CHƯƠNG 4
CÁC VẤN ĐỀ VÀ THUẬT TOÁN

Trong chương này, chúng ta sẽ thiết kế các xử lý cơ bản trên các đối tượng như keyphrase, đồ thị keyphrase biểu diễn tài liệu và câu truy vấn, ontology, cơ sở dữ liệu, hệ thống tập tin và kho tài liệu. Đề xuất một số phương pháp và kỹ thuật điều khiển giúp tính toán độ tương đồng về ngữ nghĩa giữa các keyphrase, so khớp đồ thị keyphrase, đo lường mức độ tương quan ngữ nghĩa giữa tài liệu và câu truy vấn, xây dựng đồ thị keyphrase cho mỗi tài liệu, xử lý câu truy vấn người dùng và tìm kiếm theo ngữ nghĩa các tài liệu. Từ đó làm cơ sở cho việc xây dựng các động cơ suy diễn và tìm kiếm trong hệ thống quản lý kho tài nguyên nói chung và quản lý kho tài liệu học tập lĩnh vực CNTT nói riêng. Ứng dụng này sẽ được xây dựng và trình bày trong chương sau.

Dựa trên mô hình biểu diễn tri thức, biểu diễn tài liệu, mô hình tổ chức lưu trữ kho tài liệu theo ngữ nghĩa (như đã giới thiệu trong chương 3), ta xây dựng một số thuật giải cùng với những xử lý cơ bản nhằm giải quyết các vấn đề chính đặt ra như sau:

Vấn đề 1: So khớp đồ thị keyphrase, trên cơ sở đó đo lường mức độ liên quan giữa tài liệu và câu truy vấn.

Cho trước một ontology CK_ONTO và hai đồ thị keyphrase biểu diễn tài liệu hay câu truy vấn. Yêu cầu thực hiện tính toán độ tương quan về ngữ nghĩa giữa hai đồ thị. Ý tưởng cơ bản là tìm ra độ đo sự tương đồng, sự giống nhau về ngữ nghĩa giữa các đỉnh keyphrase và giữa các đỉnh quan hệ có trong hai đồ thị.

Vấn đề 2: Xây dựng đồ thị keyphrase biểu diễn ngữ nghĩa cho tài liệu.

Từ một tập tin tài liệu bất kỳ cùng với các thông tin mô tả (siêu dữ liệu) kèm theo nếu có, thực hiện việc rút trích các keyphrase đặc trưng của tài liệu và biểu diễn (nội dung) tài liệu thành đồ thị keyphrase tương ứng.

Vấn đề 3: Xử lý câu truy vấn: tương tự như quá trình xử lý tài liệu bao gồm thao tác rút trích tự động keyphrase và thiết lập đồ thị keyphrase cho câu truy vấn.

Vấn đề 4: Bài toán tìm kiếm theo ngữ nghĩa các tài liệu.

Từ câu truy vấn người dùng nhập vào, hệ thống tìm kiếm và trả về danh sách các tài liệu (được sắp hạng) có nội dung liên quan và phù hợp với thông tin truy vấn. Những tài liệu này không nhất thiết phải chứa chính xác từ khóa tìm kiếm. Giải pháp là sử dụng một hàm so khớp đồ thị keyphrase biểu diễn câu hỏi với các đồ thị keyphrase biểu diễn tài liệu để đánh giá độ tương quan về ngữ nghĩa của các tài liệu với câu truy vấn.

Vấn đề 5: Xác định thư mục lưu trữ cho một tài liệu mới cập nhật vào kho, nghĩa là xác định lĩnh vực hay chủ đề mà nội dung tài liệu đề cập đến và gán tài liệu vào thư mục lưu trữ tương ứng với chủ đề đó.

4.1. SO KHỚP ĐỒ THỊ KEYPHRASE VÀ ĐO LƯỜNG MỨC ĐỘ TƯƠNG QUAN VỀ NGỮ NGHĨA

Như đã giới thiệu trong chương trước, có nhiều phương pháp tính độ đo khoảng cách ngữ nghĩa giữa các khái niệm đã được đề xuất. Các nghiên cứu này tập trung chủ yếu vào các hướng tiếp cận chính như dựa trên kho ngữ liệu, dựa trên ontology hay phương pháp lai ghép hai cách tiếp cận trên bằng cách kết hợp tri thức của một ontology với các ước lượng xác suất tìm được từ kho ngữ liệu.

Hướng tiếp cận dựa trên kho ngữ liệu mặc dù được hỗ trợ bởi các công cụ toán học mạnh mẽ nhưng vẫn có những thiếu sót trong việc xử lý một số khía cạnh sâu hơn của ngôn ngữ, cụ thể là mối liên hệ về mặt ngữ nghĩa khác nhau giữa các từ lại không được xét đến. Hầu hết các kho ngữ liệu có sẵn chưa được gán nhãn từ loại do đó không xác định được độ liên quan giữa các nghĩa của từ dẫn đến hậu quả là các quan hệ giữa các nghĩa của từ có tần suất thấp sẽ không được xem xét trong các phương pháp thống kê. Một vấn đề nghiêm trọng khác là tính thiếu đầy đủ, thậm chí ngay cả trong những kho ngữ liệu lớn.

Hướng tiếp cận dựa trên ontology được xem là một phương pháp giàu ngữ

nghĩa hơn, trong đó sử dụng tất cả các tri thức ngữ nghĩa được định nghĩa trước. Tuy nhiên, cách tiếp cận này cũng vẫn còn mắc phải nhiều hạn chế do quá phụ thuộc vào những tài nguyên từ vựng vốn được xây dựng một cách thủ công theo ý kiến chủ quan của con người nên dễ dẫn tới nhiều trường hợp thiếu sót hay dư thừa từ vựng trong miền tri thức khảo sát. Ngoài ra, tiêu chuẩn phân loại, phân lớp các từ có thể không rõ ràng, cách phân loại kém và không cung cấp đủ sự phân biệt giữa các từ và trên hết là đòi hỏi nhiều công sức của con người nhằm tạo ra danh sách lớn các từ đồng nghĩa, gần nghĩa, các quan hệ phân cấp hay có liên quan khác một cách thủ công. Tuy nhiên, cách tiếp cận dựa trên các ontology được xem là cách tiếp cận hiện đại và phù hợp nhất cho biểu diễn và xử lý ngữ nghĩa, các tài nguyên tri thức của ontology vẫn là những tài nguyên hết sức có giá trị. Nếu những tài nguyên từ vựng hay các ontology được xây dựng tốt, mô tả được tương đối đầy đủ tri thức của lĩnh vực thì việc sử chúng sẽ làm tăng độ chính xác và khả năng vét cạn trong quá trình tính toán các độ đo ngữ nghĩa cũng như tìm kiếm thông tin. Hơn nữa, các độ đo khoảng cách ngữ nghĩa giữa các từ của cách tiếp cận dựa trên ontology thì đơn giản, trực quan và dễ hiểu hơn.

Hướng tiếp cận lai ghép dựa trên lý thuyết thông tin: đây là phương pháp lai ghép giữa khảo sát dựa trên kho ngữ liệu và các ontology bằng cách dựa trên sự kết hợp cấu trúc phân loại từ vựng với thông tin thống kê (các ước lượng xác suất) có từ kho ngữ liệu. Hướng tiếp cận này sử dụng khái niệm "lượng tin" trong lý thuyết thông tin. Mục tiêu là khắc phục tính không ổn định của các khoảng cách liên kết các khái niệm đã xuất hiện trong hướng tiếp cận dựa trên ontology, bằng cách bổ sung vào các thông số chuẩn hóa của lý thuyết thông tin. Tuy nhiên với việc dùng một kho ngữ liệu để tính ra các giá trị lượng tin sẽ thừa hưởng tất cả những thiếu sót của phương pháp tiếp cận dựa trên kho ngữ liệu chẳng hạn như vấn đề dữ liệu rải rác thiếu tập trung, vấn đề cần thiết kho ngữ liệu gán nhãn ngữ nghĩa và cú pháp.

