Structural Information Retrieval in XML Documents: A Graph-based Approach

Imane Belahyane, Mouad Mammass, Hasna Abioui, Assmaa Moutaoukkil, Ali Idarrou
IRF-SIC Laboratory
Ibn Zohr University, Agadir, Morocco

Abstract—**Although retrieval engines are becoming more and more functional and efficient, they still have the drawback of not being able to locate the relevant documentary granularity, which results in ignoring the structural aspect. In the context of XML document, Information Retrieval Systems allow to return the user's documentary granules. Several studies have used graphs to represent XML documents. However, in the scope of this research, the semi-structured document's structure and that of a user's query can be seen as arborescences composed of a hierarchy of nested elements. By using graph theory, by calculating the structural proximity and especially the intersection between these two arborescences. The article presents a model for structural information retrieval based on graphs. A collection of multimedia documents are randomly extracted from INEX (Initiative for the Evaluation of XML Retrieval) 2010 to validate the approach. The first results shows the interest of such an approach.**

Keywords*—*Semi-structured document; XML document; largest common sub-graph; structural Information retrieval

I. Introduction

Due to the digital revolution in terms of documentary masses and their exhaustiveness as well as heterogeneity, it is becoming more difficult to access information. Saying so, this requires a lot of efforts to respond to a user's queries. In this respect, it is necessary to use an accurate tool to facilitate Information Retrieval (IR) in order to increase the performance of an Information Retrieval System (IRS). The research falls within the context of IR in Semi-Structured Documents (SSD) which are characterized by a flexibility. The latter overshadows the limitations of using structured documents (DBMS - Database Management System).

Indeed, the semi-structured document has organizational properties that facilitate their analysis. In literature, XML (eXtensible Markup Language) documents are often referred to as SSD. As a result, XML is becoming the preferred format for information exchange [1]. In the sense, it is used today by ERP (Enterprise Ressource Planning) suppliers, middleware editors and database providers. It is also used in e-commerce and libraries. Evidently, an XML documents are complex objects because they are heterogeneous in terms of type, size, format, content (image, text, video, etc), structure, etc. In fact, a structural IR facilitates access to the documentary granule (fine information) [2]. The paper is concerned with structural IR, while taking into account the structural information (relations between documentary components). For this purpose, it is required to compare the structure of the user's query with that of the document. The underlying idea of the approach to structural IR in XML documents is to:(1) Return the fragments (document parts) that are relevant to the user's query and (2) Return the ordered results by order of relevance.

Otherwise, XML documents are composed of nested tags, which enable their representation in the form of graphs. The work is mainly interested in the representation of document structures using graphs. Comparing two documents structurally is therefore the same as comparing the graphs that represent them [3]. The multi-structurality generated by graphs induces complex and multiple relations between two similar components of a document. It is therefore necessary to use a rich representation model in order to perform IR in documents with complex structures. Indeed, graphs are used in several works and several domains, and are efficient to model structured objects [4].

The paper is divided into the following sections. While Section 2 provides some approaches that have addressed the problem of IR in XML documents using graph theory, Section 3 introduces the approach on structural IR. Section 4 reports the first results. Finally, Section 5 presents some perspectives.

II. A Graph-based Structural IR Approaches: The State-of-Art

The XML documents exploitation requires the use of a rich representation. Indeed, graph theory is widely used to represent complex objects. This section presents a brief overview of the studies that used graphs in the field of structural IR, and then introducing the mathematical formalism of graph theory.

In [5], the authors proposed an IR approach based on the edit distance and the structural summary of the trees (representing SSD). Based on two scores, it propagates the first one (content score) at the tree level in order to extract the sub-trees containing the relevant leaves. The trees' relevant leaves' extraction, representing the XML document, are performed by an algorithm based on the TF-IDF technique [6]. The second score's calculation (structure score) of the previously extracted sub-trees is based on the edit distance algorithm [7]. The structure score adapts an optimal coverage strategy [8] based on the edit distance [7] through the heavy path [9]. In this work [10], IR in XML documents is based on a model that takes the DTDs set forming the document collection and merges them into an un-oriented graph. The approach is based on the edit distance [7] and the shortest path model inspired by [11]. The selection of relevant leaves is established according to the existing nodes in the query, from the leaves to the root. These relevant node paths are merged into a sub-tree.

