synonymies*: A synonymy between two objects $x_1 \in Obj_1$ and $x_2 \in Obj_2$ exists if they have a high similarity degree, the same type (i.e., both of them are complex objects or simple objects or attributes) and (possibly) different names.
- *Type Conflicts*: A type conflict between two objects $x_1 \in Obj_1$ and $x_2 \in Obj_2$ exists if they have a high similarity degree but different types.
- *Overlappings*: An overlapping exists between two objects $x_1 \in Obj_1$ and $x_2 \in Obj_2$ if they have (possibly) different names, the same type and an intermediate similarity degree, in such a way that they can be considered neither synonymous nor distinct.
- *Homonymies*: A homonymy between two objects $x_1 \in Obj_1$ and $x_2 \in Obj_2$ exists if they have the same name and the same type but a low similarity degree.

Let D_1 and D_2 be two sources, let $\mathcal{M}_1$ and $\mathcal{M}_2$ be the corresponding sets of metadata, let $x_1 \in Obj_1$ and $x_2 \in Obj_2$ be two objects belonging to $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively. Finally, let $RefSim_{12}$ be the set of refined similarities involving the objects of Obj_1 and Obj_2.

First, our approach computes the maximum refined similarity degree rs_{max} present in $RefSim_{12}$. Then, it examines each similarity $\langle x_1, x_2, rs(x_1, x_2) \rangle$ registered in $RefSim_{12}$ and verifies if a semantic relationship exists between the corresponding objects as follows:

- If $(rs(x_1, x_2) \geq th_{Syn} \cdot rs_{max})$ and $(rs(x_1, x_2) \geq th_{min})$, which implies that the refined similarity degree between x_1 and x_2 is among the highest ones in $RefSim_{12}$ and, in any case, higher than or equal to a minimum threshold, then:

  - if x_1 and x_2 have the same type, it is possible to conclude that a synonymy exists between them;
  - if x_1 and x_2 have different types, it is possible to conclude that a type conflict exists between them.

- If $(rs(x_1, x_2) < th_{Syn} \cdot rs_{max})$ and $(rs(x_1, x_2) \geq th_{Ov} \cdot rs_{max})$ and $(rs(x_1, x_2) \geq th_{min})$, which implies that the refined similarity degree between x_1 and x_2 is higher than or equal to a minimum threshold, it is not among the highest ones in $RefSim_{12}$, but it is significant, then:
  - if x_1 and x_2 have the same type, it is possible to conclude that an overlapping exists between them.

- If $(rs(x_1, x_2) < th_{Hom} \cdot rs_{max})$ and $(rs(x_1, x_2) < th_{max})$, which implies that the refined similarity degree between x_1 and x_2 is among the lowest ones in $RefSim_{12}$ and, in any case, lower than a maximum threshold, then:
  - if x_1 and x_2 have the same name and the same type, it is possible to conclude that a homonymy exists between them.

Here, th_{Syn}, th_{min}, th_{Ov}, th_{Hom} and th_{max} have been experimentally set to 0.85, 0.50, 0.65, 0.25 and 0.15, respectively.

As pointed out in the Introduction, the knowledge of interschema properties is very relevant for several applications, for instance source integration, source querying, data warehouse and/or data lake construction, data analytics, and so forth. As an example, as far as source integration is concerned:

- If a synonymy exists between $x_1 \in Obj_1$ and $x_2 \in Obj_2$, then x_1 and x_2 must be merged in a unique object, when the integrated schema is constructed.
- If a homonymy exists between x_1 and x_2, then it is necessary to change the name of x_1 and/or x_2, when the integrated schema is constructed.
- If an overlapping exists between x_1 an x_2, then it is necessary to restructure the corresponding portion of network. Specifically, a node x_{12}, representing the "common part" of x_1 and x_2, is added to the network. Furthermore, each pair of arcs (x_1, x_T) and (x_2, x_T), starting from x_1 and x_2 and having the same target x_T, is substituted by a unique arc (x_{12}, x_T). Finally, an arc from x_1 to x_{12} and another arc from x_2 to x_{12} are added to the network.
- If a type conflict exists between x_1 and x_2, then it is necessary to change the type of x_1 and/or x_2 in such a way as to transform the type conflict into a synonymy. Then, it is necessary to handle this last relationship by applying the corresponding integration rule seen above.

The way of proceeding described above can be extended to the detection of hyponymies. In particular, the extension already proposed in [52] for structured and semi-structured data can be probably adapted to this scenario. We plan to investigate this issue in the future. Finally, an analogous way of proceeding can be performed when querying or other activities must be carried out on a set of sources of interest.

5.2.1 An example case

In this section, we provide an example of the behavior of our approach to the extraction of semantic relationships. To fully illustrate its potentialities, we derive these relationships between objects belonging to an unstructured source and a semi-structured one.

The unstructured source is a video. The corresponding keywords are reported in Table 1. Its "structured" representation, in our network-based model, obtained after the application of the approach described in Section 4, is reported in Figure 2. The semi-structured source is a JSON file whose structure is shown in Figure 3. Its representation in our network-based model is reported in Figure 4.

Keywords
$video, reuse, flower, easy, tips, plastic, simple, environment, pollution, garbage, wave, recycle, reduce, pollute, help,$ $natural\_environment, educational, green, environment\_awareness, bike, life, environmentalism, planet,$ $earth, climate, clime, save, nature, environmental, gardens, power, recycling, garden, protect, flowers,$ $eco, fine\_particle, o_3, atmospheric\_condition, ocean, metropolis, weather, spot, waving, aurora$

Table 1: Keywords of the unstructured source of our interest

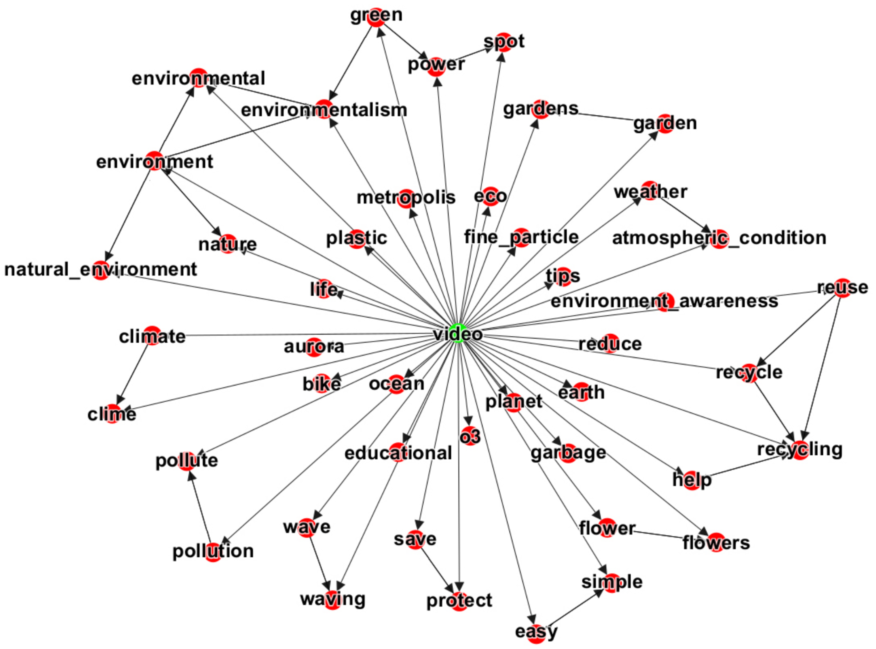


Figure 2: Representation, in our network-based model, of the unstructured source of our interest

