Hybrid Molecule-based Information Retrieval

Nathalie Charbel, Christian Sallaberry, Sébastien Laborie, Richard Chbeir

Hybrid Molecule-based Information Retrieval

Nathalie Charbel
Univ Pau & Pays Adour, E2S UPPA, LIUPPA, EA3000, Anglet, 64600, France
nathalie.charbel@univ-pau.fr

Christian Sallaberry
Univ Pau & Pays Adour, E2S UPPA, LIUPPA, EA3000, Pau, 64000, France
christian.sallaberry@univ-pau.fr

Sebastien Laborie
Univ Pau & Pays Adour, E2S UPPA, LIUPPA, EA3000, Anglet, 64600, France
sebastien.laborie@univ-pau.fr

Richard Chbeir
Univ Pau & Pays Adour, E2S UPPA, LIUPPA, EA3000, Anglet, 64600, France
rchbeir@acm.org

ABSTRACT

The increased availability of interdependent heterogeneous data generated from different sources is fostering the incorporation of semantic knowledge-based graphs and ontologies in information management and search applications. Most of the existing Information Retrieval systems mainly focus on the semantic analysis of the information contained in heterogeneous data. In their results, they provide documents as query answers without considering (i) detailed information regarding relevant granularity levels of the documents, and most importantly (ii) dependencies between the documents or parts of the documents. To overcome these limitations, we propose a graph-based search and ranking algorithm within a generic framework that retrieves the data in the form of a novel augmented data structure for query answers, which we call hybrid molecules. The latter consist of well-defined subgraphs representing relevant contextual information regarding domain-specific information coupled with structural information related to the document. This improves the search results and reduces users' efforts in tracking and interpreting them. Experiments conducted on real world data corpus using projects from the building construction industry validate the effectiveness of our approach.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; *Document representation*; *Ontologies*; *Enterprise search*;

KEYWORDS

Tightly Coupled Semantic Graphs, Hybrid Molecules

1 INTRODUCTION

Web and Information Systems are increasingly adopting semantic knowledge-based models to represent the data encapsulated in heterogeneous resources [2]. This has several proven benefits in improving users' experience in search applications [1]. As an example, in several industries involving multidisciplinary projects, users looking for a particular information have to search through heterogeneous documents provided by different sources. It is also common for the documents to have several dependencies between them (e.g., documents with related topics, references between documents or parts of documents) [4]. Figure 1 illustrates a sample of two interdependent documents from the construction industry: *d1* (a technical specification document) and *d2* (a material thermal report) related to the same project. Parts of document *d1* describe acoustic and thermal properties (*section A* and *section B* respectively). Document *d2* describes a thermal study which includes details on the solar factors of windows (*section 1*), implicitly described by "*SW coefficient*". The reference relation between the two documents shows that *section 1* of *d2* contains complementary information to *section B* of *d1*. Consider that the user is searching for *"the solar factor of windows"*. He is not interested in getting only relevant documents, but also relevant contextualized and precise answers (e.g., the value of solar factor, the part of the document describing it and the related parts of documents containing additional information), so he makes less efforts in interpreting the results.

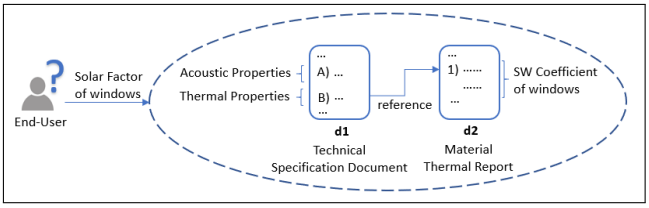


Figure 1: Sample documents from the construction industry.

Traditional Information Retrieval (IR) approaches mainly rely on syntactic keyword-based search [9]. To overcome their limitations, there has been significant interest in taking semantics into account leading to the emergence of Semantic Information Retrieval (SIR) systems. Although suitable for several applications [7, 8, 10, 13], SIR systems provide documents as query answers without considering in their results (i) detailed information regarding relevant granularity levels of the documents (e.g., *section B* of *d1* in Figure 1), and most importantly (ii) inter and intra-document dependencies (e.g., the reference between *section B* of *d1* and *section 1* of *d2*).

In this paper, we provide a solution to the aforementioned limitations of current SIR systems. In [4, 10, 13], authors demonstrated the importance of adopting a tightly coupled semantic graph[1]. Similarly, we take advantage of such a model to provide a novel data structure for query answers, which we call *hybrid molecules*. The

[1]The data objects (documents, web pages, tuples, etc.) and their metadata are individuals coupled with those of a lexical knowledge base or domain-specific ontology.

latter consist of hybrid subgraphs encapsulating domain-specific information coupled with related structural information of the documents [4]. The hybrid molecule-based query answers bring in helpful contextual information of the documents improving the search results and reducing users' efforts in tracking and interpreting them.

In this context, we tackle the following challenges: (i) formally defining the hybrid molecule's structure in view of the characteristics of a tightly coupled semantic graph and the definition of a molecule concept in the literature [5, 6], and (ii) adopting an effective graph-based approach to construct the hybrid molecules based on the associated definition and rank them conveniently.

The literature provides a vast array of graph-based search algorithms [15]. Constrained Spread Activation (CSA) [17] stands out as a simple yet effective solution in many IR applications [10, 11, 14, 17]. Inspired by this search approach, we propose an algorithm which we call *HM_CSA* that provides a ranked list of hybrid molecules as query answers instead of single nodes.

The contributions of this paper which can be summarized by

- *Hybrid molecule*, a novel data structure for query answers
- *HM_CSA*, an algorithm that constructs and ranks relevant hybrid molecules from a tightly coupled semantic graph

are presented within a SIR framework entitled FEED2SEARCH (FramEwork for hybrid molEcule-baseD SEmantic SEARCH) where users submit their natural language (e.g., plain English text) queries over a heterogeneous document corpus and obtain relevant answers in the form of hybrid molecules.

This paper is structured as follows. In Section 2, we introduce FEED2SEARCH. We present in Section 3 fundamental notions of a tightly coupled semantic graph required to understand the hybrid molecules, which we introduce in Section 4. Section 5 integrates the hybrid molecules in a graph-based search and ranking algorithm. In Section 6, we evaluate the proposed solution on data collected from projects in the construction industry. Section 7 reviews the related work, and Section 8 concludes remarks and future work.