In these works [12], the integration of the document structure enhances the performance of a graph matching system.

The method is based on the trees' abstract from which a set of vectors are extracted. Hence, the process consists of a structural analysis phase. To build the content of each tree, a linguistic analysis phase and the application of the weighting function are calculated. This is done by adapting the formulas TF-IDF and TF-IEF [2], in order to attribute to each node a weight reflecting its importance in the collection where it belongs.

For this work [13], it deals with the problem of finding the largest common sub graph applied to the compatibility graph. Indeed, a compatibility graph of two graphs is the one where its nodes belong to the nodes union of these two graphs, and are linked by compatible edges. This means that these edges are preserved. The suggested approach involves combining RT-composition with the PC (Constraint Programming) model defined in [14]. The latter is an optimization problem based on soft constraints and the generalization of hard constraints (CSP - Constraint Satisfaction Problems), by optimizing the objective function. The approach is also based on heuristics following this work [15], in order to eliminate sub-problems and look for maximal cliques in the initial compatibility graph. In fact, the TR-decomposition, introduced in [16] seeks to reduce the CSP to instantaneous problems for the detection of a clique of n size. TR-decomposition includes a triangulation process, where a given graph is transformed into a triangulated graph by adding edges. By definition, a graph is triangulated if each of its cycles of length is greater than or equal to 4 and has at least one chord. A chord of a cycle is any edge whose extremities are two consecutive nodes of the cycle. Triangulation is applied in order to take advantage of their maximal clique number limit. The search for the largest common sub graph is therefore equivalent to finding a maximal clique of the triangulated graph.

III. Graphs' Basics

A. Definition

A graph is a pair of sets $G = (V, E)$, in which V is the non-empty set of nodes, of cardinality $|V|$ and E is the non-empty set of edges between nodes, of cardinality $|E|$. A graph is mainly characterized by its ability to represent multiple relationships between the same nodes. While the tree is an undirected, acyclic and a connected graph, the arborescence is totally the opposite. It has a tree structure when ignoring the direction of the arcs.

B. Graphs' Inclusion

Inclusion is an order relation between graphs. Indeed, inclusion is designed to judge a sub-graph's belonging to a given graph. G_1 Graph is included in G_2 graph, if all the nodes and edges connecting the nodes of G_1 graph belong to G_2 graph.

$$G_1 \subset G_2 \quad \Longleftrightarrow V_{(G_1)} \subset V_{(G_2)} \quad and \quad E_{(G_1)} \subset E_{(G_2)}$$

The inclusion of graphs has been used in specific contexts. To evaluate the inclusion between two objects, represented by X and Y graphs, several works have used the function of Tversky [17] (1), Cosine (2) or Jaccard (3):

$$Tversky(X, Y) = \frac{|X| \cap |Y|}{|X| \cup |Y|} \tag{1}$$

$$Cosine(X, Y) = \frac{|X| \cap |Y|}{\sqrt{|X| \times |Y|}} \tag{2}$$

$$Jaccard(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{3}$$

avec:

- $|X|$: Cardinality of the object X
- $|Y|$: Cardinality of the object Y

IV. Structural Information Retrieval Approach

Graphs are widely used to model structural and complex data in different application domains: in imaging [18], in pattern recognition [19] [20] [21] and in chemistry [22] [23] [24]. In all these works, the problem is about looking for a relation between graphs (inclusion, intersection, etc). For example, the inclusion of graphs is used to compare graphs. More generally, the objective is to find the best match between the compared graphs, by finding the maximum number of similar nodes and edges [3].

As indicated before, this work introduces a new approach to structural IR. The underlying SSD allows to manipulate the documentary granularity in order to properly answer user's needs, expressed by a query. To optimize the IR and return the most relevant results, the approach takes into account the following remarks:

- Representing the queries and the document using graphs;
- Retaining only the relevant fragments (parts of documents);
- Using the order of relevance to order the results

The following section details the approach on structural IR. Indeed, the proposed approach is composed of three steps as shown in Fig. 1.