By applying the first phase of our approach we obtained the refined semantic similarity degrees between all the possible pairs of nodes (n_U, n_S), such that n_U belongs to the unstructured source and n_S belongs to the semi-structured one. To give an idea of these similarity degrees, in Figure 5, we report their distribution in a semi-logarithmic scale. From the analysis of this figure, we can observe

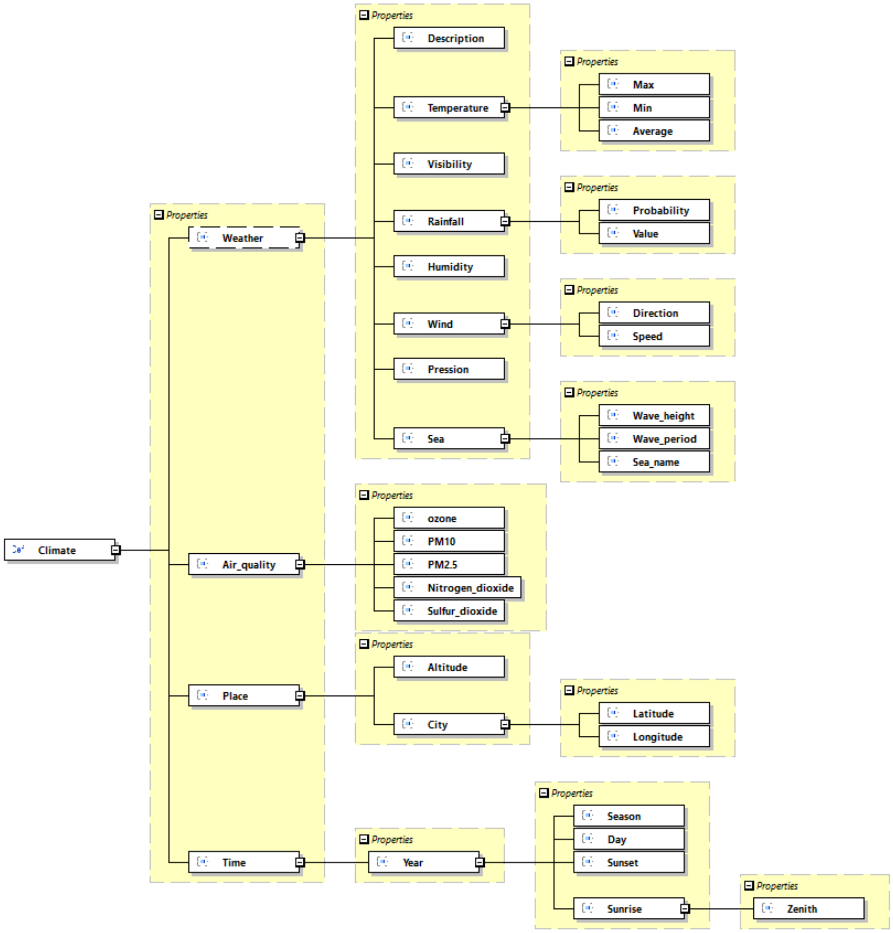


Figure 3: Structure of the JSON file associated with the semi-structured source of our interest

that a very few number of pairs have a significant similarity degree, which could make them eligible to be selected for synonymies, type conflicts and overlappings. At a first glance, this trend appeared correct and intuitive, even if this conclusion had to be confirmed or rejected by a much deeper analysis (see below).

By applying the second phase of our approach, we obtained the synonymies, the type conflicts and the overlappings reported in Tables 2 - 4. Instead, as for this pair of sources, we found no homonymies.

Semi-Structured Source Node	*Unstructured Source Node*
climate	*climate*
climate	*clime*

Table 2: Derived synonymies between objects of the two sources of interest

We asked a human expert to validate these results. At the end of this task, he reported the following considerations:

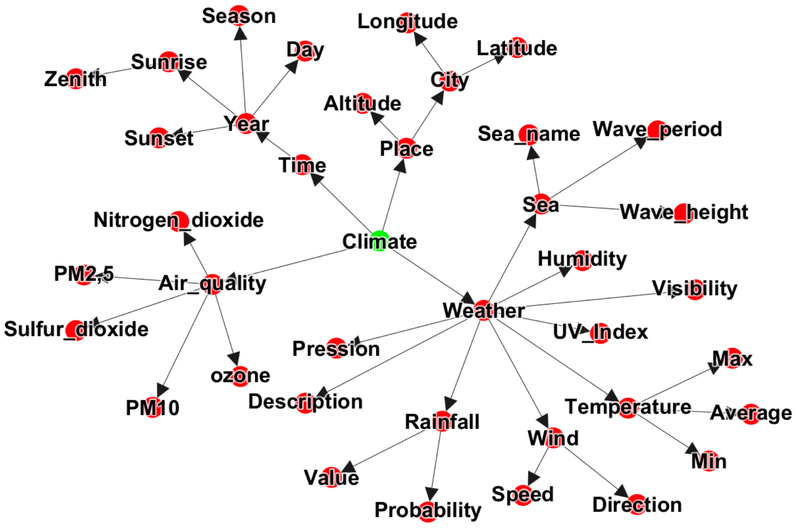


Figure 4: Representation, in our network-based model, of the semi-structured source of our interest

Semi-Structured Source Node	*Unstructured Source Node*
$pm10$	$fine\_particle$
$ozone$	o_3

Table 3: Derived type conflicts between objects of the two sources of interest

Semi-Structured Source Node	*Unstructured Source Node*
sea	*ocean*
city	*metropolis*
sunrise	*aurora*
place	*spot*
wind	*tips*
sulfur_dioxide	*garbage*
weather	*clime*

Table 4: Derived overlappings between objects of the two sources of interest

- The synonymies provided by our approach are correct. No further synonymy can be manually found in the two considered sources.

- The type conflicts provided by our approach are correct. No further type conflict can be manually found in the two sources.

- The overlappings provided by our approach are correct, except for the one linking "wind" and "tips", which actually represents two different concepts. A very interesting overlapping found by our approach is the one between "sulfur_dioxide" and "garbage", in that, even if they represent two seemingly different concepts, both of them denote harmful substances. Some further