2 FEED2SEARCH FRAMEWORK

This section presents FEED2SEARCH, a novel generic framework which stands for FramEwork for hybrid molEcule-baseD SEmantic SEARCH over a heterogeneous document corpus. The main purpose is to facilitate the query processing over a heterogeneous document corpus for non computer expert users by providing them with relevant answers in response to their natural language queries. As presented in Figure 2, FEED2SEARCH is made of two interconnected layers where the upper layer uses data and services provided by the lower layer.

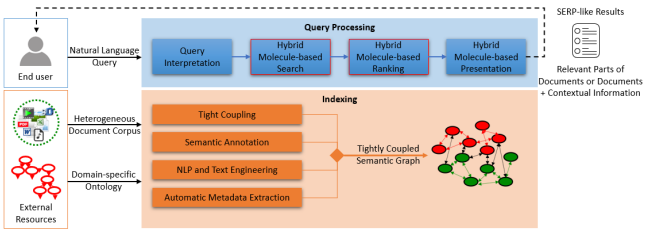


Figure 2: Overall architecture of FEED2SEARCH.

The main purpose of the bottom layer (Indexing) is to index a given heterogeneous document corpus and provide a tightly coupled semantic graph (See Definition 3) made of instances describing the document corpus. This is done through annotators handling: (1) Automatic Metadata Extraction of the documents to generate structural-based instances i.e., instances describing the documents, their metadata, their decomposition into different granularity levels and relations between these granularities [4]; (2) Natural Language Processing (NLP) and Text Engineering [3] to identify regular expressions and generate further relations (such as references) between documents or parts of documents; (3) Semantic Annotation [8] to automatically generate domain-specific instances based on any relevant external domain-specific ontology; and (4) Tight Coupling to generate relations between domain-specific instances and structural-based instances. A coupling between two instances of different types is generated if a domain-specific instance was previously extracted, in the Semantic annotation module, from the content of a structural-based instance.

The upper layer (Query processing) comprises four sequential modules to process the query from the end-user's input (natural language query) to the final system's output (documents or parts of documents with relevant contextual information presented in SERP[2]-like results). In the first module, query interpretation runs classical NLP techniques followed by semantic annotation. This module outputs a set of domain-specific instances extracted from the query based on the domain-specific ontology at hand. These instances are then considered as input for the second module where they are matched[3] against instance nodes of the tightly coupled semantic graph. The matched instance nodes are used to initialize HM_CSA as described in Section 5. The third module uses HM_CSA to output a ranked list of hybrid molecules. The forth module transforms, through several operations, the ranked list of hybrid molecules into SERP-like results.

In this paper, we particularly focus on the *Hybrid Molecule-based Search and Ranking* modules as they comprise our two contributions (the Hybrid Molecules and HM_CSA).

3 PRELIMINARIES

We rely on a specific type of semantic graphs i.e., a tightly coupled semantic graph to couple domain-specific information with its related structural information within a heterogeneous document corpus. This type of graph leverages semantics at the finest granularity levels of the documents. We generate this graph using two underlying external resources: a heterogeneous document corpus and a background domain-specific ontology.

3.1 Underlying External Resources

We define a heterogeneous document corpus and a domain-specific ontology as follows:

Definition 1 (*Heterogeneous Document Corpus*). A heterogeneous document corpus δ is defined as a set of n documents originated from different sources. The structure of the documents does not necessarily follow a common standard. Formally, $\delta = \{d_1, \ldots, d_n\}$. A document $d_i \in \delta$ is characterized by a set of p metadata (e.g., author, format, creation date, etc.) and a set of q media (e.g.,

[2]Search Engine Results Page.
[3]Using concept matching i.e., instances are said to be matched if they are instances of the same concept.

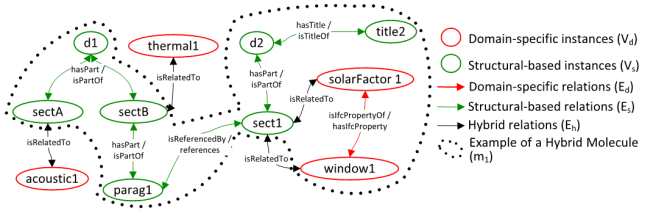


Figure 3: Extract of a tightly coupled semantic graph representing a heterogeneous document corpus δ from the construction industry.

text, image, video, or audio), such that $d_i = \{meta_1, \ldots, meta_p, med_1, \ldots, med_q\}$. Further, a media $med_l \in d_i$ is made of s media components, such that $med_l = \{medComp_1, \ldots, medComp_s\}$, (e.g., section, paragraph, etc. for the text; still region, object, etc. for the image).

Definition 2 (*Domain-specific Ontology*). Given an application domain $\mathcal{D}$, a domain-specific ontology designated as $O_{\mathcal{D}}(C, R, Lit, A, L, f_L)$ is the semantic knowledge describing information in $\mathcal{D}$, where:

- C is the set of domain-specific concepts. For instance, considering the construction industry, such concepts include, but not limited to, entities describing building elements (e.g., *Window*, *Door*, *Wall*) and thermal properties (e.g., *SolarFactor*, *Insulation*).
- $R \subseteq C \times C$ is the set of relations between domain-specific concepts in C. For instance, $r_1 = (Wall, Window)$ is a spatial containment relation between *Wall* and *Window*, where $r_1 \in R$.
- $Lit = \{Integer, Decimal, String, \ldots\}$ is the set of literal types.
- $A \subseteq C \times Lit$ is the set of attributes describing domain-specific concepts i.e., relations between domain-specific concepts in C and literals in Lit. For instance, *WindowHeight* is an attribute property linking *Window* to *Decimal*.
- L is the set of relation labels.
- $f_L : R \rightarrow L$ is the association function that assigns a label $l \in L$ to a domain-specific relation $r \in R$, hence $f_L(r) = l$. For instance, $f_L(r_1) =$ *"containsElement"*.

For ease of presentation, $O_{\mathcal{D}}(C, R, Lit, A, L, f_L)$ will be referred to as $O_{\mathcal{D}}$ in the remainder of the paper.

3.2 Tightly Coupled Semantic Graph

In the context of a SIR application, we need to leverage the semantics of $O_{\mathcal{D}}$ over a heterogeneous document corpus δ related to a domain $\mathcal{D}$. A coupling is required between concepts of $O_{\mathcal{D}}$ and elements of δ on different granularity levels. This ensures the ability to locate a relevant information at different levels of precision, and most importantly tracking the document interdependencies at finest structural elements.