Nhìn chung, các hướng tiếp cận trong việc tính toán độ đo tương tự ngữ nghĩa giữa các khái niệm của các công trình nghiên cứu trước đây vẫn chưa đưa ra được một

độ đo có xét đến nhiều mối quan hệ ngữ nghĩa khác nhau giữa các khái niệm. Hầu hết các phương pháp dựa trên mạng phân cấp ngữ nghĩa đều sử dụng WordNet - một ontology tổng quát - để thực hiện việc nghiên cứu. Theo đó, khoảng cách ngữ nghĩa giữa hai khái niệm chỉ được tính dựa trên thông tin về cạnh hay nút dọc theo đường nối giữa chúng và liên kết giữa hai khái niệm bất kỳ chỉ biểu diễn cho mối quan hệ phân cấp is-a trong WordNet. Tuy nhiên, đối với từng lĩnh vực hay miền tri thức khác nhau thì sẽ tồn tại nhiều mối quan hệ ngữ nghĩa khác nhau. Hơn nữa, khoảng cách ngữ nghĩa giữa hai khái niệm không chỉ phụ thuộc vào số nút hay cạnh trong đường nối giữa chúng mà còn phụ thuộc vào những quan hệ nào được sử dụng để liên kết các khái niệm với nhau vì có những liên kết có thể thể hiện một khác biệt lớn về nghĩa trong khi có các liên kết khác chỉ có sự phân biệt rất nhỏ.

Dựa trên ý tưởng trong cách tiếp cận của D.Gennest và M.Chein [11] chúng tôi đã đưa ra với một số biến đổi và đề xuất cải tiến nhằm xây dựng một mô hình tính toán độ tương tự về ngữ nghĩa giữa các keyphrase và giữa các quan hệ trên keyphrase dựa trên việc khai thác nguồn tri thức ontology CK_ONTO, trên cơ sở đó xây dựng công thức tính độ tương quan về ngữ nghĩa giữa hai đồ thị keyphrase biểu diễn nội dung văn bản cùng với một số thuật toán so khớp tương ứng.

4.1.1. Tính toán và so khớp các đồ thị keyphrase

Việc giải quyết bài toán so trùng các đồ thị keyphrase là tìm ra các độ đo về mặt ngữ nghĩa giữa hai đồ thị. Đồ thị keyphrase bao gồm các keyphrase và quan hệ tạo thành, nên phương hướng để thực hiện việc đo độ giống nhau về ngữ nghĩa giữa hai đồ thị là tìm ra độ đo tương tự ngữ nghĩa giữa các keyphrase và giữa các quan hệ có trong hai đồ thị đó.

Xét hai hàm: $\alpha : K \times K \to [0,1]$ và $\beta : R_{KK} \times R_{KK} \to [0,1]$ dùng để đo sự giống nhau, tương đồng nhau về ngữ nghĩa giữa hai keyphrase và hai quan hệ. Giá trị 1 sẽ đại diện cho sự bằng nhau, tương đương về nghĩa giữa hai đối tượng và giá trị 0 tương ứng với không có bất kỳ liên kết ngữ nghĩa nào giữa chúng. Trên thực tế, khó có thể đạt được

một giá trị có độ chính xác cao bởi vì ngữ nghĩa chỉ được hiểu đầy đủ khi được xét trong một ngữ cảnh xác định.

4.1.1.1. Xác định α và β

Hàm β có thể được xác định tùy ý (không bằng một công thức hay quy tắc tính nhất định) bằng một bảng giá trị tương ứng giữa các cặp r, r' $\in R_{KK}$. Do số quan hệ giữa các keyphrase được định nghĩa là không nhiều nên ta có thể xác định hàm β theo phương pháp liệt kê từng giá trị cụ thể. Ví dụ:

$\beta(r_9, r_{10}) = 0.8, \quad \beta(r_{11}, r_{17}) = 0.7 \qquad (r_9 : cause, r_{10} : influence, r_{11} : instrument, r_{17} : support)$. Tuy nhiên, cho dù sự xác định này là tùy ý, nhưng do đặc thù của những quan hệ ngữ nghĩa được chọn, một vài ràng buộc đặt ra như sau:

$\forall r \in R_{KK}, \beta(r, r) = 1$

$\forall r, r' \in R_{KK}, \beta(r, r') = \beta(r', r)$

Định nghĩa: Cho k, k' $\in$ K, ta định nghĩa một quan hệ hai ngôi P trên K, gọi là quan hệ "tồn tại một **dẫn xuất** từ k đến k'" như sau: P (k,k') khi và chỉ khi k = k' hoặc tồn tại $S = (s_1, s_2, ..., s_n)$ là dãy các số nguyên $\in$ [1, t] (với t = $|R_{KK}|$) sao cho $k\, r_{s_1}\, k_1,\ k_1\, r_{s_2}\, k_2,\ ...,\ k_{n-1}\, r_{s_n}\, k'$, khi đó k' được gọi là một dẫn xuất của k và k, k' có liên kết ngữ nghĩa với nhau.

Ta gọi $k\, r_{s_1}\, k_1,\ k_1\, r_{s_2}\, k_2,\ ...,\ k_{n-1}\, r_{s_n}\, k'$ là một dãy dẫn xuất (hay một đường nối, đường đi) từ k đến k'. Số quan hệ được dùng để liên kết các keyphrase trong dãy là độ dài (chiều dài) của dãy.

Hàm α được định nghĩa như sau:

$\alpha(k, k') = 0\ if\ notP(k, k')$ // không có bất kỳ liên kết ngữ nghĩa nào giữa k và k'

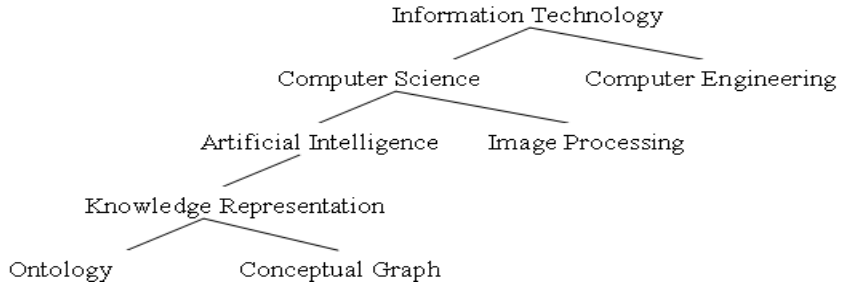
$\alpha(k, k') = Max\{V(k\, r_{s_1}\, k_1,\ k_1\, r_{s_2}\, k_2,\ ...,\ k_{n-1}\, r_{s_n}\, k')\}$ nếu tồn tại một dãy dẫn xuất $k\, r_{s_1}\, k_1,\ k_1\, r_{s_2}\, k_2,\ ...,\ k_{n-1}\, r_{s_n}\, k'$ từ k đến k'.

Hàm V được cho bởi công thức:

$$V(k\, r_{s_1}\, k_1,\ k_1\, r_{s_2}\, k_2,\ \ldots,\ k_{n-1}\, r_{s_n}\, k') = \prod_{1}^{n} val\_r_{s_i}(k_{i-1}, k_i) \qquad (k_n \equiv k')$$

trong đó, 0< $val\_r_{s_i}(k_{i-1}, k_i)$<1 là trọng số được gán cho mỗi quan hệ r_{s_i} tính trên cặp keyphrase k_{i-1}, k_i phản ánh độ đo tương đồng ngữ nghĩa giữa hai keyphrase này. Khi đó, giá trị của V và α nằm trong khoảng từ 0 đến 1.

Hàm V cho phép đánh giá sự kết hợp giữa những quan hệ ngữ nghĩa được dùng trong dãy dẫn xuất. Sự đánh giá này là cần thiết do sự tương đồng về ngữ nghĩa giữa hai keyphrase được liên kết với nhau bởi một quan hệ ngữ nghĩa có thể khác nhau tùy thuộc vào quan hệ nào được sử dụng, hay nói cách khác là khoảng cách ngữ nghĩa giữa hai keyphrase phụ thuộc vào các mối quan hệ khác nhau liên kết giữa chúng, trong đó có những liên kết thể hiện một khác biệt lớn về nghĩa trong khi có các liên kết khác chỉ có sự phân biệt rất nhỏ. Ví dụ, những keyphrase được liên kết bởi quan hệ đồng nghĩa thì giống nhau về nghĩa hơn là những keyphrase được liên kết bởi nhóm quan hệ phân cấp. Hơn nữa mức độ tương đồng về nghĩa khi xét trên một quan hệ r_{S_i} bất kỳ cũng khác nhau tùy theo cặp keyphrase nào được liên kết. Ví dụ, khi xét quan hệ phân cấp (thể hiện trên mạng phân cấp ngữ nghĩa), các liên kết nằm ở mức cao trong phép phân loại (gần với nút gốc) thường thể hiện khoảng cách ngữ nghĩa lớn hơn, các liên kết ở mức thấp thể hiện khoảng cách ngữ nghĩa nhỏ hơn, gần nghĩa nhau hơn. Cụ thể trong mạng phân cấp hình 4.1, khoảng cách ngữ nghĩa giữa Computer Science với Artificial Intelligence thì lớn hơn so với Knowledge Representation với Ontology.