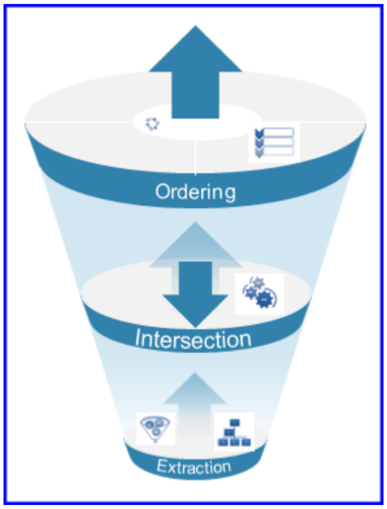


Fig. 1. General Process of the Approach on Structural IR.

A. Structure Extraction Phase

This phase consists of:

1) Extracting the structure of the document and representing it as an arborescence;
2) Extracting the structure of the query and representing it as an arborescence;
3) Extracting the paths of the arborescence representing the document;
4) Extracting the paths of the arborescence representing the query.

In the first and second steps, the structure of the document (respectively of the query) is extracted and an arborescence of the document is presented.

In the third and forth steps, the approach adapts an arborescence linearization technique. The latter system automatically reduces one of the well-known problems in graph theory which is the combinatorial cost. Proposing to consider a graph as a set of paths. Thus, comparing two graphs is equivalent to comparing the paths composing them [3]. In [3]:

- A path of k length is defined as: a series of nodes $u_0, u_1, \cdots, u_k$, such as:
  $\forall i \in [0, k-1]$, u_i and u_{i+1} are adjacent, u_0 is the path's origin and u_k is its extremity.
- **The path of G graph is a G sub-graph.**
- The depth of an arc in a path is the length of the path from the root to this arc. In the example of Fig. 2, the depth of the A/C arc is 3 when considering the $doc/E/A/C$ path.
- B/M relation is defined from B node to M node if there is an arc that associates the former with the latter (Fig. 2).

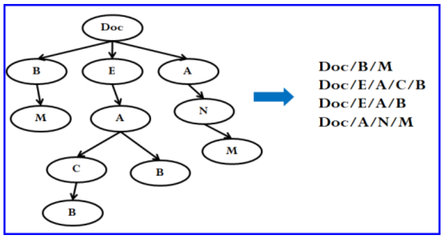


Fig. 2. Example of Path Extraction from a given Arborescence.

B. The Intersection Between the Query and the Document

In this phase, calculation of the inclusion $DegIn(p_q, p_d)$ between the query and the document paths is implemented. This inclusion (according to the proposed approach) enables the evaluation of the intersection between p_q and p_d paths.

$$DegIn(p_q, p_d) = \frac{|NCR(p_q, p_d)|}{|p_q|} \qquad (4)$$

with;

- p_d: a path of the document;
- p_q: a path of the query;
- $|NCR(p_q, p_d)|$: The number of the common relations that exists between p_q and p_d;
- $|p_q|$: number of relations in p_q.

If all relations of the query's path p_q exist in the document's path p_d, the value of equation (4) is equal to 1. The value of (4) reflects the ratio of the number of common relations between the document's path and the query's one.
In the arborescence structure representing the structure of a document (respectively of the query), it is interesting to take into account the depth of a relation (arc) because it expresses the context of this relation in the path.

C. Ordering Results

This phase presents results in order of relevance. Hence, it is composed of three main steps:

1) Considering the depth of common relations between the query and document paths;
2) Applying the smoothing function;
3) Ordering the results according to the $score(p_q, p_d)$.

The final score $score(p_q, p_d)$ is defined by:

$$score(p_q, p_d) = DegIn(p_q, p_d) \times smooth(p_q, p_d) \qquad (5)$$

This shows that taking into account the depth of common relations between the query and document paths is effective. In this case, the smoothing technique is established as follows.

$$smooth(p_q, p_d) = \log_{10}(e + \frac{1}{1 + |dist(p_q, p_d)|}) \qquad (6)$$

The smoothing function takes into consideration the distance $dist(p_q, p_d)$ (see Fig. 3).

$$dist(p_q, p_d) = depLast(p_d) - depCom(p_q, p_d) \qquad (7)$$

- $dist(p_q, p_d)$: the distance between the last relation of the document path and the last common relation between p_q and p_d;
- $depLast(p_d)$: depth of the last relation of the document path;
- $depCom(p_q, p_d)$: depth of the last common relations between p_q and p_d.