Existing works [4, 10, 13] have adopted tightly coupled semantic graphs in their approaches. Rocha et al. [10] couple web pages and a domain-specific ontology. Tekli et al. [13] couple a lexical semantic network and a standard inverted index. Charbel et al. [4] couple two semantic graphs: the first one represents the documents, their structure and relations between them, and the second one represents concepts and relations of pluggable domain-specific ontologies. Inspired by these approaches, we formally define our tightly coupled semantic graph. We also provide examples based on Figure 3, which resumes our motivating scenario (See Figure 1) in the form of the following graph model:

Definition 3 (*Graph Model*). Given a heterogeneous document corpus δ and a domain-specific ontology $O_{\mathcal{D}}$, we define $\mathcal{G}_{\delta}(V, E, Val, f_{Val}, Lab, f_{Lab}, W, f_{W_v}, f_{W_e})$ as the instances graph describing the structural and domain-specific knowledge in δ, where:

- V is the set of nodes representing instances of $O_{\mathcal{D}}$ and δ:
  - $V = V_d \cup V_s$.
  - $V_d \subset V$ is the subset of domain-specific nodes where a node $v_d \in V_d$ represents an instance of $c \in C$ in $O_{\mathcal{D}}$, e.g., $solarFactor1$ (instance of the concept *SolarFactor*).
  - $V_s \subset V$ is the subset of structural-based nodes where a node $v_s \in V_s$ represents any granularity element in δ i.e., a document $d_i \in \delta$ (e.g., $d1$), a metadata $meta_k \in d_i$ (e.g., $title2$), a media $med_l \in d_i$ or more precisely a media component $medComp_r \in med_l$ (e.g., $sectA$).
  - $V_d \cap V_s = \emptyset$[4].
- E is the set of directed edges representing relations between nodes:
  - $E = E_d \cup E_s \cup E_h$.
  - $E_d \subseteq V_d \times V_d$ is the subset of domain-specific edges where an edge $e_d \in E_d$ represents an instance of a relation $r \in R$ in $O_{\mathcal{D}}$, e.g., $e_{d1} = (window1, solarFactor1)$ is the edge linking $window1$ to $solarFactor1$.
  - $E_s \subseteq V_s \times V_s$ is the subset of structural-based edges where an edge $e_s \in E_s$ represents a relation between structural-based nodes in V_s (such as *part-whole*, *reference*, etc.), thus augmenting the representation of δ; e.g., $e_{s1} = (d2, sect1)$ is the edge linking $d2$ to $sect1$.
  - $E_h \subseteq V_d \times V_s$ is the subset of hybrid edges where an edge $e_h \in E_h$ represents a tight coupling, i.e. a relation between a node $v_d \in V_d$ and a node $v_s \in V_s$; e.g., $e_{h1} = (solarFactor1, sect1)$ is the edge linking $solarFactor1$ to $sect1$.
  - $E_d \cap E_s = \emptyset$, $E_s \cap E_h = \emptyset$, $E_d \cap E_h = \emptyset$, and $E_d \cap E_s \cap E_h = \emptyset$[5].
- Val is the set of node literal values[6]. For the sake of simplicity, we omit these values from the graph depicted in Figure 3.
- $f_{Val} : V \rightarrow Val$ is the association function that assigns to a node $v \in V$ a literal value $val \in Val$, hence $f_{Val}(v) = val$; e.g., $f_{Val}(sect1) =$ *"SW Coefficient of Windows"* where *"SW Coefficient of Windows"* is the string value associated to $sect1$.
- Lab is the set of edge labels.
- $f_{Lab} : E \rightarrow Lab$ is the association function that assigns a label $lab \in Lab$ to an edge $e \in E$, hence $f_{Lab}(e) = lab$; e.g., $f_{Lab}(e_{s1}) =$ *"hasPart"*.
- W is the set of both nodes and edges' weights.
- $f_{W_v} : V \rightarrow W$ is the node weight mapping that assigns a weight $w_v \in W$ to a node $v \in V$.
- $f_{W_e} : E \rightarrow W$ is the edge weight mapping that assigns a weight $w_e \in W$ to an edge $e \in E$. f_{W_e} consists of a set

[4]We distinguish nodes of the subset V_s from those of the subset V_d to explicitly differentiate between structural characteristics and domain-specific ones.
[5]We also distinguish between edges of different subsets such as the case for nodes.
[6]The value of a node $v_s \in V_s$ is made of its content, whereas the value of a node $v_d \in V_d$ consists of the concatenated values of its attributes $A_{v_d} \subseteq A$ in $O_{\mathcal{D}}$.

of functions which adapt to E_s, E_d, and E_h depending on different rationales.
The weight mapping functions f_{W_v} and f_{W_e} are used in the search process of query answers. They are detailed in Section 5.3.

For ease of presentation, $\mathcal{G}_\delta(V, E, Val, f_{Val}, Lab, f_{Lab}, W, f_{W_v}, f_{W_e})$ will be referred to as $\mathcal{G}_\delta$ in the remainder of the paper.

4 HYBRID MOLECULES

We introduce hybrid molecules which we build upon the definition of molecules in the literature (e.g., [5]). Molecules are subgraphs of connected nodes. They are extracted from an initial graph using a decomposition function. We propose adjusting the decomposition to better cope with our tightly coupled semantic graph and provide meaningful subgraphs.

Definition 4 (*Hybrid Molecule*). Given the instances graph $\mathcal{G}_\delta$ describing a heterogeneous document corpus δ, we define a hybrid molecule $m(e_h, V_m, E_m, w_m, f_{W_m})$, also denoted by $m \in M$, as a subgraph decomposition result from the initial graph $\mathcal{G}_\delta$ based on a coupling between a domain-specific node and its related structural-based node, where:

- $M = d_{E_h}(\mathcal{G}_\delta)$ is the set of hybrid molecules representing subgraphs obtained from the decomposition function d_{E_h}. This function splits the initial graph $\mathcal{G}_\delta$ into molecules whenever a hybrid edge $e_h \in E_h$ is identified, such that each molecule $m \in M$ has a unique $e_h \in E_h$.
- $e_h \in E_h$ is the hybrid edge identifying the molecule m. Also, we refer to e_h as the *core* of the molecule m. We denote by $e_h.v_d \in V_d$ the domain-specific node of the molecule's core and $e_h.v_s \in V_s$ the structural-based node of the molecule's core.
- $V_m \subseteq (V_s \cup V_d)$ is the subset made of domain-specific and structural-based nodes forming m.
- $E_m \subseteq (\{e_h\} \cup E_d \cup E_s)$ is the subset made of the core edge, domain-specific and structural-based edges forming m and linking nodes $v \in V_m$.
- w_m is the overall weight of the molecule m such that $w_m \in W_m$, where W_m is the set of molecules' weights.
- $f_{W_m} : M \rightarrow W_m$ is the molecule weight mapping that assigns the weight $w_m \in W_m$ to the molecule $m \in M$.
The molecule weight mapping f_{W_m} is used in the ranking process of query answers. It is detailed in Section 5.3.3.