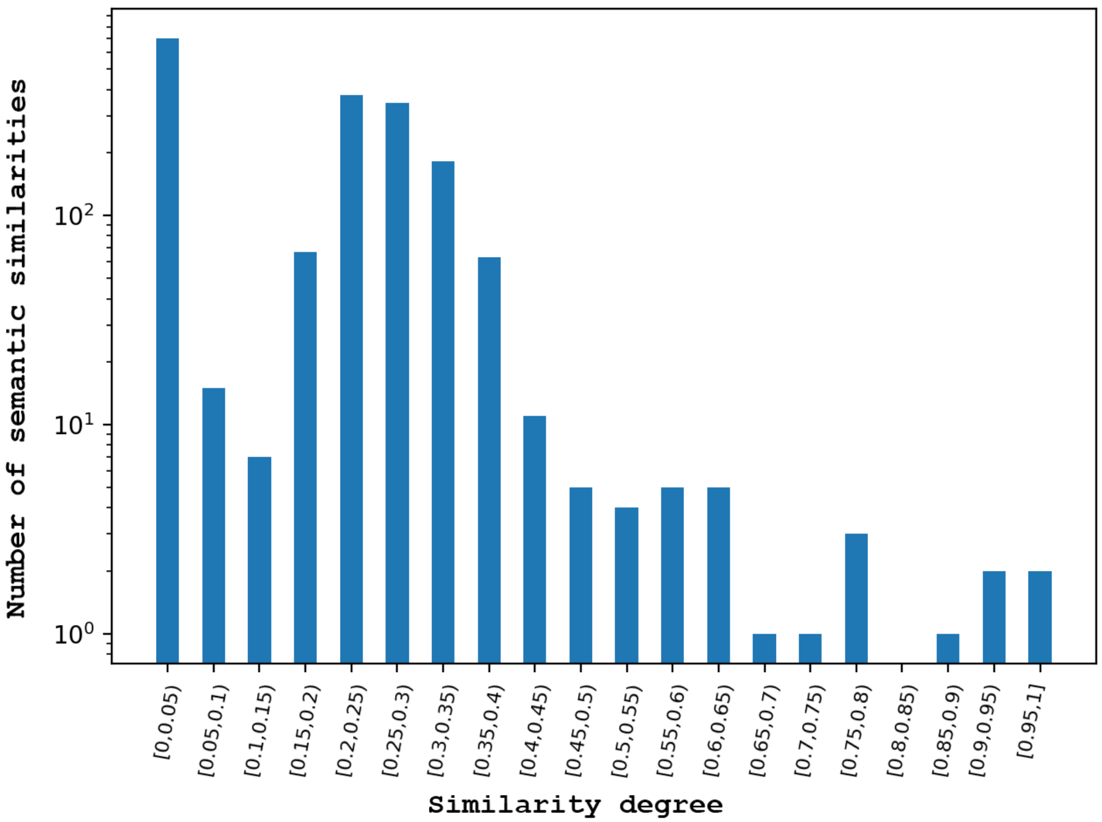


Figure 5: Distribution, in a semi-logarithmic scale, of the values of the the semantic similarity degrees of the objects belonging to the two sources of interest

overlappings could be manually found in the two sources into consideration (for instance, the one between “climate” and “environment”), even if they are semantically weak, and considering them as overlappings or as distinct concepts is subjective.

6 Experiments

Our test campaign had four main purposes, namely: *(i)* evaluating the performance of our interschema property derivation approach when applied to the scenario for which it was thought, *(ii)* evaluating the pros and the cons of this approach w.r.t. analogous ones thought for structured and semi-structured sources, *(iii)* evaluating its scalability, and *(iv)* evaluating the role of our approach for structuring unstructured sources. We describe these four experiments in the next subsections.

6.1 Overall performances of our approach

To perform our experiments, we constructed a set DS of data sources consisting of 2 structured sources, 4 semi-structured ones (2 of which were XML sources and 2 were JSON ones), and 4 unstructured ones (2 of which were books and 2 were videos). All these sources stored data about environment and pollution. To describe unstructured sources, we considered a list of keywords for each of them. These keywords were derived from Google Books, for books, and from YouTube, for videos. The interested reader can find the schemas, in case of structured and semi-structured sources, and the keywords, in case of unstructured sources, at the address `http://daisy.dii.univpm.it/dl/datasets/dl1`. The password to type is "`za.12&;lq74:#`". A summary of the size of these sources is reported in Table 5.

Data Source	*Size (order)*
Structured Sources	Gigabytes
Semi-structured Sources	Gigabytes (2 sources), Hundreds of Gigabytes (2 further sources)
Unstructured (books)	Megabytes
Unstructured (videos)	Gigabytes

Table 5: Size of the sources involved in the tests

It could appear that taking only 10 sources is excessively limited. However, we made this choice because we wanted to fully analyze the behavior and the performance of our approach and, as it will be clear below, this requires the human intervention for verifying obtained results. This intervention would have become much more difficult with a higher number of sources to examine. At the same time, our test set is fully scalable. As a consequence, an interested reader, starting from the data sources provided at the address `http://daisy.dii.univpm.it/dl/datasets/dl1`, can construct a data set with a much higher number of sources, if necessary.

For our experiments, we used a server equipped with an Intel I7 Dual Core 5500U processor and 16 GB of RAM with the Ubuntu 16.04.3 operating system. Clearly, the capabilities of this server were limited. However, they were adequate for the (small) data set DS we have chosen to use in our tests.

As the first task of our experiment, we represented the metadata of all the sources by means of the data model described in Section 3. Then, we applied the approach described in Section 4 to (at least partially) "structure" the unstructured sources of our test data set. Finally, we extracted semantic relationships existing between all the possible pairs of objects belonging to our test sources. After this, we asked the human expert to examine all the possible pairs of our test sources and to indicate us the semantic relationships that, in his opinion, existed among the corresponding objects.

At this point, we were able to evaluate the correctness and the completeness of our approach by measuring the classical parameters adopted in the literature for this purpose, i.e., Precision, Recall, F-Measure and Overall [68].

Precision is a measure of correctness. It is defined as:

$$Precision = \frac{|TP|}{|TP|+|FP|}$$

where TP are the true positives (i.e., semantic relationships detected by our approach and confirmed by the human expert), whereas FP are the false positives (i.e., semantic relationships proposed by our approach but not confirmed by our expert).

Recall is a measure of completeness. It is defined as:

$$Recall = \frac{|TP|}{|TP|+|FN|}$$

Property	*Precision*	*Recall*	*F-Measure*	*Overall*
Synonymies	0.82	0.87	0.84	0.68
Overlappings	0.77	0.69	0.73	0.48
Type Conflicts	0.78	0.73	0.75	0.52
Homonymies	0.95	0.92	0.93	0.87

Table 6: Precision, Recall, F-Measure and Overall of our approach

where FN are the false negatives (i.e., semantic relationships detected by the human expert that our system was unable to find).

F-Measure is the harmonic mean of Precision and Recall. It is defined as:

$$F\text{-}Measure = 2 \cdot \frac{Pecision \cdot Recall}{Precision + Recall}$$

Overall measures the post-match effort needed for adding false negatives and removing false positives from the set of matchings returned by the system to evaluate. It is defined as:

$$Overall = Recall \cdot (2 - \frac{1}{Precision})$$

Precision, Recall and F-Measure fall within the interval $[0, 1]$, whereas Overall ranges between $-\infty$ and 1; the higher Precision, Recall, F-Measure and Overall, the better the performance of the evaluated approach.

In Table 6, we report obtained results. From the analysis of this table, we can observe that, although our approach has been designed with the intent of privileging quickness and lightweightness over accuracy, for the reasons explained in the Introduction, its performance, in terms of correctness and completeness, is extremely satisfying.

We also point out that the values reported in Table 6 are those obtained by applying the threshold values reported in Section 5. These are the ones guaranteeing the best tradeoff between Precision and Recall and, consequently, the best values of F-Measure and Overall.

Interestingly, if, in a given application context, a user must privilege correctness (resp., completeness) over completeness (resp., correctness), it is sufficient to increase (resp., decrease) the values of th_{min} and to decrease (resp., increase) the values of th_{Ov} and th_{max}.

6.2 Evaluation of the pros and the cons of our approach

In order to provide a quantitative evaluation of the pros and the cons of our interschema property extraction approach w.r.t. the past ones thought for structured and semi-structured sources[6] [61, 10], we compared our approach with XIKE [19]. Indeed, in [19], XIKE was already compared with several other systems having the same purposes (namely, Autoplex, COMA, Cupid, LSD, GLUE, SemInt, Similarity Flooding) and it was shown that it obtained comparable or better results.

[6] Actually, to the best of our knowledge, no approach to uniformly extract interschema properties from structured, semi-structured and unstructured sources have been proposed in the past.

Application context	*Number of Schemas*	*Max depth*	*Average Number of nodes*	*Average Number of complex elements*
Biomedical Data	11	8	26	8
Project Management	3	4	40	7
Property Register	2	4	70	14
Industrial Companies	5	4	28	8
Universities	5	5	17	4
Airlines	2	4	13	4
Biological Data	5	8	327	60
Scientific Publications	2	6	18	9

Table 7: Characteristics of the sources adopted for evaluating our approach

First, we evaluated Precision, Recall, F-Measure and Overall of our approach and XIKE. Clearly, since this last system (as well as all the other ones mentioned above) did not handle unstructured data sources, we had to limit ourselves to consider only structured or semi-structured sources. Furthermore, as performed in [19], we limited our attention to synonymies and homonymies.