Moreover, this logarithm shows that a significant increase in the distance $dist(p_q, p_d)$ does not generate a large increase in the final score $score(p_q, p_d)$. Several studies have used the smoothing function to reduce irregularities. Smoothing is a technique that consists in reducing irregularities and singularities of a curve in mathematics.
Fig. 3 introduces a computational application to calculate the final score.

TABLE I. Comparing the Proposed Method with other Approaches

Tversky method	Cosine method	Jaccard method	The proposed method
$Tve(R, G_1) = 1$	$Cos(R, G_1) = 1$	$Jac(R, G_1) = 1$	$DegIn(R, G_1) = 0.25$
$Tve(R, G_2) = 1$	$Cos(R, G_2) = 1$	$Jac(R, G_2) = 1$	$DegIn(R, G_2) = 0.5$
$Tve(G_0, G_3) = 1$	$Cos(R, G_3) = 1$	$Jac(R, G_3) = 1$	$DegIn(R, G_3) = 0$

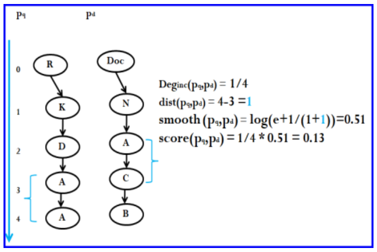

pq
pd
0
1
2
3
4
R
K
D
A
A
Doc
N
A
C
B
Deginc(pq,pd) = 1/4
dist(pq,pd) = 4-3 =1
smooth (pq,pd) = log(e+1/(1+1))=0.51
score(pq,pd) = 1/4 * 0.51 = 0.13

Fig. 3. Numerical Application for Score Calculation.

V. Evaluating the Proposed Approach

A. Comparing the Proposed Approach with that of Tversky, Jaccard and Cosinus

Through the example illustrated in Fig. 4 , it is demonstrated in Table I that Tversky, Jaccard and Cosine methods do not take into account the distribution of the graph's component. G_1, G_2 and G_3 arborescences represent the structure of the four SSD, and R stand for the query's structure.

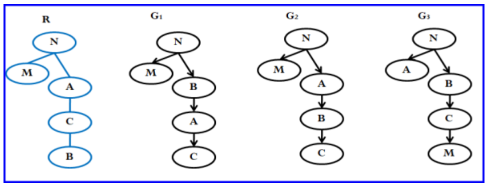


Fig. 4. Example of Arborescences representing the Documents and the Query.

In the proposed approach, the information conveyed by the document structure is unavoidable. The relations constitute the backbone of the graph since they make explicit the nature of the links and add contextual, structural and semantic information [4].

In [3], the methods of Jaccard, Cosine, Tversky and more broadly the surface measures do not take into consideration the information conveyed by the structure. However, the same components (image, text, audio, etc) may not play the same role in two different documents.

This is explained through the exploitation of the different arcs connecting the nodes of the graph that represents the document.

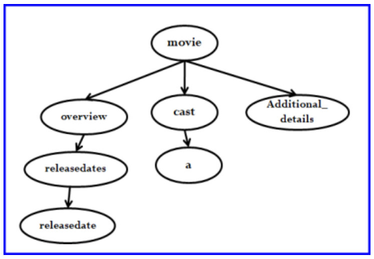


Fig. 5. Arborescence of the Query.

B. Experimentation and Discussion

To evaluate the approach, collections of multimedia documents extracted from INEX (Initiative for the Evaluation of XML Retrieval) 2010 [1] and a corpus of queries are used.

Table II and III presents the description of the corpus of queries and documents.