The core of a hybrid molecule holds the molecule's central information as it is where the domain specific knowledge is anchored to a structural component in the document corpus. The rest of the molecule's nodes and edges augments the core with additional relevant information.

Illustrative Example: The dashed area in Figure 3 is an example of a hybrid molecule m_1, where $e_{h1} = (solarFactor1, sect1)$ is the molecule's core. $sect1$ contains relevant information on the solar factor. Other structural-based components (e.g., $e_{s1} = (d2, sect1)$) and domain-specific ones (e.g., $e_{d1} = (window1, solarFactor1)$) in V_m and E_m provide m_1 with further useful information (i.e., $sect1$ is part of document $d2$ and $solarFactor1$ is a property of $window1$).

5 HYBRID MOLECULE-BASED SEARCH AND RANKING BY CONSTRAINED SPREAD ACTIVATION (HM_CSA)

After having introduced the hybrid molecule's structure, we propose in this section a graph-based search and ranking algorithm in order to generate a ranked list of hybrid molecules as query answers within a SIR system.

CSA [17] is one form of the Breadth-First Search (BFS) family of graph-based search algorithms [15]. It works by spreading out the activation from a set of start nodes to adjacent nodes progressively until predefined constraints are met. We consider CSA as a suitable search strategy for our tightly coupled semantic graph as (i) it handles the heterogeneity of the graph since it can explore possibly relevant structural-based and domain-specific nodes located anywhere in the graph, (ii) it constructs progressively multiple target nodes from activated nodes, and (iii) it supports the incorporation of useful constraints, either at the beginning to select start nodes or at termination point to stop the spreading in the graph. Although, in its standard form, CSA stands out as an effective and suitable solution for many IR applications [10, 11, 14], it outputs a ranked list of single nodes. Thus, we propose extending CSA to HM_CSA to generate a ranked list of hybrid molecules as query answers. HM_CSA also adapts edge weights differently to handle the characteristics of hybrid molecules and rank them appropriately.

5.1 Constrained Spread Activation (CSA)

HM_CSA builds on the CSA theory as presented in Algorithm 1 (excluding the red-highlighted area). The input of HM_CSA consists of (i) a set of domain-specific nodes V_{d_in} generated from the query interpretation module as they matched the user query (See Section 2), (ii) a tightly coupled semantic graph $\mathcal{G}_\delta$ representing a heterogeneous document corpus δ, (iii) weight parameters $\alpha, \beta \in [0, 1]$ that balance the contribution of structural and domain-specific nodes respectively in the calculation of a molecule's weight, and (iv) a set of constraints parameters, $params$. The latter consist of pre-adjustment parameters to be checked before each spread iteration occurs (e.g., a firing threshold F, and a maximum spread distance D from start nodes), post-adjustment parameters (e.g., a maximum number of iterations I, and a maximum processing time T), and spread configurations (e.g., an activation percent decrease γ which imposes a decay on the propagation of the activation through the graph). The choice of these parameters is application-dependent.

We use two sets of nodes V_{in} and V_{out}. The former is fed with activated nodes as the spread activation processes through the graph while the second consists of spread nodes i.e., nodes which have activated others and will go in the final output. At start time, V_{in} contains previously selected start nodes V_{d_in} where the activation value $Activation(v_i)$ of each node $v_i \in V_{in}$ is set to 1 (max value), and V_{out} is initially empty. Each iteration consists of removing v_i from V_{in} (lines 5-6) i.e., the node with the highest activation value[7], and check whether it is allowed to spread its activation or not to its neighbors by verifying if its distance to start nodes is less than or equal to D and its activation value is higher than or equal to F (line 7). If so, 4 main steps are applied: (1) exploring neighbors (line 8); (2) spreading out activation to each neighbor (lines 9-13); (3) processing each neighbor (lines 14-15) i.e., adding it to the set of activated nodes V_{in} if it is visited for the first time; and (4) adding

[7] If nodes have equal highest activation values, HM_CSA selects the first node.

Algorithm 1: HM_CSA

Inputs : Set of domain-specific start nodes V_{d_in}; Tightly coupled semantic graph $\mathcal{G}_\delta$; Molecule's weight parameters $\alpha, \beta \in [0, 1]$; Constraint parameters $params$;
Output: Ranked List of hybrid molecules $M_{out} \subseteq M$