In a first experiment, we considered the same sources adopted in [19] for evaluating the performance of XIKE. In particular, we considered sources relative to Biomedical Data, Project Management, Property Register, Industrial Companies, Universities, Airlines, Biological Data and Scientific Publications. According to what reported in [19], Biomedical Schemas have been derived from several sites; among them we cite `http://www.biomediator.org`[7]. Project Management, Property Register and Industrial Companies Schemas have been derived from Italian Central Governmental Office (ICGO) sources and are shown at the address `http://www.mat.unical.it/terracina/tests.html`. Universities Schemas have been downloaded from the address `http://anhai.cs.uiuc.edu/archive/domains/courses.html`[8]. Airlines Schemas have been found in [58]; Biological Schemas have been downloaded from the addresses `http://smi-web.stanford.edu/projects/helix/pubs/ismb02/schemas/`[9], `http://www.cs.toronto.edu/db/clio/data/GeneX_RDB-s.xsd`[10] and `http://www.genome.ad.jp/kegg/soap/v3.0/KEGG.wsdl`. Finally, Scientific Publications Schemas have been supplied by the authors of [37].

We considered 35 sources whose characteristics are reported in Table 7. The minimum, the maximum and the average number of concepts of our sources were 12, 829 and 79, respectively.

A summary of the size of tested sources is shown in Table 8.

The number of synonymies (resp., homonymies) really present in these sources was 498 (resp, 66).

[7] Currently, this web address is no more available. However, the interested reader can find the corresponding source at the address `https://web.archive.org/web/20100412034606/http://www.biomediator.org/`

[8] Currently, this web address is no more available. However, the interested reader can find the corresponding source at the address `https://web.archive.org/web/20061212142107/http://anhai.cs.uiuc.edu/archive/domains/courses.html`

[9] Currently, this web address is no more available. However, the interested reader can find the corresponding source at the address `https://web.archive.org/web/20050314041246/http://smi-web.stanford.edu/projects/helix/pubs/ismb02/schemas/`

[10] Currently, this web address is no more available. However, the interested reader can find the corresponding source at the address `https://web.archive.org/web/20060718122245/http://www.cs.toronto.edu/db/clio/data/GeneX_RDB-s.xsd`

Data Source	*Size (order)*
Biomedical Data	Between Gigabytes and Hundreds of Gigabytes
ICGO Databases	Between Hundreds of Gigabytes and Terabytes
Universities Data	Megabytes
Airlines Data	Gigabytes
Biological Data	Terabytes and more
Scientific Publication Data	Hundreds of Gigabytes

Table 8: Size of the sources involved in the tests

System	*Precision*	*Recall*	*F-Measure*	*Overall*
XIKE (Synonymies)	0.92	0.90	0.91	0.82
XIKE (Homonymies)	0.87	0.95	0.91	0.81
Our approach (Synonymies)	0.84	0.87	0.85	0.70
Our approach (Homonymies)	0.79	0.92	0.85	0.68

Table 9: Precision, Recall, F-Measure and Overall of XIKE and our approach

The number of synonymies (resp., homonymies) returned by XIKE was 541 (resp, 76). Finally, the number of synonymies (resp., homonymies) returned by our system was 593 (resp., 84). By comparing real synonymies and homonymies with the ones returned by XIKE and our approach we computed Precision, Recall, F-Measure and Overall for these two systems. They are reported in Table 9.

From the analysis of this table we can observe that Precision, Recall, F-Measure and Overall are better in XIKE, even if those obtained by our approach are satisfying. This was expected because our approach has been designed to be lightweight and, therefore, it introduces some approximations. For instance, while XIKE considers the neighbors of many levels in the computation of the similarity degree of two objects, our approach considers only the neighbors of levels 1 and 2.

Until now, our experimental campaign highlighted the cons of our approach. To evidence and quantify the pros, we measured its response time and the one of XIKE when the number of involved concepts represented in the corresponding metadata to examine increases. Obtained results are reported in Figure 6.

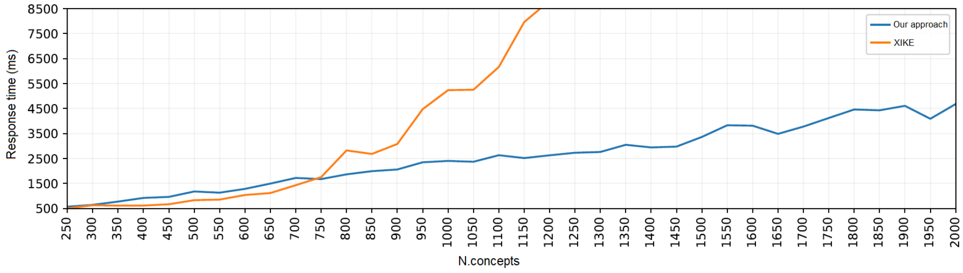


Figure 6: Computation time of XIKE and our approach against the number of concepts to process

From the analysis of this figure, it clearly emerges that, as for this aspect, our approach is much better than XIKE. Indeed, the difference in the computation time between it and XIKE is of various orders of magnitude and is such to make XIKE, and the other systems mentioned above, unsuitable to handle the number and the size of the data sources characterizing the current big data scenario.

With reference to this claim, we observe that, in this experiment, the response time is measured against the number of concepts in the source metaschema. As such, already a set of sources with 1500 concepts can be considered "large". Indeed, it would correspond, for instance, to a set of E/R schemas consisting of about 1500 entities or a set of XML Schemas defining about 1500 different element types.

Furthermore, in this analysis, we must not forget that XIKE and the approaches mentioned above are not capable of handling unstructured data, which represents the second (and, for many verses, most important) peculiarity of our approach.

6.3 A deeper investigation on the scalability of our approach

The previous experiment represents a first confirmation of the quickness and the scalability of our approach. In this section, we aim at finding a further confirmation of this trend by considering a much more numerous and articulated set of sources and by comparing the accuracy and the response time of our approach, of XIKE [19] and DIKE [53]. This last is one of the approaches of its generation showing the highest accuracy, as witnessed by the comparison tests described in [61].

Clearly, if we want to compare these three approaches, the only way of proceeding is to consider structured sources because they are the only ones handled by DIKE. In particular, we considered the database schemas of Italian Central Government Offices (hereafter, ICGO). They include about 300 databases that are heterogeneous both in the data model and languages (e.g., hierarchical, network, relational), as well as in their structure and complexity, ranging from simple databases, with schemas including few objects, to very complex databases [55]. Information about the size of these data sources is provided in Table 8.

Observe that our approach, XIKE and DIKE are all based on graphs and on the computation of similarities of the neighbors of the involved objects. However, DIKE was thought for relatively small structured databases. As a consequence, when it computes the similarity of two objects belonging to different sources, it considers the similarity of their direct neighbors, the one of the neighbors of their direct neighbors, and so forth, until it terminates a fixpoint computation. In the worst case, the number of iterations of the fixpoint computation could be equal to the number of concepts of one of the involved sources. Clearly, performing such a high number of iterations allows DIKE to return very accurate results, but the required computation time is generally very high not only from the worst case computational complexity viewpoint, but also from the real computation time point of view. In XIKE, the possible number and dimension of data sources is higher than DIKE and they can be both structured and semi-structured. As a consequence, there is the need to limit the number of iterations of the fixpoint computation. For this reason, the concept of severity level is introduced. In particular, a user can choose a severity level u between 1 and n and the fixpoint computation is not completed but terminates after u iterations. The higher u the more accurate and slower XIKE. Our approach privileges lightweightness over accuracy for the reasons explained above. As a consequence, in this case, we limited the fixpoint computation to only 2 iterations. This could cause a reduction of accuracy but it is the only way to extend the approach of DIKE and XIKE also to a big data scenario.