Hình 4.1: Ví dụ về quan hệ phân cấp của Information Technology

Giá trị của V ứng với dãy dẫn xuất từ k đến k’ càng lớn thì độ tương tự về ngữ

nghĩa giữa hai keyphrase càng lớn (khoảng cách ngữ nghĩa càng nhỏ) và ngược lại. Trong trường hợp tồn tại nhiều dãy dẫn xuất khác nhau liên kết giữa hai keyphrase, độ đo tương đồng ngữ nghĩa giữa hai keyphrase chính là giá trị lớn nhất của V.

Khoảng cách ngữ nghĩa giữa các keyphrase phụ thuộc chặt chẽ vào ngữ nghĩa (hay sự khác biệt về nghĩa) của các quan hệ liên kết chúng. Ngữ nghĩa của những quan hệ này cho ta một số điều kiện ràng buộc độc lập với các biểu thức hàm như sau:

1). $\forall k \in K, \alpha(k,k) = 1$

2). $\forall k, k' \in K, \forall r \in R_{KK}, if\ k\ r\ k'\ then\ \alpha(k,k') \neq 0$

3). $\forall k, k' \in K, \alpha(k,k') = \alpha(k',k)$

$\forall k_1, k_2, k_3, k_4, k_5, k_6 \in K,$

4). $if\ k_1\ r_i\ k_2 and\ k_3\ r_j\ k_4\ and\ k_5\ r_t\ k_6, i \in \{1,2,3\}, j \in \{4,5\}, t \in \{6,7,...,25\}\ then$
$\alpha(k_1,k_2) > \alpha(k_3,k_4) > \alpha(k_5,k_6)$

nghĩa là, những keyphrase có quan hệ thuộc nhóm quan hệ tương đương sẽ có độ tương đồng về ngữ nghĩa lớn hơn so với những keyphrase có quan hệ phân cấp, nhỏ nhất là nhóm quan hệ không phân cấp.

5). $if\ k_1\ r_4\ k_2\ and\ k_3\ r_5\ k_4\ then\ \ \alpha(k_3,k_4) > \alpha(k_1,k_2)$

6). $if\ k_1\ r_1\ k_2\ and\ k_3\ r_2\ k_4\ and\ k_5\ r_3\ k_6\ \ then\ \ \alpha(k_3,k_4) > \alpha(k_1,k_2) > \alpha(k_5,k_6)$

7). $\forall k_1, k_2 \in K, if\ k_1\ r_1\ k_2\ or\ k_1\ r_2\ k_2\ then\ \ \alpha(k_1,k_2) \cong 1$

Việc xác định giá trị của $val\_r_{s_i}(k_{i-1,}k_i)$ được thực hiện dựa trên phương pháp chuyên gia. Mỗi quan hệ r_i sẽ được gắn một trọng số có giá trị nằm trong khoảng $[\min_{R_i}, \max_{R_i}]$ (tùy thuộc vào cặp keyphrase nào được liên kết) và thỏa các ràng buộc trên như sau:

Bảng 4.1 Trọng số được gán cho mỗi quan hệ

	Quan hệ ngữ nghĩa	$[\min_{R_i}, \max_{R_i}]$
r_1	Synonym	[0.95, 0.99]

r_2	Acronym	[0.95, 0.99]
r_3	Near synonym	[0.9, 0.94]
r_4	A part of	[0.8, 0.84]
r_5	A kind of	[0.85, 0.89]
r_6	Extension	[0.75,0.79]
r_7	Same class	[0.75,0.79]
r_8	Relation	[0.7,0.74]
r_9	Cause	[0.65, 0.69]
r_{10}	Influence	[0.65, 0.69]
r_{11}	Instrument	[0.65, 0.69]
r_{12}	Make	[0.65, 0.69]
r_{13}	Possession	[0.65, 0.69]
r_{14}	Source	[0.65, 0.69]
r_{15}	Aim	[0.65, 0.69]
r_{16}	Location	[0.65, 0.69]
r_{17}	Temporal	[0.65, 0.69]
r_{18}	Manner	[0.65, 0.69]
r_{19}	Support	[0.65, 0.69]
r_{20}	Beneficiary	[0.65, 0.69]
r_{21}	Property	[0.65, 0.69]
jr_{22}	Agent	[0.65, 0.69]
r_{23}	Circumstance	[0.65, 0.69]
r_{24}	Person	[0.65, 0.69]
r_{25}	Application	[0.65, 0.69]

Ví dụ: Dựa trên sơ đồ phân cấp hình 3.1, 4.1, ta có thể tính được các giá trị tương đồng ngữ nghĩa giữa các cặp keyphrase:

$$\alpha(artificial\ intelligence, conceptual\ graph) = val\_r_4(artificial\ intelligence, knowledge\ representation) * val\_r_4(knowledge\ representation, conceptual\ graph) = 0.8 * 0.84 = 0.672$$

$$\alpha(network, ISDN) = val\_r_4(network, internet\ access) * val\_r_4(internet\ access, Integrated\ Services\ Digital\ Network) * val\_r_2(ISDN, Integrated\ Services\ Digital\ Network) = 0.8 * 0.82 * 0.99 = 0.64944$$

4.1.1.2. So khớp đồ thị keyphrase

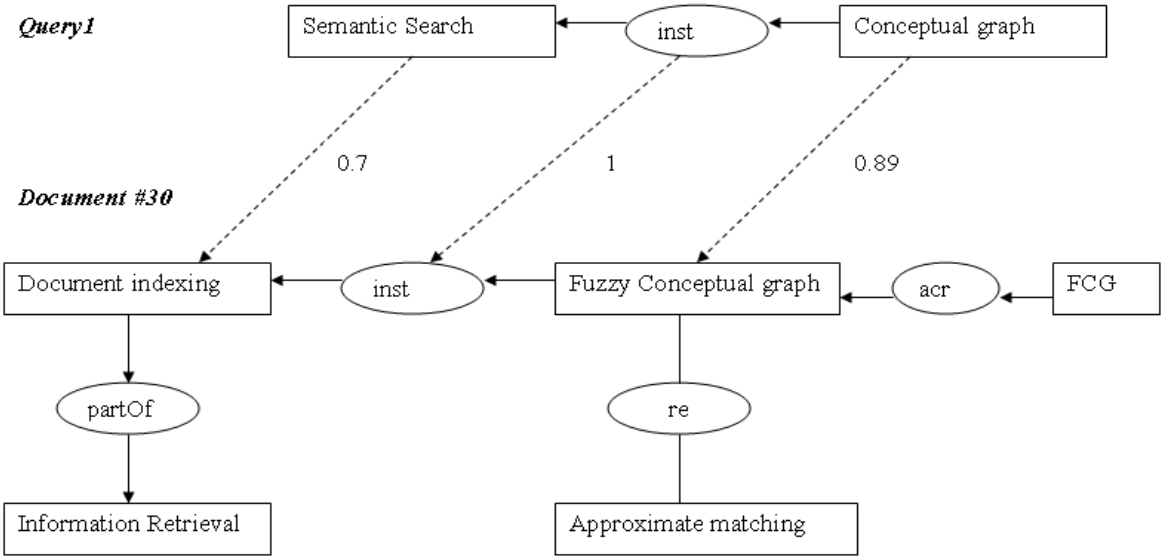
Định nghĩa: **Một phép chiếu** từ đồ thị keyphrase H = (KH, RH, EH) tới đồ thị keyphrase G = (KG, RG, EG) là một cặp có thứ tự $\Pi = (f, g)$ của 2 ánh xạ $f : RH \to RG, g : KH \to KG$ thỏa điều kiện:

- Đơn ánh
- Phép chiếu bảo toàn "quan hệ kề" giữa các đỉnh và cung, nghĩa là với mọi $r \in RH, g(adj_i(r)) = adj_i(f(r))$ với $1 \le i \le 2$. Trong đó, adj$_i$(r) là đỉnh keyphrase thứ i kề với đỉnh quan hệ r. Nếu hai đỉnh kề kề nhau trong H thì các đỉnh tương ứng của nó cũng kề nhau trong G.
- $r \in RH, \beta(r, f(r)) \ne 0$
- $k \in KH, \alpha(k, g(k)) \ne 0$

Định nghĩa: Một mô hình lượng giá cho phép chiếu từ đồ thị H đến đồ thị G được định nghĩa như sau (tỉ lệ về khoảng [0,1]):

$$v(\Pi) = \frac{\sum_{k \in KH} \alpha(k, g(k)) + \sum_{r \in RH} \beta(r, f(r))}{|KH| + |RH|}$$

Ví dụ: Khi thực hiện so khớp giữa 2 đồ thị keyphrase biểu diễn cho *Document #30* và câu truy vấn *Query 1,* ta được một phép chiếu $\prod$ (được xem là tốt nhất) tương ứng giữa hai đồ thị:



Giá trị của phép chiếu $\prod$ được tính: $v(\Pi) = \frac{0.7 + 0.89 + 1}{3} = 0.86$

Định nghĩa: Tồn tại một **phép chiếu bộ phận** từ đồ thị keyphrase H tới đồ thị keyphrase G nếu và chỉ nếu tồn tại một phép chiếu từ H', một đồ thị keyphrase con của H , tới G.

Mô hình lượng giá cho phép chiếu bộ phận $v(\Pi_{partial})$ chỉ phụ thuộc vào tập đỉnh của H' và được định nghĩa như $v(\Pi)$.

Nếu α, β được định nghĩa tốt thì mô hình lượng giá cho phép chiếu trên sẽ cung cấp cho ta một công thức so khớp giữa hai đồ thị. Độ tương quan ngữ nghĩa giữa hai đồ thị keyphrase là một giá trị thuộc khoảng [0,1] và được biểu diễn bởi công thức sau:

$$Rel(H,G) = Max\{v(\Pi) \mid \Pi \text{ là phép chiếu bộ phận từ H tới G}\}$$

Ví dụ: tương quan ngữ nghĩa giữa câu truy vấn *Query 1* và tài liệu *Document #30* được tính là Rel (Query 1, Document #30) = 0.89. Mặc dù phép chiếu $\prod$ có giá trị lớn nhất trong số các phép chiếu từ đồ thị Query 1 tới đồ thị biểu diễn tài liệu là 0.86, nhưng nếu xét trong không gian các phép chiếu bộ phận thì giá trị của Rel (Query 1, Document #30) được tính theo giá trị 0.89 của phép chiếu từ đồ thị con chỉ bao gồm một đỉnh keyphrase *Conceptual graph* tới đồ thị của Document #30.

4.1.2. Thuật toán tính độ tương đồng ngữ nghĩa giữa hai keyphrase

Bài toán được đặt ra như sau: Cho một ontology CK_ONTO (K, C, R_{KC}, $R_{CC,}$ R_{KK}, label) và hai keyphrase k_1, k_2. Yêu cầu tính giá trị $\alpha(k_1,k_2) \in [0,1]$ phản ánh độ đo sự tương tự nhau, giống nhau về ngữ nghĩa giữa hai đối tượng, giá trị này càng lớn thì sự giống nhau về nghĩa của chúng càng lớn và ngược lại. Ý tưởng cơ bản là sử dụng phương pháp lan truyền kết hợp với một số qui tắc heuristic (về độ ưu tiên của các quan hệ), trong đó qui tắc lan truyền chính là dò tìm các mối quan hệ ngữ nghĩa có thể có trên tập keyphrase đã được định nghĩa trong ontology so với keyphrase đã kích hoạt trước đó và sử dụng một hàng đợi ưu tiên theo tiêu chuẩn trọng số lớn nhất để lưu lại các đỉnh keyphrase đã kích hoạt theo qui tắc trên.

Input: Ontology CK_ONTO

Hai keyphrase k_1, k_2

Output: một giá trị $\alpha(k_1,k_2) \in [0, 1]$

Ghi nhận thông tin về nguồn tri thức ontology CK_ONTO, bao gồm các tập sau:

Keyphrases :={}; // tập các keyphrase

Classes := {}; // tập các lớp keyphrase

KC_Rela := {}; // quan hệ thuộc về giữa keyphrase và lớp.

CC_Hypo := {}; // quan hệ "phân cấp" trên lớp.

CC_Rela := {}; // quan hệ "có liên quan" giữa các lớp.

R_i := {}; // quan hệ r_i giữa các keyphrase với i = 1, 2, …25 tương ứng 25 quan hệ có độ ưu tiên (thứ tự dò tìm trên tập các quan hệ) giảm dần. Mỗi phần tử trong R_i là một bộ [keyphrase 1, keyphrase 2, val_r_i (keyphrase 1, keyphrase 2)]

minValR := [] // lưu giá trị $\min_{R_i}$ của 25 quan hệ trên keyphrase (bảng 4.1).

Các bước thực hiện:

Bước 1: Khởi tạo

Đặt trạng thái ban đầu cho một số biến điều khiển

KQueue: ={}; // hàng đợi ưu tiên

Threshold: = 0,5; // ngưỡng, khoảng cách ngữ nghĩa nhỏ nhất cho phép giữa những keyprhase.

Bước 2: Thêm vào hàng đợi keyphrase k_1 cùng với giá trị ưu tiên là 1

Queue.add(k_1, 1);

Bước 3: Thực hiện một quá trình dò tìm các keyphrase có quan hệ ngữ nghĩa với k_1 để phát sinh các keyphrase mới lưu vào hàng đợi.

while not (KQueue.empty())

<3.1> Lấy ra khỏi hàng đợi phần tử có độ ưu tiên lớn nhất (truy xuất và xóa phần tử có độ ưu tiên lớn nhất từ hàng đợi)

(key, val) : = KQueue.dequeue();

<3.2> Kiểm tra mục tiêu

if (key = = k_2) then return val;

<3.3> Dò tìm trên từng quan hệ ngữ nghĩa các keyphrase có quan hệ với key

for i from 1 to 25 do

if (val*minValR[i]) > Threshold) then

for each k such that k r_i (r_i^{-1}) key do

KQueue.add(k, val*val_r_i (key, k)); // bổ sung vào hàng đợi keyphrase mới cùng với độ ưu tiên tương ứng

return 0; // trả về 0 khi không tìm được bất kỳ một liên kết ngữ nghĩa nào giữa k_1 và k_2.

4.1.3. Thuật toán tính độ tương quan ngữ nghĩa giữa hai đồ thị keyphrase

Bài toán được đặt ra như sau: Cho trước một ontology CK_ONTO và hai đồ thị keyphrase H, G. Yêu cầu tính giá trị Rel (H, G) phản ánh độ tương quan về ngữ nghĩa giữa hai đồ thị.

Input: Ontology CK_ONTO

Hai đồ thị keyphrase H, G (ở dạng mở rộng)

Output: một giá trị Rel(H,G) ∈ [0,1]

Các bước thực hiện:

Bước 1: Khởi tạo

Đặt trạng thái ban đầu cho một số biến điều khiển

Sub_KG:={} // lưu các đồ thị con của H

Projection:={} // lưu các phép chiếu bộ phận từ H đến G

Value:={} // lưu giá trị tương ứng của từng phép chiếu trong Projection

Bước 2: Tìm các đồ thị con của H

Nhận xét: Một subKG của đồ thị G có thể nhận được từ G bằng cách xóa đi một hay nhiều đỉnh quan hệ (và các cung kề tương ứng) hoặc các đỉnh keyphrase cô lập.