TABLE II. Description of the Corpus Including the Documents

	Characteristics
Corpus size	10,5 Go
Number of documents in the corpus	4 418 083
Number of documents in the sample	1000
Total number of nodes in the sample	47882
Average depth/document	3.63

TABLE III. Description of the Corpus Containing the Queries

	Characteristics
Queries' number in the corpus	5
Total number of nodes in the corpus	21
Average depth /queries	3.18

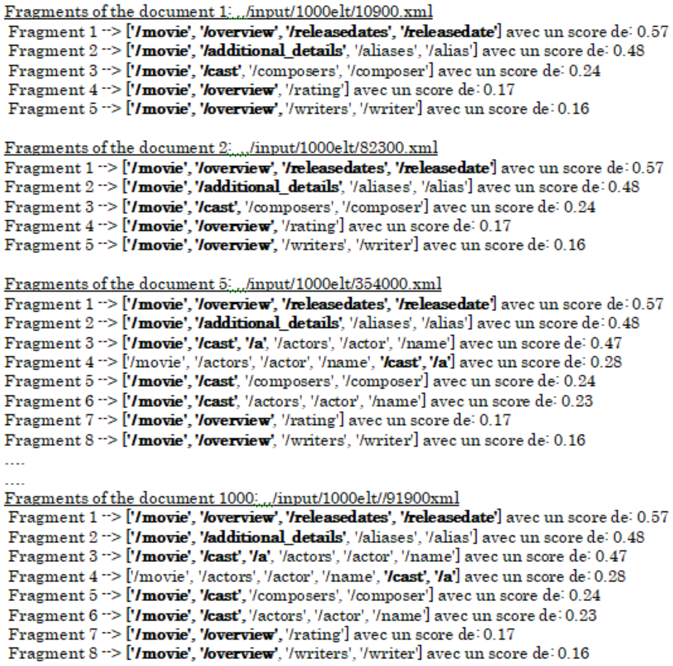
The paper have designed a prototype using Python language which is composed of two modules: the first one relies on SAX (Simple API for XML) API (Application Programming Interface), to carry out the first parsing. This module provides an intermediate file intercepted by a second module to refine the query and order the results according to their relevance.

Next, an extract of the results obtained by applying a query on all the documents of the corpus is presented. Fig. 6 presents an extract of obtained results, when applying the query represented by the arborescence structure in Fig. 5 on the documents of the corpus. The approach allows to return the fragments (parts of documents) considered relevant to the user's query.

[1] It focuses on rich structures in which classical IR models are not sufficient. In other words, it is about the labels of structural elements that carry essential information that the textual content does not specify. This new task is based on a corpus extracted from the IMDB website which contains 1,590,000 movies and several million actors, producers, directors, etc.

For example, the results of the ordering of the **document** 1 in Fig. 6 are all correct, according to the proposed approach: The path ***[’/movie’, ’/overview’, ’/releasedates’, ’/releasedate’]*** is ordered first in the retrieval, while ***[’/movie’, ’/additional_details’, ’/aliases’, ’/alias’]*** is ordered second in the retrieval.
For the third ordering, it is taken by the path ***’/movie’, ’/cast’, ’/composers’, ’/composer’]***, because, it provides a score value of 0.24.
The fourth position is taken by ***[’/movie’, ’/overview’, ’/rating’]***, because, it presents a value of 0.17 for the score value.
The fifth position is taken by ***[’/movie’, ’/overview’, ’/writers’, ’/writer’]***, because, it presents a value of 0.16 for the value of the score.