Analogously to what happened in the previous section, in order to verify the theoretical conjectures

System	*Precision*	*Recall*	*F-Measure*	*Overall*
DIKE (Synonymies)	0.98	0.93	0.95	0.91
DIKE (Homonymies)	0.96	0.95	0.95	0.91
XIKE $u = 5$ (Synonymies)	0.96	0.91	0.93	0.87
XIKE $u = 5$ (Homonymies)	0.93	0.93	0.93	0.86
XIKE $u = 2$ (Synonymies)	0.84	0.86	0.85	0.70
XIKE $u = 2$ (Homonymies)	0.85	0.86	0.85	0.71
Our approach (Synonymies)	0.83	0.81	0.82	0.64
Our approach (Homonymies)	0.81	0.83	0.82	0.64

Table 10: Precision, Recall, F-Measure and Overall of DIKE, XIKE ($u = 5$, $u = 2$) and our approach

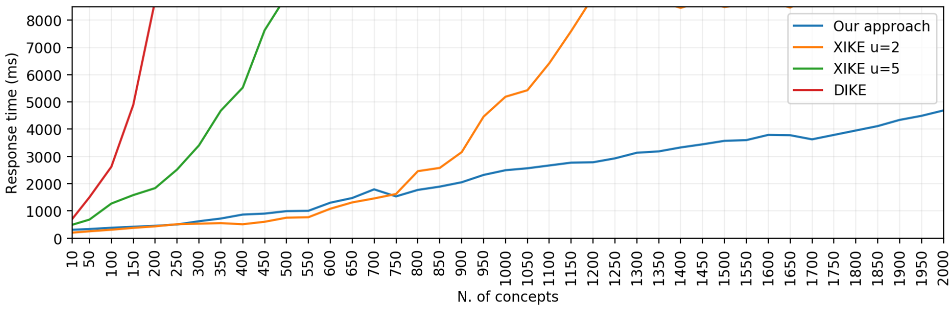


Figure 7: Computation time of DIKE, XIKE ($u = 5$ and $u = 2$) and our approach against the number of concepts to process

explained above, we applied our approach, DIKE and XIKE (with $u = 5$ and, then, with $u = 2$) to ICGO databases. The obtained results are reported in Table 10.

The results of this table confirm our conjectures. DIKE provides a higher Precision, Recall, F-Measure and Overall than XIKE which, in turn, provides better results than our approach. Finally, XIKE, with a severity level equal to 5, provides better results than XIKE with a severity level equal to 2. The former tend to be comparable with the ones of DIKE; the latter tend to be comparable with the ones of our approach. This is in line with the fact that, when u tends to 5 the fixpoint computation tends to be complete; instead, when $u = 2$, it is substituted by only three iterations.

In any case, we would like to remark that, analogously to what happened in the previous experiment, the results obtained by our approach are still acceptable.

After having verified our conjectures about accuracy, we analyzed the ones regarding computation time. In particular, the average computation time of DIKE, XIKE (with $u = 5$ and $u = 2$) and our approach is reported in Figure 7.

From the analysis of this figure, it is easy to observe that the lower performance in terms of accuracy of our approach is largely balanced by an increased performance in terms of computation time. In a big data context, this aspect is mandatory. As a matter of fact, Figure 7 shows that DIKE and XIKE (especially when the severity level is high), even if very accurate, could not be applied in a big data scenario.

Property	*Precision*	*Recall*	*F-Measure*	*Overall*
Synonymies	0.76	0.82	0.79	0.56
Overlappings	0.69	0.65	0.67	0.36
Type Conflicts	0.72	0.64	0.68	0.39
Homonymies	0.91	0.88	0.89	0.79

Table 11: Precision, Recall, F-Measure and Overall of our approach when a clustering-based technique for structuring unstructured sources is applied

6.4 Evaluation of the role of our approach for structuring unstructured sources

As previously pointed out, one of the main contributions of this paper is the approach for structuring unstructured sources. In the Introduction, we have seen that an important theoretical property of our approach (that distinguishes it from several possible alternative ones, like those based on ontologies) is its applicability to all possible scenarios, because it does not require a support knowledge, except for a (possibly generic) thesaurus, like BabelNet. In this section, we test its accuracy by comparing it with an alternative approach. For this purpose, we extended to unstructured data the clustering-based family of approaches defined for structured and semi-structured sources (see, for instance [4, 60]).

We performed this extension as follows: we considered the keywords associated with an unstructured source and used WordNet to derive a semantic distance coefficient for each pair of keywords. Then, we applied a clustering algorithm (specifically, Expectation Maximization [24]) to group keywords into homogeneous clusters. In this way, we obtained a possible structure for unstructured sources. This structure is in line with what was done in the past for the clustering-based family of approaches, when they were applied on structured and semi-structured sources. This way of proceeding gave us the possibility to still apply the interschema property extraction approach defined in Section 5. In this case, we assumed that, given a keyword, the corresponding neighborhood consisted of the other keywords of its clusters.

We performed the same experiment described in Section 6.1 on the same sources. The only difference was the substitution of our approach for structuring unstructured sources with the clustering-based approach outlined above. The obtained results are shown in Table 11. Clearly, the differences between the performance reported in Tables 6 and 11 were due exclusively to the merits or demerits of our approach for structuring unstructured sources. From the analysis of this table we can observe that our approach presents a better performance than the corresponding clustering-based one described above. The differences are evident even if not extremely marked. For instance, we can observe a gain in Precision (resp., Recall, F-Measure, Overall) ranging from 4% (resp., 4%, 4%, 9%) to 10% (resp., 12%, 10%, 25%).

The results of this experiment, coupled with the theoretical analysis performed in the Introduction and mentioned above, allow us to conclude that our approach for structuring unstructured data is really capable of satisfying the requirements for which it was defined.

6.5 Effectiveness vs Efficiency

In any context characterized by a huge amount of data, such as those of interest to most current computer applications, efficiency plays a fundamental role. In fact, in these contexts, effectiveness (defined in terms of accuracy, precision, recall, etc.) is certainly an aspect to be taken into account, but it is not the only one and, in some cases, it may not be the main one. Indeed, if a high level of effectiveness can be achieved only at the price of adopting methods computationally incapable of handling huge data, then it is necessary to resort to approaches that, while preserving an acceptable level of effectiveness, are able to guarantee a computation time compatible with the huge amount of data to process. From what we have seen in the previous subsections, our approach falls exactly in this case. In fact, it may be extremely useful in all those cases in which it is necessary to obtain interschema properties, extracted from huge amounts of data, to be used in other applications, such as querying, integration, data lake and data warehouse construction, knowledge extraction, etc. In all these cases, although our approach is not paramount as far as effectiveness is concerned, it continues to return acceptable results and is able to complete its tasks. By contrast, the approaches of the previous generations examined above, which can give better results in terms of effectiveness, are not able to complete their tasks in a reasonable amount of time.

In the scenario described above, our approach presents another interesting feature as it is able to extract interschema properties from unstructured data. In this feature, it differs from the ones presented in the past. Therefore, it is extremely interesting to investigate the effectiveness/efficiency of our approach with regard to this kind of data source. In fact, all the experiments proposed above have shown that our approach is the only one, among those analyzed, able to operate with the sizes characterizing the current data sources. On the other side, a great number of these sources are unstructured. Therefore, analyzing the efficiency and effectiveness of our approach when it works with huge unstructured sources is compulsory.