1 $V_{in} \leftarrow V_{d\_in}$; // set of activated nodes
2 $V_{out} \leftarrow \emptyset$; // set of spread nodes
3 $stopSpread \leftarrow false$; // boolean checking whether to stop CSA or not
4 **while** $(|V_{in}| > 0\ AND\ !stopSpread)$ **do**
5 $v_i \leftarrow getFiringNode(V_{in})$; // node with highest activation value
6 $V_{in} \leftarrow V_{in} - \{v_i\}$; // remove firing node from V_{in}
7 **if** $(checkRestrictions(v_i, params.preadjustment))$ **then**
8 $E_{ij} \leftarrow getNeighbors(v_i)$; // set of outgoing edges from v_i to the set of direct neighbors v_j
9 **foreach** $e_{ij} \in E_{ij}$ **do**
10 // Using edge weight mapping $f_{We}(e_{ij})$
11 $\Delta_{input(v_j)} \leftarrow Output(v_i) \times f_{We}(e_{ij}) \times (1-\gamma)$; // the contribution of neighbor v_i through e_{ij}. $Output(v_i) = Activation(v_i)$ i.e., the activation value of v_i
12 $Input(v_j) \leftarrow Input(v_j) + \Delta_{input(v_j)}$; // input value of v_j
13 $Output(v_j) \leftarrow normalize(Input(v_j))$; // output of v_j i.e., its activation value after normalization function
14 **if** $(v_j \notin V_{in})$ **then**
15 $V_{in} \leftarrow V_{in} \cup \{v_j\}$; // add neighbor v_j to the set of activated nodes V_{in}
16 **else**
17 // Molecules Construction and Processing
18 **if** $(v_j \in V_{out})$ **then**
19 **if** $(isHybrid(e_{ij}))$ **then**
20 $m_i \leftarrow createMolecule(e_{ij})$; // new molecule m_i
21 $m_i \leftarrow appendFromMolecules(m_i, M_{out})$; // append nodes and edges from existing molecules in M_{out} to the newly created molecule m_i
22 $M_{out} \leftarrow M_{out} \cup \{m_i\}$; // add m_i to M_{out}
23 **else**
24 $M_{out} \leftarrow appendToMolecules(e_{ij}, M_{out})$; // append current neighbor e_{ij} to existing molecules in M_{out}
25 **end**
26 **end**
27
28 **end**
29 **end**
30 $V_{out} \leftarrow V_{out} \cup \{v_i\}$; // add firing node v_i to V_{out} after activation's propagation is done
31 **end**
32 $stopSpread \leftarrow checkRestrictions(params.postadjustment)$;
33 **end**
34 $M_{out} \leftarrow scoreAndRankMolecules(M_{out}, alpha, beta)$; // assign weights to molecules in M_{out} calculated by f_{Wm} using α and β parameters. Rank the molecules in descending order following their individual weights.
35 **return** M_{out};

the current firing node to the set of spread nodes V_{out} (line 30) so it could be visited for molecules processing in future iterations. The same process repeats until V_{in} has no further nodes to process or post-adjustment constraints are met i.e., I and T (lines 4 and 32).

As for the activation process, each explored node v_i contributes to the input of its neighbor v_j by $\Delta_{input(v_j)} \leftarrow Output(v_i) \times f_W(e_{ij}) \times (1-\gamma)$ (line 11) where $Output(v_i)$ is its current activation value (i.e., $Activation(v_i)$), $f_W(e_{ij})$ computes the weight $w_{e_{ij}}$ of the link e_{ij} connecting v_i to v_j (See Section 5.3.1) and $(1-\gamma)$ is the decay factor. Each $\Delta_{input(v_j)}$ is then added to to the input of v_j i.e., $Input(v_j)$[8] (line 12). The actual activation value $Activation(v_j) = Output(v_j)$ of a node v_j is obtained by normalizing the sum of all the contributions it receives (line 13). HM_CSA uses a simple feature scaling function[9] to rescale values between 0 and 1.

The usual output of a CSA-based algorithm is a list of spread nodes V_{out} ranked by their activation values. In some applications, the output is augmented with subgraphs consisting of the shortest paths connecting start nodes to output nodes [10].

Illustrative Example: consider the following user's natural language query: q = *"Solar factors of windows"* applied on the graph illustrated in Figure 3. After query interpretation module, $solarFactor1$ and $window1$ are matched and selected as start nodes. Consider applying the standard CSA with these start nodes, the given graph, $D = 5$, and $F = 0$ as inputs. At termination point, one of the resulting spread nodes is $parag1$. The latter is related to $d1$ as it is part of it and to $d2$ as it references one of its sections. Thus, neither presenting $parag1$ as a single node result, nor within its augmented subgraph that connects it to the start nodes (e.g., $parag1 \longrightarrow sect1 \longrightarrow solarFactor1$) would help the user in getting this information. He still need to search for relevant structural and domain-specific context to understand the result, which is time and effort consuming especially in large graphs.

5.2 Molecules Construction and Processing

One naive way to construct the molecules from the standard CSA is to post-process the nodes in V_{out}. However, this would require re-exploring connected nodes in V_{out}. Instead, HM_CSA integrates the molecule construction process in the graph traversal while fetching and activating neighboring nodes as described in the red-highlighted section of Algorithm 1 (lines 17-26).

Two cases arise when processing a neighbor $v_j \in V_{out}$ connected to v_i through an edge e_{ij}: either e_{ij} is hybrid (i.e., $e_{ij} = e_h$) or not (i.e., $e_{ij} = e_s$ or $e_{ij} = e_d$). In the first case (lines 19-22), a new molecule m_i is constructed (according to Definition 4) with e_{ij} being the core of this molecule (line 20). Since a molecule should be also made of contextual information, HM_CSA appends connected nodes (except those connected through hybrid edges) from existing molecules in M_{out}, created in previous iterations, whenever one of the core's nodes (i.e., $e_h.v_s$ or $e_h.v_d$) is matched in these molecules (line 21). For instance, if $e_h.v_s$ already exists in molecule $m_k \in M$, the structural nodes of m_k will be selected and added to the newly created molecule. In the second case where an edge e_{ij} is not hybrid (lines 23-25), HM_CSA adds it to existing molecules in M_{out} where either v_j or both v_j and v_i exist (line 24).

Illustrative Example: Considering the same query q as in Section 5.1 and the same inputs, we apply our HM_CSA on the graph of Figure 3. This results in three molecules constructed from spread nodes (i.e., all nodes except $acoustic1$) and their connecting edges: $m1$, $m2$, and $m3$ having cores $e_{h1} = (solarFactor1, sect1)$, $e_{h2} = (window1, sect1)$, and $e_{h3} = (thermal1, sectB)$ respectively. The core of each molecule holds the central information from which other relevant contextual information is provided by either its connected structural nodes or its domain-specific ones. Following the example from Section 5.1, $parag1$ is now part of the three molecules. $parag1$ plays different roles in each of the molecules. In $m1$ and $m2$, the relation $(parag1, sect1)$ adds to e_{h1} and e_{h2} an inter-document

[8] At start time, the input value $Input(v_j)$ is set to the initial activation value of v_j i.e., 1 for start nodes and 0 for others.

[9] $Output(v_j) = \frac{Input(v_j) - Input_{min}}{Input_{max} - Input_{min}}$ where $Input_{min}$, $Input_{max}$ are respectively the minimum and the maximum values of all nodes' inputs in the graph.

link information. In m_3, the relation $(parag1, sectB)$ adds to e_{h3} containment information. This allows the user to better interpret the results, especially when he tracks cross-document dependencies, which is challenging and time-consuming.