Sub_KG ← Find_SubKG(H);

Bước 3: Thực hiện vòng lặp for để dò tìm các phép chiếu từ các đồ thị con của H tới G

for kg in Sub_KG do

//Tìm các phép chiếu từ kg đến G và bổ sung vào Projection

Projection ← Projection ∪ Find_Projection(kg, G)

Bước 4: Tính giá trị của mỗi phép chiếu $\nu(\Pi)$ trong Projection và lưu vào biến Value

Bước 5: Tìm $Rel(H,G) = Max(Value)$

❖ **Thuật giải tìm đồ thị con của một đồ thị keyphrase mở rộng**

```
Find_SubKG(H) {
        Tìm các đỉnh keyphrase cô lập trong đồ thị H
        for (keyphrase_node i  in  Tập đỉnh cô lập) {
            Bỏ đỉnh i, ta được H' là một đồ thị con của H
            Sub_KG ←H'
            Find_SubKG(H'); // gọi đệ qui hàm tìm đồ thị con
        }

        for ( relation_node j in H) {
            Bỏ đỉnh quan hệ j và các cạnh kề tương ứng, ta được H" là một đồ thị con của H
            Sub_KG ←H"
            Find_SubKG(H"); // gọi đệ qui hàm tìm đồ thị con
        }
}
```

❖ **Thuật giải tìm phép chiếu từ đồ thị keyphrase H tới đồ thị keyphrase G**

```
Find_Projection(H, G) {
   maxvalue = 0;
    for ( relation_node h_node in H)
        for (relation_node g_node in G{
            if (β(hnode, gnode) > Threshold_relation ){
               Gọi h_a0 và h_a1 là hai đỉnh keyphrase kề với h_node
               Gọi g_a0 và g_a1 là hai đỉnh keyphrase kề với g_node

               if ( α(h_a0, g_a0) > Threshold and  α(h_a1, g_a1) > Threshold) {
                   projection = ∅;
                   queue = new empty_queue();
                   projection [hnode] = gnode;
                   projection [h_a0] = g_a0;
                   projection [h_a1] = g_a1;

                   queue.push(h_a0);
                   queue.push(h_a1);
                   h_close = {h_a0, h_a1};
                   g_close = {g_a0, g_a1};

                   while (count(queue) > 0 ) {
                       h_cur = queue_pop (queue) ;
                       h_exp = tập các đỉnh keyphrase kề với h_cur trong H
                       g_exp = tập các đỉnh keyphrase kề với projection [h_cur] trong G
                       h_exp = h_exp \ h_close;
                       g_exp = g_exp \ g_close;

                       // tìm một tương ứng từ tập các đỉnh h_exp vào tập các định g_exp, mỗi
               đỉnh keyphrase có một đỉnh quan hệ đính kèm
                       project_cur = stable_marriage_problem_match (h_exp, g_exp);
                       if (project_cur !=∅) {
                          projection = projection ∪ project_cur;
                          h_close = h_close ∪ h_exp;
                          g_close = g_close ∪ g_exp;
                          queue.append (h_exp);
                       }
                       else{
                          projection = ∅;
                          break while;
                       }
                   }
                   h_left = tập các đỉnh keyphrase cô lập trong H;
                   g_left = tập các đỉnh keyphrase cô lập trong G;
                   if (projection !=∅){
                       project_isolate = stable_marriage_problem_match (h_left, g_left);
                       projection = projection ∪ project_isolate;
                       maxvalue = max(evaluate_projection(project_isolate), maxvalue);
                   }
               }
           }
       }
       if (count (h->relation_list()) ==0)
              // nếu H chỉ chứa các đỉnh keyphrase cô lập
          if (stable_match_group (h->keyphrase_list(), g->keyphaselist(), projection)){
               project_isolate = stable_marriage_problem_match (h_left, g_left);
               projection = projection ∪ project_isolate;
               maxvalue  =  max(evaluate_projection(project_isolate),  maxvalue); // tính giá trị
       của phép chiếu theo công thức
          }
       return maxvalue;
}
```

Vấn đề cần tiếp tục nghiên cứu: Nâng cao hiệu quả thuật toán so khớp đồ thị bằng cách xem xét bài toán Tìm đồ thị con đẳng cấu - một bài toán quyết định (decision problem) thuộc loại NP-đầy đủ (NP-complete).

4.2. XÂY DỰNG ĐỒ THỊ KEYPHRASE BIỂU DIỄN TÀI LIỆU

4.2.1. Rút trích tự động các keyphrase đặc trưng ngữ nghĩa của tài liệu

Rút trích tự động các keyphrase đặc trưng ngữ nghĩa (KĐTNN) của tài liệu là quá trình tự động chọn lọc các từ hay cụm từ có khả năng mô tả ngắn gọn và chính xác các chủ đề được thảo luận trong tài liệu, mang thông tin về nội dung nồng cốt của một tài liệu. Rút trích KĐTNN là nhiệm vụ khó khăn và cốt lõi nhất của một hệ thống tìm kiếm hướng ngữ nghĩa. Kết quả rút trích KĐTNN đạt được rất hữu dụng trong nhiều ứng dụng như tóm lược văn bản (Text Summarization), phân loại văn bản (Text Classification), truy hồi thông tin ngữ nghĩa (Semantic Information Retrieval) …Mặc dù các KĐTNN được dùng rộng rãi trong các hệ thống ứng dụng khác nhau, nhưng việc rút trích các KĐTNN tương ứng cho từng tài liệu bằng phương pháp thủ công tốn rất nhiều thời gian và công sức. Nhu cầu này là động lực thúc đẩy các nghiên cứu rút trích tự động các KĐTNN. Hiện nay đã có nhiều nghiên cứu xây dựng các công cụ hỗ trợ rút trích keyphrase tự động từ các tài liệu theo nhiều hướng tiếp cận khác nhau như: Bibclassify, Extractor, TerMine, Topia term extractor, Orchestr8 Keyword Extraction, Wikifier, Wikipedia Miner, SEO keyword extraction, Scorpion, Tagthe.net, Yahoo term extraction, Keyphrase Extraction Tool của Niraj Kumar, Carrot2, KEA/KEA++, Maui, Stanford topic Modeling tool, Mallet, … Có thể phân các nghiên cứu về rút trích tự động các KĐNN thành 3 hướng chính:

- Hướng tiếp cận sử dụng từ điển: sử dụng một từ điển để rút trích các keyphrase đặc trưng trong câu hay văn bản bằng cách so trùng các từ mục trong từ điển với các cụm từ trong tài liệu. Thuận lợi của hướng tiếp cận này là nhanh và đơn giản, tuy nhiên hiệu suất lại phụ thuộc vào độ lớn của từ điển và không hiệu quả khi giải quyết bài toán nhận dạng danh từ riêng hay các thuật ngữ mới trong những phạm vi chuyên biệt.
- Hướng tiếp cận ngôn ngữ học: dùng cơ sở tri thức ngữ nghĩa từ vựng như WordNet, Wikipedia, …các phương pháp đánh giá theo kinh nghiệm, các phương pháp

luật để rút trích các keyphrase. Hướng tiếp cận này có thể đạt được độ chính xác cao, tuy nhiên còn phụ thuộc vào việc thiết kế từng hệ thống cụ thể. Khó khăn chính là việc xây dựng một cơ sở tri thức cho những miền chuyên biệt có phạm vi lớn, việc này đòi hỏi rất nhiều thời gian và công sức.

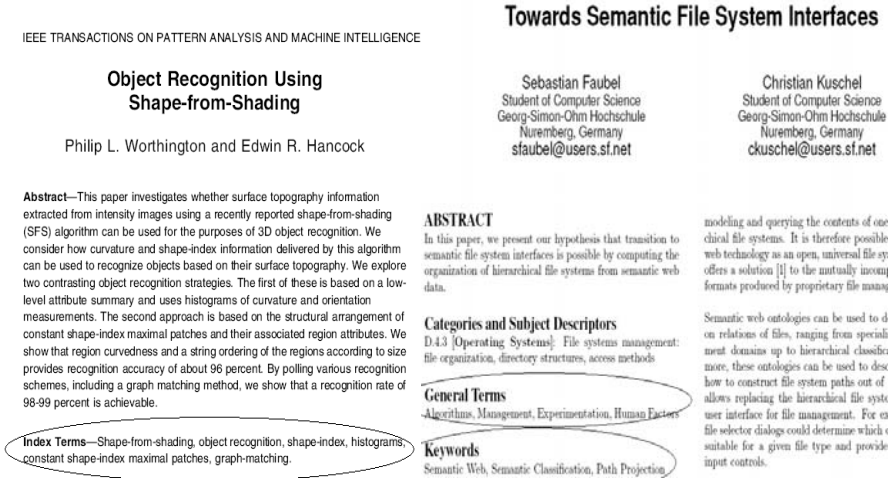
- Hướng tiếp cận bằng phương pháp thống kê: là quá trình học các giá trị đã được thống kê từ một kho ngữ liệu lớn để rút trích các cụm từ và nó liên quan mật thiết với hướng tiếp cận n – gram (n có giá trị 2, 3, 4). Mặc dù có gia tăng về mặt tính toán, kỹ thuật này không đòi hỏi nhiều công sức để tạo ra từ điển hay cơ sở tri thức mà còn có khả năng lấy được các thuật ngữ có trọng số cao trong kho ngữ liệu. Tuy nhiên, hạn chế của phương pháp này là có thể không rút trích được các keyphrase đặc trưng có tần số thấp.