In this analysis, there are two important points to consider. The first concerns the fact that our approach assumes that the keywords representing each unstructured source are already known. If these keywords were unknown, it would be necessary to extract them. In this case, if the extraction task requires an excessive effort, for instance of some orders of magnitude higher than the subsequent extraction of interschema properties, our approach would become inefficient, and therefore not usable, in all those cases in which the keywords of the unstructured sources are not known a priori. The second point concerns the performance of our approach in terms of effectiveness, compared to a naive approach that considers only the basic similarities between keywords (see Section 5.1.1). Indeed, this last approach would presumably be more efficient than ours.

To address both these points we conducted the following experiment. We selected four popular approaches to text/information extraction, namely RAKE (Rapid Automatic Keyword Extraction) [62], LDA (Latent Dirichlet Allocation) [12], YAKE! (Yet Another Keyword Extractor) [15] and TopicRank [13], and applied them to the unstructured data sources used in the experiments in Section 6.1. Each of these approaches returned its own set of keywords for each source. Let $\mathcal{K}^R$ (resp., $\mathcal{K}^L$, $\mathcal{K}^Y$ and $\mathcal{K}^T$) be the set of the sets of keywords returned by RAKE (resp., LDA, YAKE! and TopicRank) when applied to the unstructured sources considered in our tests. We applied our interschema property extraction approach, as well as the naive one based only on basic similarities, on the sets of the keywords of $\mathcal{K}^R$

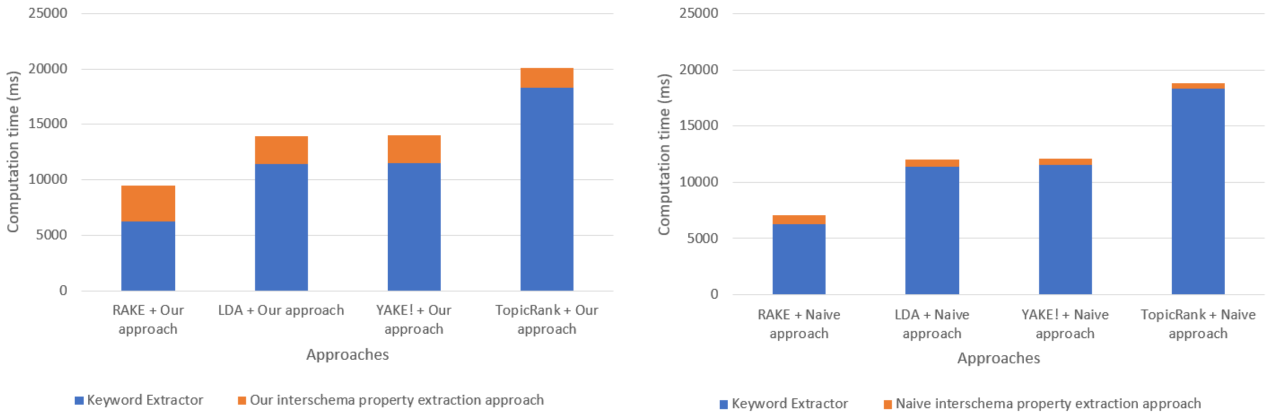


Figure 8: Computation time of RAKE, LDA, YAKE! and TopicRank coupled with our interschema property extraction approach and a naive one considering only basic similarities

(resp., $\mathcal{K}^L$, $\mathcal{K}^Y$ and $\mathcal{K}^T$). The computation times characterizing the eight overall approaches under consideration are shown in Figure 8, while the approaches' average Precision, Recall, F-Measure and Overall are shown in Table 12.

In our opinion, the results reported in Figure 8 and Table 12 are very important and encouraging. In fact, they tell us that, in case of unstructured sources without associated keywords, the keyword computation requires a longer time, but of a comparable order of magnitude, than the interschema property extraction task. Therefore, the possible preliminary detection of the keywords does not change the conclusions emerged from the analysis of Figures 6 and 7, i.e., that our approach is the only one that can be adopted in presence of huge data sources. At the same time, the adoption of our approach, which, as far as the examination of neighborhoods is concerned, is a compromise between DIKE and XIKE (which consider all possible neighborhoods) and the naive approach (which considers only the immediate neighborhoods), guarantees an effectiveness certainly lesser than the one of DIKE and XIKE, but much greater than the one of the naive approach.

Therefore, our approach appears to be the best compromise between the ones of the past generation, having a very high effectiveness but an unacceptable efficiency, and a naive one, having a slightly higher efficiency but a much lower effectiveness than our approach.

7 Conclusion

In this paper, we have presented an approach to uniformly derive interschema properties from structured, semi-structured and unstructured data sources. Initially, we have observed that, in the current big data scenario, where more than 80% of available resources are unstructured, the past approaches for the extraction of interschema properties (operating on structured and/or semi-structured sources) are not adequate. Furthermore, they privilege accuracy at detriment of response time, which make them unsuitable for scenarios, such as data lakes, where the number of sources to analyze is huge.

We have argued that a new approach to perform this task should be characterized by two pe-

Property	*Precision*	*Recall*	*F-Measure*	*Overall*
Synonymies (RAKE + our approach)	0.80	0.83	0.82	0.62
Overlappings (RAKE + our approach)	0.74	0.65	0.69	0.42
Type Conflicts (RAKE + our approach)	0.75	0.71	0.73	0.47
Homonymies (RAKE + our approach)	0.92	0.89	0.91	0.81
Synonymies (RAKE + naive approach)	0.68	0.70	0.69	0.37
Overlappings (RAKE + naive approach)	0.63	0.62	0.63	0.26
Type Conflicts (RAKE + naive approach)	0.63	0.59	0.61	0.24
Homonymies (RAKE + naive approach)	0.81	0.77	0.79	0.59
Synonymies (LDA + our approach)	0.81	0.88	0.84	0.67
Overlappings (LDA + our approach)	0.78	0.68	0.73	0.49
Type Conflicts (LDA + our approach)	0.77	0.74	0.76	0.52
Homonymies (LDA + our approach)	0.96	0.90	0.93	0.86
Synonymies (LDA + naive approach)	0.68	0.75	0.71	0.40
Overlappings (LDA + naive approach)	0.65	0.57	0.61	0.26
Type Conflicts (LDA + naive approach)	0.66	0.63	0.65	0.31
Homonymies (LDA + naive approach)	0.84	0.77	0.80	0.62
Synonymies (YAKE! + our approach)	0.83	0.85	0.84	0.68
Overlappings (YAKE! + our approach)	0.76	0.70	0.73	0.48
Type Conflicts (YAKE! + our approach)	0.80	0.71	0.75	0.53
Homonymies (YAKE! + our approach)	0.92	0.90	0.91	0.82
Synonymies (YAKE! + naive approach)	0.70	0.74	0.72	0.42
Overlappings (YAKE! + naive approach)	0.64	0.57	0.60	0.25
Type Conflicts (YAKE! + naive approach)	0.67	0.58	0.62	0.29
Homonymies (YAKE! + naive approach)	0.78	0.80	0.79	0.57
Synonymies (TopicRank + our approach)	0.84	0.89	0.86	0.72
Overlappings (TopicRank + our approach)	0.79	0.70	0.74	0.51
Type Conflicts (TopicRank + our approach)	0.79	0.74	0.76	0.54
Homonymies (TopicRank + our approach)	0.95	0.94	0.95	0.89
Synonymies (TopicRank + naive approach)	0.71	0.76	0.73	0.45
Overlappings (TopicRank + naive approach)	0.67	0.59	0.63	0.30
Type Conflicts (TopicRank + naive approach)	0.68	0.60	0.64	0.32
Homonymies (TopicRank + naive approach)	0.85	0.81	0.83	0.67

Table 12: Precision, Recall, F-Measure and Overall of RAKE, LDA, YAKE! and TopicRank coupled with our interschema property extraction approach and a naive one considering only basic similarities

culiarities, namely: *(i)* the capability of handling unstructured sources; *(ii)* the lightweightness. We showed that our approach has both these features and, in spite of its lightweightness, the accuracy it can reach is surely acceptable.