5.3 Weight Mapping

In HM_CSA, the edge weight mapping function f_{W_e}, used in the blue-highlighted section of Algorithm 1, is the weighting function that affects the most the output of the search algorithm. This is because it directly controls the contributions of neighboring nodes on the activation value of a given node (lines 11-13), which in turn, affects the final weight of the hybrid molecule query answers encapsulating it (line 34). In the literature, however, defining an edge weighting function remains application dependent [10]. In the following, we present weighting functions suitable for the characteristics of hybrid molecules:

5.3.1 Edge Weight Mapping. The weight w_e of an edge $e \in E$ is calculated by $f_{W_e}(e)$. We use the following strategies depending on the edge type in $\mathcal{G}_\delta$:

- a *structural-based* edge weight is a data design issue. It is set by the corpus expert to best suit the application. For instance, some applications might favor inter-document dependencies (such as references between documents) over other structural-based relations such as the *whole-part* relations between structural elements.
- a *domain-specific* edge weight uses the specificity measure to reflect the importance of domain-specific edge. The rationale is that the less incoming edges with the same label, the more important the edges become for a node. This measure is commonly used in the literature of semantic graphs [10, 13].
- a *hybrid* edge weight sets the importance of a link between two different nodes: a domain-specific node describing the content of a structural-based node. The value of a domain-specific node can be perceived as a term and the structural-based node as a document in a TF-IDF like notion [13]. The rationale is that the weight value is directly proportional to the number of occurrences of the domain-specific information contained in a structural element and inversely proportional to the number of occurrences of the domain-specific information contained in other structural elements.

Based on the above strategies, the edge weight mapping $f_{W_e}(e_{ij})$ (line 11) that assigns a weight $w_{e_{ij}} \in W$ to an edge $e_{ij} \in E$ connecting a node $v_i \in V$ to a node $v_j \in V$, is defined as follows:

$$f_{W_e}(e_{ij}) = \begin{cases} static(e_{ij}) \in [0,1], & \text{if } e_{ij} = e_s \\ \frac{1}{fan\text{-}in_{lab}(v_j)} \in]0,1], & \text{if } e_{ij} = e_d \\ TF(s_i, v_j) \times IDF(s_i, V_s) \in [0,1[, & \text{if } e_{ij} = e_h \end{cases}$$

where:

- $static(e_{ij})$ assigns a static weight w_{e_s} to a structural-based edge $e_s \in E_s$ based on prior knowledge provided by the corpus expert.
- $\frac{1}{fan\text{-}in_{lab}(v_j)}$ assigns a weight w_{e_d} to a domain-specific edge $e_d \in E_d$ based on the specificity of the edge, such that:
  - $fan - in_{lab}(v_j)$ is the number of incoming edges toward v_j having the same label lab of e_{ij}, where $lab = f_{Lab}(e_{ij})$ (See Definition 3).
- $TF(s_i, v_j) \times IDF(s_i, V_s)$ assigns a weight w_{e_h} to a hybrid edge $e_h \in E_h$, such that:
  - $TF(s_i, v_j)$ is the frequency of the value s_i of the domain-specific node $v_i \in V_d$ occurring in a structural-based node $v_j \in V_s$ (e.g., a document or a part of a document), where $s_i = f_{Val}(v_i)$ (See Definition 3).
  - $IDF(s_i, V_s)$ is the inverse frequency of the value s_i of the domain-specific node $v_i \in V_d$ occurring in the set of all structural-based nodes V_s.

5.3.2 Node Weight Mapping. The weight w_v of a node $v \in V$ is calculated by $f_{W_v}(v)$ based on its final activation value i.e., $f_{W_v}(v) = Activation(v) \in [0,1], \forall v \in V$.

5.3.3 Molecule Weight Mapping. The weight w_m of a molecule $m \in M$ is calculated by $f_{W_m}(m)$ based on the weights of its nodes:

- $f_{W_m}(m) = \frac{\alpha \times \sum w_{v_s} + \beta \times \sum w_{v_d}}{|V_m|} \in [0,1]$, where:
  - w_{v_s} and w_{v_d} are the weights computed by $f_{W_v}(v)$ and assigned to a structural-based node $v_s \in V_m$ and a domain-specific node $v_d \in V_m$ respectively.
  - $|V_m|$ is the total number of nodes in m.
  - α and β are the weight parameters that balance the contribution of the structural and domain-specific parts of m, such that α and $\beta \in [0,1]$.

Note that the molecule weight mapping $f_{W_m}(m)$ is used in HM_CSA by the *scoreAndRankMolecules* function (line 34) which computes a weight w_m for each hybrid molecule-based answer $m \in M_{out}$. The weights of molecules are then ranked in descending order.

6 EXPERIMENTAL STUDY

We conducted experiments on real-world implementation of our approach in the construction industry within FEED2SEARCH framework. The main purpose of the experiments is to validate that our approach can provide relevant hybrid molecules using HM_CSA w.r.t. the user queries. In this paper, we will present the results related to the effectiveness of HM_CSA.

6.1 Experimental Configuration

6.1.1 Queries. We collected 25 queries from an Institute for the Energy Transition of the building based on frequently required information searched by actors with different expertise (architects, technicians and engineers) throughout the different stages of real world construction projects. We divided the queries into two groups: Query Group 1 ($q_1 \rightarrow q_{12}$) for simple queries (firing start nodes of 1 concept type) and Query Group 2 ($q_{13} \rightarrow q_{25}$) for more diverse queries (firing start nodes of 2 or more concept types).

6.1.2 Tightly Coupled Semantic Graph. We generated a dataset of 30 000 RDF triples[10] over a corpus of 15 heterogeneous interdependent documents (total corpus size of 112MB) provided by the construction industry (See Table 1). We relied on the ifcOWL[11] ontology to create an adapted domain-specific ontology (built on OWL 2 in the Protégé environment and serialized in RDF/XML). We relied on Java-based libraries[12] and tailored XSLT processors to automatically generate the RDF instances graph.

[10] We used Resource Description Framework (https://www.w3.org/RDF/) to represent our instances graph to take advantage of its simplicity and modularity features.
[11] Available at http://www.buildingsmart-tech.org/ifcOWL/IFC4
[12] Mainly GATE 8.1, Apache Jena 3.2.0, Apache Tika 1.14 Toolkit and OxGarage.