While analyzing the displayed results, it is noted that the system privileges the paths composed of the relations closest to the leaves, specifically, where two paths of a document have the same number of relations in common with a given path of the query. The proposed approach privileges the paths with relations that are (in common) relatively close to the leaf (Fig. 6).



```
Fragments of the document 1: ../input/1000elt/10900.xml
 Fragment 1 --> ['/movie', '/overview', '/releasedates', '/releasedate'] avec un score de: 0.57
 Fragment 2 --> ['/movie', '/additional_details', '/aliases', '/alias'] avec un score de: 0.48
 Fragment 3 --> ['/movie', '/cast', '/composers', '/composer'] avec un score de: 0.24
 Fragment 4 --> ['/movie', '/overview', '/rating'] avec un score de: 0.17
 Fragment 5 --> ['/movie', '/overview', '/writers', '/writer'] avec un score de: 0.16

Fragments of the document 2: ../input/1000elt/82300.xml
Fragment 1 --> ['/movie', '/overview', '/releasedates', '/releasedate'] avec un score de: 0.57
Fragment 2 --> ['/movie', '/additional_details', '/aliases', '/alias'] avec un score de: 0.48
Fragment 3 --> ['/movie', '/cast', '/composers', '/composer'] avec un score de: 0.24
Fragment 4 --> ['/movie', '/overview', '/rating'] avec un score de: 0.17
Fragment 5 --> ['/movie', '/overview', '/writers', '/writer'] avec un score de: 0.16

Fragments of the document 5: ../input/1000elt/354000.xml
Fragment 1 --> ['/movie', '/overview', '/releasedates', '/releasedate'] avec un score de: 0.57
Fragment 2 --> ['/movie', '/additional_details', '/aliases', '/alias'] avec un score de: 0.48
Fragment 3 --> ['/movie', '/cast', '/a', '/actors', '/actor', '/name'] avec un score de: 0.47
Fragment 4 --> ['/movie', '/actors', '/actor', '/name', '/cast', '/a'] avec un score de: 0.28
Fragment 5 --> ['/movie', '/cast', '/composers', '/composer'] avec un score de: 0.24
Fragment 6 --> ['/movie', '/cast', '/actors', '/actor', '/name'] avec un score de: 0.23
Fragment 7 --> ['/movie', '/overview', '/rating'] avec un score de: 0.17
Fragment 8 --> ['/movie', '/overview', '/writers', '/writer'] avec un score de: 0.16
....
....
Fragments of the document 1000: ../input/1000elt//91900xml
 Fragment 1 --> ['/movie', '/overview', '/releasedates', '/releasedate'] avec un score de: 0.57
 Fragment 2 --> ['/movie', '/additional_details', '/aliases', '/alias'] avec un score de: 0.48
 Fragment 3 --> ['/movie', '/cast', '/a', '/actors', '/actor', '/name'] avec un score de: 0.47
 Fragment 4 --> ['/movie', '/actors', '/actor', '/name', '/cast', '/a'] avec un score de: 0.28
 Fragment 5 --> ['/movie', '/cast', '/composers', '/composer'] avec un score de: 0.24
 Fragment 6 --> ['/movie', '/cast', '/actors', '/actor', '/name'] avec un score de: 0.23
 Fragment 7 --> ['/movie', '/overview', '/rating'] avec un score de: 0.17
 Fragment 8 --> ['/movie', '/overview', '/writers', '/writer'] avec un score de: 0.16
```

Fig. 6. Results Extract.

VI. Conclusion and Perspectives

This work comes in the context of IR in SSD that uses more particularly XML documents. With respect to the works studied in section 2, the proposed approach is a part of the approaches using graphs to represent objects. The proposed approach seeks to improve the automation of IR, by facilitating access to the document granule (while preserving the characteristics of each document: content and structure). For instance, this enables to optimize the access to relevant information. it has been proven that: (1) The approach is based on graphs, taking into account the relations between the components of the documents and proving that this is a crucial parameter in the proposed structural IR approach. As a matter of fact, the content of a multimedia document does not only depend on the content of its components, but also on the relations between these components. (2) To overcome the combinatorial problem (well-known in graph theory), by conceiving a graph as a set of paths. The correspondence of the paths of the graphs allows to preserve both the contextual and hierarchical aspects of the components. (3) It have shown that the so-called surface measures cannot answer the problem. Hence, the first results obtained are encouraging.

In future research, a comparative study with other works will be done. Finally, to make the proposed approach more complete, it is important to take into account both the documentary content and the structure that conveys a significant amount of information.

References

[1] Martin Bryan et al. Guidelines for using xml for electronic data interchange. *SGML Centre*, 1998.

[2] Karen Sauvagnat. *Modèle flexible pour la recherche d’information dans des corpus de documents semi-structurés*. PhD thesis, Université Paul Sabatier-Toulouse III, 2005.

[3] Ali Idarrou. *Entreposage de documents multimédias: comparaison de structures*. PhD thesis, Toulouse 1, 2013.