This paper is not to be intended as an ending point. Instead, it could be the starting point of a new generation of approaches conceived to address the major issues, typical of information system research, in the new big data scenario. For instance, we plan to define an approach to manage the flexible and lightweight extraction of complex knowledge patterns involving concepts that belong to structured, semi-structured and unstructured sources, as well as a flexible and lightweight approach for the extraction of thematic views from data lake sources.

Acknowledgments

This work was partially supported by: *(i)* the Italian Ministry for Economic Development (MISE) under the project “Smarter Solutions in the Big Data World”, funded within the call “HORIZON2020” PON I&C 2014-2020 (CUP B28I17000250008), and *(ii)* the Department of Information Engineering at the Polytechnic University of Marche under the project “A network-based approach to uniformly extract knowledge and support decision making in heterogeneous application contexts” (RSAB 2018).

References

[1] B.M. Albassuny. Automatic metadata generation applications: a survey study. *International Journal of Metadata, Semantics and Ontologies*, 3(4):260–282, 2008.

[2] Z. Aleksovski, M.C.A. Klein, W.T. Kate, and F. van Harmelen. Matching Unstructured Vocabularies Using a Background Ontology. In *Proc. of the International Conference on Knowledge Engineering and Knowledge Management (EKAW’06)*, pages 182–197, Prague, Czech Republic, 2006. Lecture Notes in Computer Science. Springer.

[3] A. Algergawy, E. Schallehn, and G. Saake. Improving XML schema matching performance using Pr. *Data & Knowledge Engineering*, 68(8):728–747, 2009. Elsevier.

[4] S. P. Algur and P. Bhat. Web Video Object Mining: Expectation Maximization and Density Based Clustering of Web Video Metadata Objects. *International Journal of Information Engineering and Electronic Business*, 8(1):69, 2016. Modern Education and Computer Science Press.

[5] A. Alserafi, A. Abello, O. Romero, and T. Calders. Towards information profiling: data lake content metadata management. In *Proc. of the International Conference on Data Mining Workshops (ICDMW’16)*, pages 178–185, Barcelona, Spain, 2016. IEEE.

[6] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* 1999. Addison Wesley Longman.

[7] S. Bergamaschi, S. Castano, and M. Vincini. Semantic integration of semistructured and structured data sources. *SIGMOD Record*, 28(1):54–59, 1999.

[8] S. Bergamaschi, S. Castano, M. Vincini, and D. Beneventano. Semantic integration and query of heterogeneous information sources. *Data & Knowledge Engineering*, 36(3):215–249, 2001.

[9] J. Bernabé-Moreno, A. Tejeda-Lorente, C. Porcel-Gallego, and E. Herrera-Viedma. Leveraging Localized Social Media Insights for Industry Early Warning Systems. *International Journal of Information Technology & Decision Making*, 17(01):357–385, 2018. World Scientific.

[10] P.A. Bernstein, J. Madhavan, and E. Rahm. Generic Schema Matching, Ten Years Later. *Proceedings of the VLDB Endowment*, 4(11):695–701, 2011.

[11] L. Bing, S. Jiang, W. Lam, Y. Zhang, and S. Jameel. Adaptive Concept Resolution for document representation and its applications in text mining. *Knowledge Based Systems*, 74:1–13, 2015.

[12] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003. Microtone Publishing.

[13] A. Bougouin, F. Boudin, and B. Daille. Topicrank: Graph-based topic ranking for keyphrase extraction. In *Proc. of the International Joint Conference on Natural Language Processing (IJCNLP’13)*, pages 543–551, Nagoya, Japan, 2013. Asian Federation of Natural Language Processing.

[14] A. Boukottaya and C. Vanoirbeek. Schema matching for transforming structured documents. In *Proc. of the ACM Symposium on Document Engineering (DocEng’05)*, pages 101–110, Bristol, United Kingdom, 2005. ACM.

[15] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289, 2020. Elsevier.

[16] S. Castano, V. De Antonellis, and S. De Capitani di Vimercati. Global viewing of heterogeneous data sources. *IEEE Transactions on Data and Knowledge Engineering*, 13(2):277–297, 2001.

[17] J. Chen, N. Zhong, and J. Feng. Developing a Provenance Warehouse for the Systematic Brain Informatics Study. *International Journal of Information Technology & Decision Making*, 16(06):1581–1609, 2017. World Scientific.

[18] A. Corbellini, C. Mateos, A. Zunino, D. Godoy, and S.N. Schiaffino. Persisting big-data: The NoSQL landscape. *Information Systems*, 63:1–23, 2017. Elsevier.

[19] P. De Meo, G. Quattrone, G. Terracina, and D. Ursino. Integration of XML Schemas at various "severity" levels. *Information Systems*, 31(6):397–434, 2006.

[20] R. dos Santos Mello, S. Castano, and C.A. Heuser. A method for the unification of XML schemata. *Information & Software Technology*, 44(4):241–249, 2002. Elsevier.

[21] H. Elmeleegy, M. Ouzzani, and A.K. Elmagarmid. Usage-Based Schema Matching. In *Proc. of the International Conference on Data Engineering (ICDE'08)*, pages 20–29, Cancún, México, 2008. IEEE.

[22] F. Feng and W.B. Croft. Probabilistic techniques for phrase extraction. *Information Processing & Management*, 37(2):199–220, 2001.

[23] E. Gabrilovich and S. Markovitch. Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI'07)*, pages 1606–1611, Hyderabad, India, 2007.

[24] J. Han and M. Kamber. *Data Mining: Concepts and Techniques - Second Edition.* Morgan Kaufmann notes, 2006.

[25] S.M. Harding, W.B. Croft, and C. Weir. Probabilistic retrieval of ocr degraded text using n-grams. In *Proc. of the International Conference on Theory and Practice of Digital Libraries (ECDL'97)*, pages 345–359, Pisa, Italy, 1997. Springer.

[26] N.Q.V. Hung, N.T. Tam, V.T. Chau, T.K. Wijaya, Z. Miklós, K. Aberer, A. Gal, and M. Weidlich. SMART: A tool for analyzing and reconciling schema matching networks. In *Proc. of the International Conference on Data Engineering (ICDE'15)*, pages 1488–1491, Seoul, South Korea, 2015. IEEE.

[27] S. Jain and S. Tanwani. Schema matching technique for heterogeneous web database. In *Proc. of the International Conference on Reliability (ICRITO'15)*, pages 1–6, Noida, India, 2015. IEEE.

[28] S. Jiang, L. Bing, B. Sun, Y. Zhang, and W. Lam. Ontology enhancement and concept granularity learning: keeping yourself current and adaptive. In *Proc. of the International Conference on Knowledge Discovery and Data Mining (KDD'11)*, pages 1244–1252, San Diego, CA, USA, 2011. ACM.

[29] S. Kapidakis. Rating quality in metadata harvesting. In *Proc. of the International Conference on PErvasive Technologies Related to Assistive Environments (PETRA'15)*, pages 65:1–65:8, New York, NY, USA, 2015. ACM.

[30] H. Kim, P. Howland, and H. Park. Dimension Reduction in Text Classification with Support Vector Machines. *Journal of Machine Learning Research*, 6:37–53, 2005.