6.1.3 Metrics. We use the following metrics: (i) Precision P to identify the number of relevant hybrid molecules among the retrieved results, (ii) Recall R to identify the number of relevant hybrid molecules that are retrieved among the total number of expected relevant results, (iii) F_1score to evaluate the harmonic mean of P and R, and (iv) Mean Average Precision MAP to evaluate the ranking of relevant results. These metrics are widely adopted in IR. They are detailed in [9]. Note that, for the assessment of the relevance of a molecule used in the calculation of the above metrics, we relied on users' judgments. For each query, we asked 12 users (who did not take part in the queries formulation), highly involved in the construction project from which the documents were taken, to provide a score for each answer (1 for relevant and 0 for not relevant) independently from each other. Afterwards, the users were asked to validate the judgments collectively. These users also provided the false negative hybrid molecules[13] for each query.

6.1.4 Implementation. Our query processing prototype is implemented in Java using Apache Jena 3.2.0 (for Ontology and RDF graph manipulation) and GATE 8.1 APIs (for NLP and Semantic Annotation of queries). It also implements the proposed HM_CSA.

6.2 Experiment 1: *Precision (P), Recall (R)* and F_1 *scores*

We first evaluated the effectiveness of HM_CSA for the 25 given queries in terms of P, R and F_1score. We studied the impact of constraint parameters, mainly the firing threshold F and the maximum spread distance D, on the query answers. To do so, we considered 3 different values for F (0.1, 0.3, and 0.5) and 4 different values for D (2,4,6, and 8). This resulted in 12 run configurations for each query execution. We further examined the influence of the diversity of the queries considering the two groups. Figure 4 shows the average values of P, R, and F_1score per run configuration per query group.

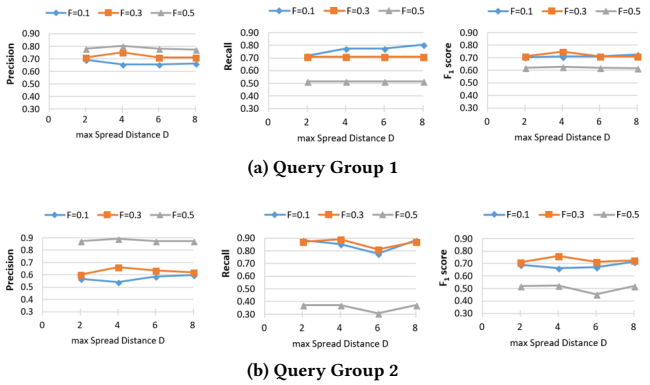
We select the optimal values of constraint parameters based on the results of the F_1score. Figure 4 shows that the highest average values of F_1score for both Query Group 1 (See Figure 4a) and Query Group 2 (See Figure 4b) are attained with $F = 0.3$ and $D = 4$ ($F_1score \simeq 0.75$). The optimal values of the constraint parameters portray a trade off between P and R. For instance, high precision is achieved with higher F values as only the most relevant nodes (with very high activation values) are selected. However, a high F value restricts the spread of the activations in the graph resulting in lower recall values. We also notice that, with the optimal constraint parameters ($F = 0.3$ and $D = 4$), HM_CSA attains slightly higher average precision and lower average recall with Query Group 1 when compared to Query Group 2. This is because increasing the concept types of the start nodes ensures that larger portion of the graph is searched but at the cost of increased false positive results.

6.3 Experiment 2: *Mean Average Precision (MAP)* values

We also evaluated the ranking of the hybrid molecule-based query answers considering the optimal constraint parameters of HM_CSA ($F = 0.3$ and $D = 4$ from the previous experiment) over the same two groups of queries. We varied α and β parameters (See Section 5.3) and studied their impact on the MAP values of HM_CSA. We chose 3 configurations: (i) $\alpha = 0$ and $\beta = 1$, (ii) $\alpha = 1$ and $\beta = 0$, and (iii) $\alpha = 1$ and $\beta = 1$ to emphasize respectively domain-specific nodes' contribution, structural-based nodes' contribution and the equal contribution of both in the overall weight of a hybrid molecule. These contributions also reflect the impact of the different strategies adopted for the weight mapping (See Section 5.3). Figure 5 shows average values of MAP per configuration per group.

The results show that considering only the contribution of domain-specific nodes (i.e., $\alpha = 0$, $\beta = 1$), Query Group 2 attains a higher average MAP value in comparison to the result obtained when considering only the contribution of structural-based nodes (i.e., $\alpha = 1$, $\beta = 0$). Query Group 1 shows an opposite behavior as it has less concept types for starting nodes, thus it is less influenced by the domain-specific contribution. The highest values of MAP

[13] For the sake of simplicity, they only pointed missing hybrid molecules' cores.



(a) Query Group 1

(b) Query Group 2

Figure 4: Average *Precision (P), Recall (R),* and F_1score of HM_CSA considering different values of threshold F and maximum spread distance D for (a) Query Group 1 and (b) Query Group 2.

Table 1: Heterogeneous document corpus.

Format	Content Description
5 docx	General Technical Specifications
	Electrical Specifications
	Exterior Facades and Carpentry
	Thermal Properties
	Acoustic Properties
7 pdf	Electrical Drawing
	HVAC Drawing
	Wall Composition
	Confort Analysis
	Environmental Impacts
	Environmental and Energy Impacts
	Carpentry and Glazing
1 xlsx	Thermal Regulations
2 png	Material Pattern Photo
	Sealing Test

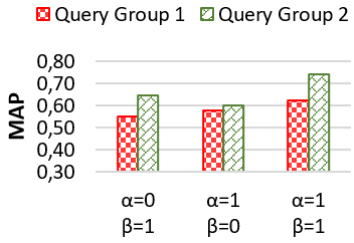


Figure 5: Average *MAP* values of HM_CSA per α and β configuration per Query Group.

(*MAP*= 0.62 for Query Group 1, *MAP*= 0.74 for Query Group 2) are reached when taking into account both the structural-based and domain-specific aspects in the weight calculation of the molecules (i.e., $\alpha = 1$, $\beta = 1$). This further highlights the importance of the hybrid aspect of the molecule.

To sum up, in the context of the given heterogeneous document corpus, the two experiments validate that, using optimal constraint and weight parameters, HM_CSA reaches an overall $F_1 score$ of 0.75 and average *MAP* values > 0.6, which is a promising result in IR. This demonstrates that HM_CSA is capable of providing relevant query answers in the form of hybrid molecules i.e., the augmented contextualized results satisfy the user's needs.