Dựa trên kết quả đã thử nghiệm thì các công cụ hỗ trợ rút trích keyphrase tự động kể trên lại không mang lại hiệu quả trích xuất cao, kết quả thu được không chính xác. Hệ thống không rút trích được những keyphrase trọng yếu, đặc trưng cho tài liệu, những keyphrase cần thiết thì bị bỏ qua, số keyphrase dư thừa, không phù hợp thì nhiều.

Trong phần này, chúng tôi sẽ trình bày cơ chế rút trích keyphrase đặc trưng bằng cách sử dụng phương pháp so trùng dựa trên việc khai thác nguồn tri thức ontology CK_ONTO. Ý tưởng cơ bản là thực hiện một quá trình dò tìm và so trùng giữa những keyphrase đã được định nghĩa trong Ontology với các cụm từ có trong tài liệu, sử dụng kỹ thuật so chuỗi gần đúng kết hợp với một số qui tắc heuristic nhằm tăng cường tốc độ truy xuất và đạt được kết quả tốt hơn. Việc rút trích keyphrase được thực hiện bằng cách không duyệt hết nội dung tài liệu mà chỉ rút trích trong một số trường đặc biệt vì với những tài liệu có kích thước lớn, việc lọc ra tất cả các keyphrase trong tài liệu là công việc khổng lồ, tốn nhiều thời gian và tài nguyên, trong khi các keyphrase phản ánh nội dung chính hay chủ đề của tài liệu thường được nêu lên tại một số trường (đoạn, mục) được xem là chứa đầy đủ các keyphrase trọng yếu cần thiết. Ví dụ, so với toàn bộ

tài liệu, tiêu đề sẽ biểu diễn ngắn gọn thông tin trong tài liệu, thường chỉ ra nội dung của tài liệu đó và do đó giúp người đọc nhanh chóng nắm bắt được đại ý của toàn văn bản, vì thế các keyphrase đặc trưng có khả năng xuất hiện trong tiêu đề là rất lớn. Dựa trên đặc thù của mỗi loại tài liệu, ta có những cách thức khác nhau trong việc chọn lọc các phần nội dung dùng cho việc rút trích keyphrase đặc trưng như sau:

- Nếu tài liệu là một bài báo khoa học: các keyphrase đặc trưng thường xuất hiện tại các trường Title, Keywords , Index Terms, Abstract.



IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE

Object Recognition Using Shape-from-Shading

Philip L. Worthington and Edwin R. Hancock

Abstract—This paper investigates whether surface topography information extracted from intensity images using a recently reported shape-from-shading (SFS) algorithm can be used for the purposes of 3D object recognition. We consider how curvature and shape-index information delivered by this algorithm can be used to recognize objects based on their surface topography. We explore two contrasting object recognition strategies. The first of these is based on a low-level attribute summary and uses histograms of curvature and orientation measurements. The second approach is based on the structural arrangement of constant shape-index maximal patches and their associated region attributes. We show that region curvedness and a string ordering of the regions according to size provides recognition accuracy of about 96 percent. By polling various recognition schemes, including a graph matching method, we show that a recognition rate of 98-99 percent is achievable.

Index Terms—Shape-from-shading, object recognition, shape-index, histograms, constant shape-index maximal patches, graph-matching.

Towards Semantic File System Interfaces

Sebastian Faubel
Student of Computer Science
Georg-Simon-Ohm Hochschule
Nuremberg, Germany
sfaubel@users.sf.net

Christian Kuschel
Student of Computer Science
Georg-Simon-Ohm Hochschule
Nuremberg, Germany
ckuschel@users.sf.net

ABSTRACT
In this paper, we present our hypothesis that transition to semantic file system interfaces is possible by computing the organization of hierarchical file systems from semantic web data.

Categories and Subject Descriptors
D.4.3 [Operating Systems]: File systems management: file organization, directory structures, access methods

General Terms
Algorithms, Management, Experimentation, Human Factors

Keywords
Semantic Web, Semantic Classification, Path Projection

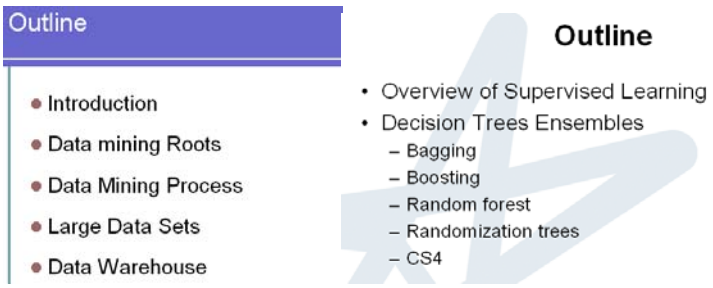
modeling and querying the contents of one
chical file systems. It is therefore possible
web technology as an open, universal file sy
offers a solution [1] to the mutually incomp
formats produced by proprietary file manag

Semantic web ontologies can be used to d
on relations of files, ranging from speciali
ment domains up to hierarchical classific
more, these ontologies can be used to desc
how to construct file system paths out of
allows replacing the hierarchical file syste
user interface for file management. For ex
file selector dialogs could determine which
suitable for a given file type and provide
input controls.

- Nếu tài liệu là Ebook, luận văn, luận án: có thể rút trích tại Title, Table of content/ Contents, Index của tài liệu.

Contents

- Nếu tài liệu là slide: có thể rút trích tự động từ các mục Title, Content, Outline



4.2.2. Qui trình biểu diễn văn bản thành đồ thị keyphrase

Input:

- Ontology CK_ONTO
- Một tập tin tài liệu
- Các thông tin mô tả tài liệu đi kèm (từ cơ sở dữ liệu DB)

Output: Một đồ thị keyphrase biểu diễn ngữ nghĩa cho tài liệu

Quá trình biểu diễn văn bản thành đồ thị keyphrase có thể được tiến hành tuần tự theo các bước chính sau:

Bước 1: Chuyển đổi định dạng tài liệu thành dạng thức xử lý chung là TEXT (.txt)

Bước 2: Xác định loại hình tài liệu (như paper, ebooks, slide, …)

Bước 3: Rút trích các keyphrase đặc trưng phản ánh nội dung chính trong tài liệu

<3.1> Trích xuất một số trường đặc biệt bên trong nội dung tài liệu được xem là chứa đầy đủ các keyphrase trọng yếu cần thiết và lưu vào biến Fields, tùy biến theo từng loại hình tài liệu.

- Nếu tài liệu là paper thì trích các keyphrase đã được khai báo sẵn trong mục Keywords hay Index Terms rồi lưu trực tiếp vào Doc_Keys và rút trích các mục Title, Abstract, Conclusion, Reference của bài báo lưu vào biến Fields.
- Nếu tài liệu là ebook, luận văn hay slide thì trích xuất các mục Title, Table of Content hay Content/Outline, Preface lưu vào Fields.