[31] H.K. Kim, H. Kim, and S. Cho. Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. *Neurocomputing*, 266:336–352, 2017.

[32] G. Kondrak. N-gram similarity and distance. In *String Processing and Information Retrieval*, pages 115–126, 2005. Springer.

[33] G. Kou, X. Chao, Y. Peng, F.E. Alsaadi, and E. Herrera-Viedma. Machine learning methods for systemic risk analysis in financial sectors. *Technological and Economic Development of Economy*, 25(5):716–742, 2019.

[34] G. Kou, Y. Lu, Y. Peng, and Y. Shi. Evaluation of classification algorithms using MCDM and rank correlation. *International Journal of Information Technology & Decision Making*, 11(01):197–225, 2012. World Scientific.

[35] G. Kou, Y. Peng, and G. Wang. Evaluation of clustering algorithms for financial risk analysis using MCDM methods. *Information Sciences*, 275:1–12, 2014. Elsevier.

[36] Q.V. Le and T. Mikolov. Distributed Representations of Sentences and Documents. In *Proc. of the International Conference on Machine Learning (ICML'14)*, pages 1188–1196, Beijing, China, 2014. JMLR.org.

[37] M.L. Lee, L.H. Yang, W. Hsu, and X. Yang. XClust: clustering XML schemas for effective integration. In *Proc. of the ACM International Conference on Information and Knowledge Management (CIKM 2002)*, pages 292–299, McLean, Virginia, USA, 2002. ACM Press.

[38] T. Li, G. Kou, Y. Peng, and Y. Shi. Classifying with adaptive hyper-spheres: An incremental classifier based on competitive learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pages 1–12, 2017. IEEE.

[39] X. Li, Y. Tian, F. Smarandache, and R. Alex. An extension collaborative innovation model in the context of big data. *International Journal of Information Technology & Decision Making*, 14(01):69–91, 2015. World Scientific.

[40] C. Lin, G. Li, Z. Shan, and Y. Shi. Thinking and Modeling for Big Data from the Perspective of the I Ching. *International Journal of Information Technology & Decision Making*, 16(06):1451–1463, 2017. World Scientific.

[41] J. Madhavan, P.A. Bernstein, and E. Rahm. Generic schema matching with Cupid. In *Proc. of the International Conference on Very Large Data Bases (VLDB 2001)*, pages 49–58, Rome, Italy, 2001. Morgan Kaufmann.

[42] B. Malysiak-Mrozek, M. Stabla, and D. Mrozek. Soft and Declarative Fishing of Information in Big Data Lake. *IEEE Transactions on Fuzzy Systems*, 26(5):2732–2747, 2018. IEEE.

[43] C.D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval.* 2008. Cambridge University Press Cambridge.

[44] J. Martinez-Gil and J.F. Aldana-Montes. Semantic similarity measurement using historical google search patterns. *Information Systems Frontiers*, 15(3):399–410, 2013. Springer.

[45] O. Mehdi, H. Ibrahim, and L. Affendey. An approach for instance based schema matching with Google similarity and regular expression. *International Arab Journal of Information Technology*, 14(5):755–763, 2017.

[46] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[47] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. In *Proc. of the International Conference on Advances in Neural Information Processing Systems (NIPS'13)*, pages 3111–3119, Lake Tahoe, NV, USA, 2013.

[48] A.G. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[49] A. Nandi and P.A. Bernstein. HAMSTER: Using Search Clicklogs for Schema and Taxonomy Matching. *Proceedings of the VLDB Endowment*, 2(1):181–192, 2009.

[50] R. Navigli and S.P. Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012. Elsevier.

[51] Q.V.H. Nguyen, T.T. Nguyen, V.T. Chau, T.K. Wijaya, Z. Miklos, K. Aberer, A. Gal, and M. Weidlich. SMART: A tool for analyzing and reconciling schema matching networks. In *Proc. of the International Conference on Data Engineering (ICDE'15)*, pages 1488–1491, Seoul, Korea, 2015. IEEE.

[52] L. Palopoli, D. Rosaci, G. Terracina, and D. Ursino. A graph-based approach for extracting terminological properties from information sources with heterogeneous formats. *Knowledge and Information Systems*, 8(4):462–497, 2005.

[53] L. Palopoli, D. Saccà, G. Terracina, and D. Ursino. Uniform techniques for deriving similarities of objects and subschemes in heterogeneous databases. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):271–294, 2003.

[54] L. Palopoli, G. Terracina, and D. Ursino. DIKE: a system supporting the semi-automatic construction of Cooperative Information Systems from heterogeneous databases. *Software Practice & Experience*, 33(9):847–884, 2003.

[55] L. Palopoli, G. Terracina, and D. Ursino. Experiences using DIKE, a system for supporting cooperative information system and data warehouse design. *Information Systems*, 28(7):835–865, 2003.

[56] J.-R. Park. Metadata quality in digital repositories: A survey of the current state of the art. *Cataloging & Classification Quarterly*, 47(3-4):213–228, 2009.

[57] J.-R. Park and Y. Tosaka. Metadata quality control in digital repositories and collections: Criteria, semantics, and mechanisms. *Cataloging & Classification Quarterly*, 48(8):696–715, 2010.

[58] K. Passi, L. Lane, S.K. Madria, B.C. Sakamuri, M.K. Mohania, and S.S. Bhowmick. A model for XML Schema integration. In *Proc. of the International Conference on E-Commerce and Web Technologies (EC-Web 2002)*, pages 193–202, Aix-en-Provence, France, 2002. Lecture Notes in Computer Science, Springer.

[59] M. Patella and P. Ciaccia. Approximate similarity search: A multi-faceted problem. *Journal of Discrete Algorithms*, 7(1):36–48, 2009. Elsevier.

[60] D.V. Prasad, S. Madhusudanan, and S. Jaganathan. uCLUST - A new algorithm for clustering unstructured data. *ARPN Journal of Engineering and Applied Sciences*, 10(5):2108–2117, 2015.

[61] E. Rahm and P.A. Bernstein. A survey of approaches to automatic schema matching. *VLDB Journal*, 10(4):334–350, 2001.

[62] S. Rose, D. Engel, N. Cramer, and W. Cowley. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20, 2010. Wiley, New York.

[63] M. Sahlgren and R. Cöster. Using Bag-of-Concepts to Improve the Performance of Support Vector Machines in Text Categorization. In *Proc. of the International Conference on Computational Linguistics (COLING'04)*, page 487, Geneva, Switzerland, 2004.

[64] M. Sahlgren and J. Karlgren. Automatic bilingual lexicon acquisition using random indexing of parallel corpora. *Natural Language Engineering*, 11(3):327–341, 2005.

[65] S.F. Sayeedunnissa, A.R. Hussain, and M.A. Hameed. Supervised Opinion Mining of Social Network Data Using a Bag-of-Words Approach on the Cloud. In *Proc. of the International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA'12)*, pages 299–309, Gwalior, India, 2012.

[66] J. Szymanski. Comparative Analysis of Text Representation Methods Using Classification. *Cybernetics and Systems*, 45(2):180–199, 2014.

[67] A. Tani, L. Candela, and D. Castelli. Dealing with metadata quality: The legacy of digital library efforts. *Information Processing & Management*, 49(6):1194–1205, 2013.

[68] C.J. Van Rijsbergen. *Information Retrieval*. 1979. Butterworth.

[69] F. Wang, Z. Wang, Z. Li, and J.R. Wen. Concept-based Short Text Classification and Ranking. In *Proc. of the International Conference on Information and Knowledge Management (CIKM'14)*, pages 1069–1078, Shangai, China, 2014. ACM.