[4] Hasna Abioui Assmaa Moutaoukkil Ali Idarrou Imane Belahyane, Mouad Mammass. Comparative study on graph-based information retrieval: the case of xml document. *International Journal of Advanced Engineering, Management and Science*, 2021.

[5] Cyril Laitang, Mohand Boughanem, and Karen Pinel-Sauvagnat. Xml information retrieval through tree edit distance and structural summaries. In *Asia Information Retrieval Symposium*, pages 73–83. Springer, 2011.

[6] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[7] Erik D Demaine, Shay Mozes, Benjamin Rossman, and Oren Weimann. An optimal decomposition algorithm for tree edit distance. In *International Colloquium on Automata, Languages, and Programming*, pages 146–157. Springer, 2007.

[8] Serge Dulucq and Hélene Touzet. Analysis of tree edit distance algorithms. In *Annual Symposium on Combinatorial Pattern Matching*, pages 83–95. Springer, 2003.

[9] Philip N Klein. Computing the edit-distance between unrooted ordered trees. In *European Symposium on Algorithms*, pages 91–102. Springer, 1998.

[10] Cyril Laitang, Karen Pinel-Sauvagnat, and Mohand Boughanem. Dtd based costs for tree-edit distance in structured information retrieval. In *European Conference on Information Retrieval*, pages 158–170. Springer, 2013.

[11] Robert W Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

[12] Samaneh Chagheri, Catherine Roussey, Sylvie Calabretto, and Cyril Dumoulin. Classification de documents combinant la structure et le contenu. In *8ème COnférence en Recherche d’Information et Applications CORIA 2012*, pages p–261, 2013.

[13] Maël Minot, Samba Ndojh Ndiaye, and Christine Solnon. Recherche d’un plus grand sous-graphe commun par décomposition du graphe de compatibilité. *Onzièmes Journées Francophones de Programmation par Contraintes (JFPC)*, pages 1–11, 2015.

[14] Samba Ndojh Ndiaye and Christine Solnon. Cp models for maximum common subgraph problems. In *International Conference on Principles and Practice of Constraint Programming*, pages 637–644. Springer, 2011.

[15] Christine Solnon and Serge Fenet. A study of aco capabilities for solving the maximum clique problem. *Journal of Heuristics*, 12(3):155–180, 2006.

[16] Philippe Jégou. Decomposition of domains based on the micro-structure of finite constraint-satisfaction problems. In *AAAI*, volume 93, pages 731–736, 1993.

[17] Amos Tversky. Features of similarity. *Psychological review*, 84(4):327, 1977.

[18] Imane Belahyane, Mouad Mammass, Hasna Abioui, and Ali Idarrou. Graph-based image retrieval: State of the art. In *International Conference on Image and Signal Processing*, pages 299–307. Springer, 2020.

[19] Pasquale Foggia, Gennaro Percannella, and Mario Vento. Graph matching and learning in pattern recognition in the last 10 years. *International Journal of Pattern Recognition and Artificial Intelligence*, 28(01):1450001, 2014.

[20] Xavier Cortés, Donatello Conte, and Hubert Cardot. Learning edit cost estimation models for graph edit distance. *Pattern Recognition Letters*, 125:256–263, 2019.

[21] Luc Brun, Pasquale Foggia, and Mario Vento. Trends in graph-based representations for pattern recognition. *Pattern Recognition Letters*, 134:3–9, 2020.

[22] Samuel Wieczorek. *Une mesure d'inclusion entre objets structurés. Application à la classification de molécules.* PhD thesis, Université Joseph-Fourier-Grenoble I, 2009.

[23] Frédéric Suard and Alain Rakotomamonjy. Mesure de similarité de graphes par noyau de sacs de chemins. In *21 Colloque GRETSI, Troyes, FRA, 11-14 septembre 2007*. GRETSI, Groupe d'Etudes du Traitement du Signal et des Images, 2007.

[24] Angiras Menon, Nenad B Krdzavac, and Markus Kraft. From database to knowledge graph—using data in chemistry. *Current Opinion in Chemical Engineering*, 26:33–37, 2019.