Nếu tài liệu input có sẵn các thông tin mô tả theo kèm (metadata lưu trong

CSDL) thì các trường đặc biệt kể trên sẽ được trích lọc trực tiếp từ các thông tin này. Trong trường hợp chưa được cung cấp sẵn thì một module rút trích tự động hoặc bán tự động được yêu cầu. Ví dụ, đối với những tài liệu có cấu trúc tương đối xác định như các bài báo gửi tham gia hội thảo quốc tế, đăng tạp chí quốc tế thường được soạn thảo và trình bày theo một định dạng chuẩn đã qui định sẵn, theo đó nội dung của từng phần như Abstract, Index Term, Keywords, Conclusion, Reference được khai báo bởi các từ khóa tương ứng, do đó có thể dễ dàng phân tách từng phần riêng biệt và thực hiện rút trích tự động keyphrase. Hoặc việc xác định title của một bài báo có thể dựa vào nhận xét: title là câu duy nhất của đoạn đầu tiên, nghĩa là ta xét đoạn đầu tiên của văn bản, nếu đây chỉ có một câu thì câu này có thể là title. Cách xác định này phụ thuộc định dạng của văn bản đầu vào.

<3.2> Thực hiện một quá trình dò tìm và so khớp (gần đúng) từng keyphrase có trong ontology với từng chuỗi cấu trúc bên trong nội dung rút trích từ tài liệu

for key in Keyphrases do

if Test_Keyphrase(key, Fields) then

Doc_Keys ← Doc_Keys ∪ {key};

// hàm Test_Keyphrase(key, Fields) sẽ kiểm tra xem key có xuất hiện trong Fields hay không dùng kỹ thuật so chuỗi gần đúng.

Bước 4: Xác định các mối quan hệ ngữ nghĩa có thể có trên tập keyphrase từ Ontology đã xây dựng được

Sau khi đã có tập keyphrase rút trích từ tài liệu, dùng ontology CK_ONTO để suy diễn ra các mối quan hệ ngữ nghĩa có thể có trên tập keyphrase này.

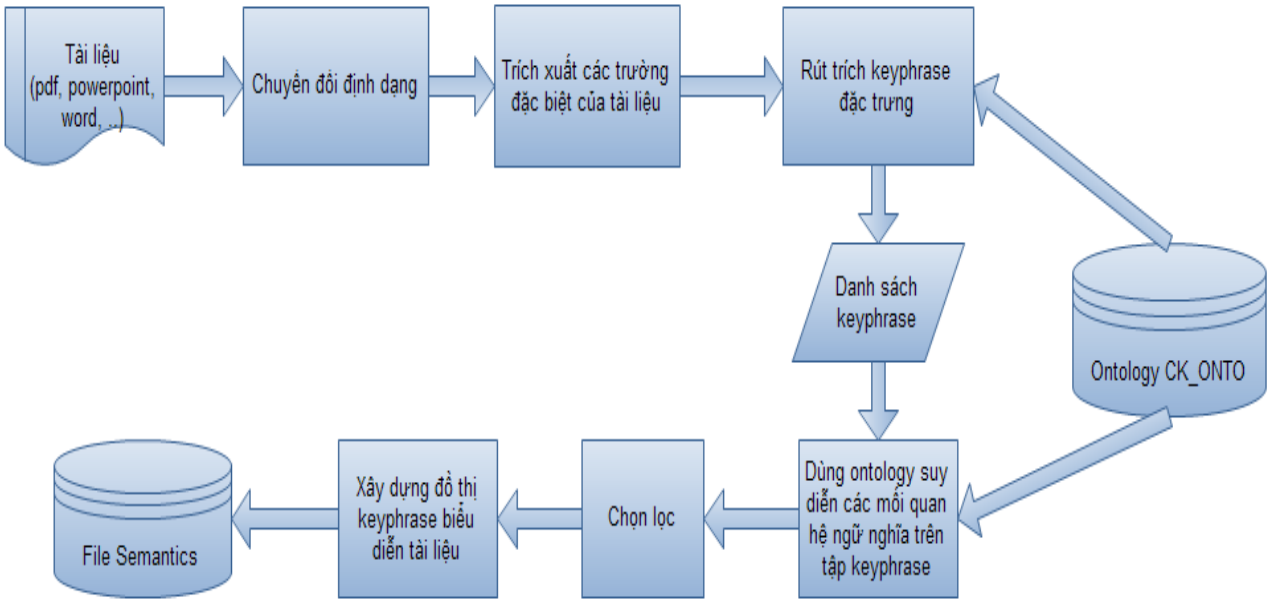
Bước 5: Chọn lọc và loại bỏ các liên kết ngữ nghĩa không cần thiết

Trong quá trình trích xuất tự động các keyphrase từ tài liệu hay dò tìm các quan hệ ngữ nghĩa trong ontology có thể có những keyphrase hay quan hệ là dư thừa, không cần thiết. Do đó, cần có sự can thiệp và giám sát của chuyên gia nhằm chọn lọc lại các keyphrase, các liên kết ngữ nghĩa phù hợp, mang ý nghĩa về mặt thể hiện nội dung

chính mà tài liệu đề cập đến.

Bước 6: Xây dựng đồ thị keyphrase

Từ tập keyphrase đặc trưng rút trích từ tài liệu và các quan hệ ngữ nghĩa, ta xây dựng đồ thị keyphrase có tập đỉnh và tập cung tương ứng biểu diễn cho tài liệu. Đồ thị keyphrase này sẽ được lưu lại trong định dạng tập tin văn bản có cấu trúc dựa trên một số từ khóa và qui ước về cú pháp như đã trình bày trong phần 3.4.



Hình 4.2. Qui trình chung biểu diễn văn bản thành đồ thị keyphrase

❖ **Lưu ý:**

Ta có nhận xét sau: Mục Index trong ebook có thể được xem xét để rút trích các keyphrase đặc trưng tuy nhiên độ ưu tiên thấp hơn so các mục khác. Theo đó, tập keyphrase rút trích được từ Index phải được chọn lọc lại, những keyphrase nào không liên quan hay ít có liên quan với những keyphrase rút trích từ các mục có độ ưu tiên cao như Title, Content thì sẽ bị loại bỏ (bằng cách tính toán độ đo tương đồng ngữ nghĩa giữa các keyphrase). Tương tự, đối với paper thì mục Abstract, Conclusion, Reference được xem là có độ ưu tiên thấp hơn so với Title, Keywords/Index Terms, một số keyphrase ít có ý nghĩa trong việc phản ánh nội dung chính của tài liệu khi rút trích từ

các mục này cần được loại bỏ bớt. Đây là một vấn đề cần tiếp tục nghiên cứu để hoàn thiện hơn cho thuật toán. Ngoài ra, để đánh giá mức độ quan trọng của keyphrase trong việc phản ánh nội dung tài liệu (chẳng hạn như những keyphrase xuất hiện trong Title có độ ưu tiên cao nhất) ta có thể gán thêm trọng số cho mỗi đỉnh keyphrase trong đồ thị biểu diễn tương ứng. Khi đó, mỗi tài liệu sẽ được biểu diễn bởi một ***đồ thị keyphrase có trọng số*** giàu ngữ nghĩa hơn.

Ví dụ: xét tài liệu #20 có nội dung như sau:

Extracting Conceptual Graphs from Japanese Documents for Software Requirements Modeling

Ryo Hasegawa[1] Motohiro Kitamura[1] Haruhiko Kaiya[2] Motoshi Saeki[1]

[1]Dept. of Computer Science, Tokyo Institute of Technology
Ookayama 2-12-1, Meguro-ku, Tokyo 152, Japan
Email: saeki@se.cs.titech.ac.jp
[2]Dept. of Computer Science, Shinshu University
Wakasato 4-17-1, Nagano 380-8553, Japan
Email: kaiya@cs.shinshu-u.ac.jp

Abstract

A requirements analysis step plays a significant role on the development of information systems, and in this step we produce various kinds of abstract models of the systems (called requirements models) according to the adopted development processes, e.g. class diagrams in the case of adopting object-oriented development. However, constructing these models of sufficient quality requires highest intellectual tasks and skills of human requirements analysts. In this paper, we develop a computerized tool to extract from a set of Japanese text documents conceptual information, called *conceptual graph*, which can be used as intermediate representation to generate software requirements models. More concretely, by applying the variation of text-mining techniques that we have developed, we extract significant words from text documents referring to the same problem domain and identify relevant relationships among them. The extracted words can be considered as concepts and they are constituents of a conceptual graph in the domain. This constructed graph can be used for generating requirements models, e.g. object oriented models, feature model, and even as a domain ontology that can be utilized during requirements analysis activities. We have made experimental analyses of our tool. This paper also includes the discussion on how the extracted conceptual graph can act as an object-oriented model, a feature model and a domain ontology, in order to show its wide applicability.