7 RELATED WORK

Molecules, mainly RDF molecules, have received a wide attention over the past two decades within different research areas. It is first introduced by Ding et al. [5] for the purpose of tracking RDF provenance and evaluating trustworthiness against RDF data in semantic Web applications. They define it as the finest and lossless connected subgraph decomposition of the original RDF graph based on Functional Properties (FP) and Inverse Functional Properties (IFP), which are specified in a background ontology. Endris et al. [6] use RDF Molecule Templates (RDF-MTs) in the context of federated SPARQL query processing over RDF datasets. They define a molecule as a set of triples sharing the same subject. Their RDF-MTs model the structure of the data and guide the query decomposition. In our work, we also perceive molecules as a graph decomposition into subgraphs. Yet, we introduce hybrid molecules, independently of the serialization technology, and adjust the decomposition to enrich the output of current IR approaches over heterogeneous data.

In IR, many research works have attested the benefits of incorporating knowledge bases and domain specific ontologies in their index, query, and search process [8] to overcome the semantic gap between keywords found in the documents and those in the user's query [7]. This have been rapidly migrating into industrial applications [16]. The main advantage resides in maximizing the precision and recall w.r.t. to traditional IR [9] where the search is limited to syntactic keyword matching. Tekli et al. [13] build upon the idea of semantic aware search to target textual databases. They propose a tightly coupled inverted index graph by combining a semantic network and a standard inverted index. In contrast, other works used tightly coupled semantic graphs to describe concepts appearing in heterogeneous document collections [4, 10]. Likewise, we leverage information semantics and domain-specific ontologies in an IR approach applied on tightly coupled semantic graphs describing heterogeneous document corpora. However, we differ from the existing approaches by further providing augmented results describing the document structure and the inter and intra-document links, which enriches the search results with helpful information.

Among the many heuristic graph-based search methods [15], CSA has been widely adopted in IR applications where it proved its effectiveness [1, 10]. Cohen et al. [17] use CSA algorithm to realize intelligent matches between user requirements and relevant agents in a Q&A application. Crestani et al. [14] were the first to apply CSA technique to the World Wide Web to retrieve information using an ostensive approach to querying similar to query-by-example. Sun et al. [11] combine CSA algorithm with a spatial ontology to improve results in associative retrieval of spatial big data. One of the most prominent uses of CSA is the one proposed by Rocha et al. [10]. It consists of the combination of CSA techniques with traditional search engine techniques for searching in the Semantic Web. Although our approach is also inspired by CSA, it searches for relevant hybrid molecules instead of single nodes. We also differ from existing CSA-based approaches in the different strategies adopted for the weight mapping to cope with the characteristics of hybrid molecules and provide effective ranking.

8 CONCLUSION

This paper introduces hybrid molecules as novel augmented query answers for IR systems over heterogeneous document corpora. The hybrid molecules are presented within a novel generic framework: FEED2SEARCH. They are generated through HM_CSA algorithm, which is built on the Constrained Spread Activation (CSA). They provide helpful structural and domain-specific contextual information. Experiments conducted on projects in the construction industry show promising real-world results.

These findings and feedback from users motivate us to further evaluate our research in other application domains. We are currently implementing our work in the medical domain, which also involves heterogeneous dependent data and would benefit from augmented query answers. In the future, we plan to evaluate the richness of the hybrid molecule-based query answers w.r.t. the state-of-art in IR in order to quantify the impact of the contextual information of the results on the user's experience, especially regarding the time needed by the users to interpret and track the results. Additionally, we propose the following future improvements to achieve even higher effectiveness values: (i) consider more strategies for firing start nodes in HM_CSA (e.g., adopting advanced disambiguation techniques [12] beforehand, in the query interpretation stage), and (ii) investigate alternative weight mapping functions.

REFERENCES

[1] Mangold Christoph. 2007. A survey and classification of semantic search approaches. *Int. J. Metadata, Semantics and Ontology* 2, 1 (2007), 23–34.
[2] Beetz Michael et al. 2015. Open-ease. In *ICRA*. IEEE, 1983–1990.
[3] Cunningham Hamish et al. 2002. A framework and graphical development environment for robust NLP tools and applications.. In *ACL*. 168–175.
[4] Charbel Nathalie et al. 2017. LinkedMDR: A Collective Knowledge Representation of a Heterogeneous Document Corpus. In *DEXA*. Springer, 362–377.
[5] Ding Li et al. 2005. Tracking rdf graph provenance using rdf molecules. In *ISWC (Poster)*. 42.
[6] Endris Kemele et al. 2017. MULDER: Querying the Linked Data Web by Bridging RDF Molecule Templates. In *DEXA*. Springer, 3–18.
[7] Giunchiglia Fausto et al. 2009. Concept search. In *ESWC*. Springer, 429–444.
[8] Kiryakov et al. 2004. Semantic annotation, indexing, and retrieval. *JWS* 2, 1 (2004), 49–79.
[9] Manning Christopher D. et al. 2008. *Introduction to information retrieval*. Cambridge University Press.
[10] Rocha Cristiano et al. 2004. A hybrid approach for searching in the semantic web. In *WWW*. ACM, 374–383.
[11] Sun Shengtao et al. 2015. A spreading activation algorithm of spatial big data retrieval based on the spatial ontology model. *Cluster Comput* 18, 2 (2015), 563–575.
[12] Tekli Joe et al. 2016. Building semantic trees from XML documents. *JWS* 37 (2016), 1–24.
[13] Tekli Joe et al. 2018. Full-fledged semantic indexing and querying model designed for seamless integration in legacy RDBMS. *DKE* (2018).
[14] Crestani Fabio and Lee Puay Leng. 2000. Searching the web by constrained spreading activation. *INFORM PROCESS MANAG* 36, 4 (2000), 585–605.
[15] Russell Stuart J. and Norvig Peter. 2016. *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,.
[16] Ontotext. 2018. Semantic Technologies for Smarter Information Retrieval and Content Management. https://ontotext.com/semantic-solutions/.
[17] Cohen Paul R. and Kjeldsen Rick. 1987. Information retrieval by constrained spreading activation in semantic networks. *INFORM PROCESS MANAG* 23, 4 (1987), 255–268.