Keywords: Conceptual Graph, Requirements Modeling, Text mining, NL processing

1 Introduction

Since a requirements analysis step is the first one in information systems development processes, the quality of

Feature-Oriented Analysis technique, we should produce a feature oriented model. Thus we can produce various kinds of model in a requirements analysis step according to the adopted development process. We call these models, i.e. abstract models of the system that produced in a requirements analysis step, *requirements models*. We should construct a requirements model of high quality as early as possible to reduce development costs and efforts. However, human engineers are required to perform highly intellectual and complicated activities and to have distinguished skills in order to construct a requirements model of high quality. In addition, they should be experts to various modeling techniques that can be adopted. A current status is that a limited number of domain experts are involved in requirements modeling in their domains, spending their large efforts. We need some supporting techniques to assist human engineers in constructing various types of requirements models of higher quality with less effort.

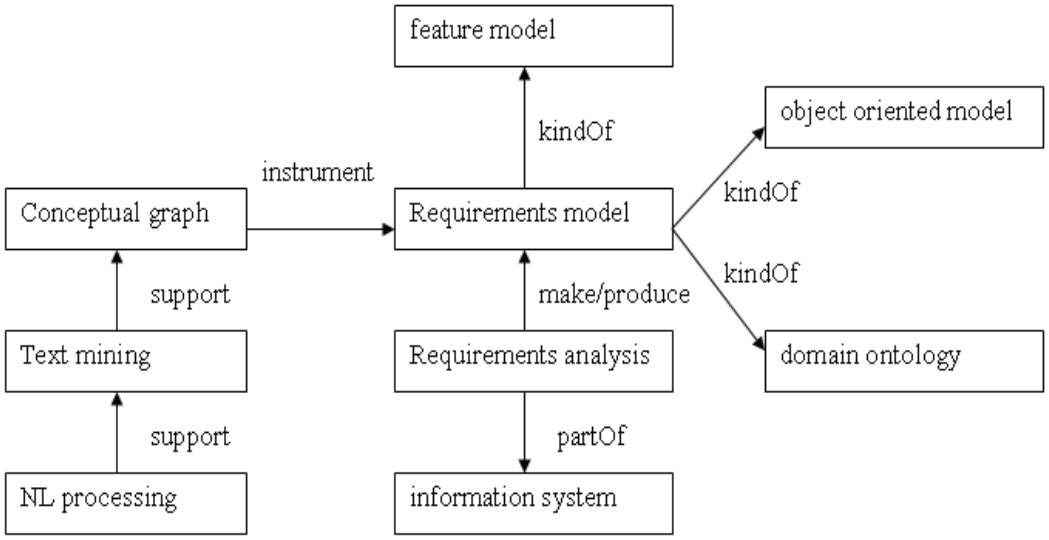
On the other hand, it is a rare case that we construct a requirements model whose domain is quite new and does not appear before. If we had reusable assets helpful for requirements modeling, we could get the model efficiently. However, we have not accumulated sufficient reusable assets of requirements models in a certain domain yet. Rather, we can get many text documents referring to the domain, including the electronic texts lying over Internet. In fact, the experts to modeling frequently use the documents regarding to the topics relevant to the problem domain so as to get important information. Thus, it can be considered as a promising support technique to extract from the documents information necessary for requirements modeling. These documents are written in natural language, and the constituents that a requirements model should have, e.g. concepts and their relationships appear in the documents as words and their co-occurrences in a suitable abstraction level, because of the abstractness of natural language descriptions. The words that commonly

Các đoạn văn bản được xem là có chứa các keyphrase trọng yếu của tài liệu được dùng để giới hạn phạm vi tìm kiếm và rút trích bao gồm: ***Title*** = "Extracting Conceptual Graphs from Japanese Documents for Software Requirements Modeling, ***Abstract*** = "A requirements analysis …its wide applicability", ***Keywords*** = " Conceptual Graph, Requirements Modeling, Text mining, NL processing."

Kết quả ở giai đoạn dò tìm những keyphrase trong CK_ONTO có xuất hiện trong các đoạn văn bản trên, ta thu được tập keyphrase đặc trưng:

Doc_Keys = { requirements analysis, conceptual graph, requirements modeling, requirements model, information system, text mining, NL processing, object oriented model, feature model, domain ontology}

Sau khi xác định và chọn lọc các quan hệ ngữ nghĩa có trên tập keyphrase, ta xây dựng được đồ thị biểu diễn cho tài liệu trên như sau:



và được lưu lại theo định dạng tập tin văn bản:

<graph_keyphrase>

id_doc: D0020;

G_K : { requirements analysis, conceptual graph, requirements model, information system, text mining, NL processing, object oriented model, feature model, domain ontology };

E : {(NL processing, text mining, support), (text mining, conceptual graph, support), (conceptual graph, requirements model, instrument), (requirements model, object oriented model, kindOf), (requirements model, feature model, kindOf), (requirements model, domain ontology, kindOf), (requirements analysis , requirements model, produce), (requirements analysis , information system, partOf)};

</graph_keyphrase>

Với mục đích cải thiện tối đa hiệu quả của hệ thống rút trích keyphrase tự động theo hướng tiếp cận ngôn ngữ học hay ontology vốn còn nhiều giới hạn, vấn đề cần được nghiên cứu và giải quyết tiếp theo là nghiên cứu các mô hình và giải pháp rút trích tự động các keyphrase từ tài liệu trên cơ sở lai ghép phối hợp các mô hình đã có, các kỹ thuật trong xác suất thống kê, máy học, kỹ thuật xử lý ngôn ngữ tự nhiên, …Theo đó xây dựng bộ công cụ hỗ trợ lập chỉ mục tự động cho tài liệu dựa trên các mô hình biểu diễn như đã nêu trên.

4.3. XỬ LÝ CÂU TRUY VẤN

Khi người dùng có nhu cầu tìm kiếm thông tin, sẽ nhập vào câu truy vấn thông qua giao diện người dùng bằng ngôn ngữ tự nhiên hay một dạng thức qui ước nào đó. Trong trường hợp tìm kiếm theo từ khóa, hệ thống tiến hành so khớp từ khoá và trả về kết quả là tập tài liệu có chứa chính xác từ khoá đã được nhập vào. Đối với chức năng tìm kiếm theo ngữ nghĩa, tương tự như các tài liệu, câu truy vấn cũng sẽ trải qua các giai đoạn rút trích keyphrase và biểu diễn thành một đồ thị keyphrase tương ứng. Không giả định câu truy vấn là một câu bằng ngôn ngữ tự nhiên, ta giới hạn lại cấu trúc câu truy vấn chỉ là một hay một số cụm từ diễn đạt nội dung chính muốn tìm kiếm.

4.3.1. Ngôn ngữ đặc tả câu truy vấn

Câu truy vấn có thể được cho dưới dạng một text có cấu trúc dựa trên một số từ khoá cùng với những qui ước khai báo trong tìm kiếm. Câu truy vấn có thể được khai báo theo cấu trúc gồm hai phần:

- Phần nội dung tìm kiếm chính được đặc tả dưới dạng một danh sách các từ hay cụm từ được phân cách với nhau bằng khoảng trắng (có thể kết hợp với các toán tử Boolean hay các ký tự đặc biệt giúp cho việc tìm thông tin chính xác và đúng yêu cầu hơn). Hệ thống cho phép tìm kiếm theo nhiều cụm từ bằng cách đặt những cụm từ cần tìm vào trong hai dấu nháy kép, ví dụ: “search engine” “information retrieval”. Phần khai báo này sẽ được dùng trong việc xây dựng đồ thị keyphrase biểu diễn câu truy vấn. Nếu cụm từ khóa truy vấn được đặt trong dấu nháy kép “”

thì chương trình sẽ tìm những tài liệu có liên quan đến "đúng" cụm từ này.

- Phần thứ hai là tìm kiếm nâng cao theo các khóa dữ liệu mô tả thuộc tính của tài liệu. Để hỗ trợ cho việc tìm kiếm nhanh chóng và chính xác hơn, hệ thống đưa ra một số những từ khóa nhằm mục đích giới hạn việc tìm kiếm vào trong những điều kiện xác định. Từ khóa luôn kèm theo dấu hai chấm ':' và những từ sau đó có thể viết dính liền hoặc cách ra bởi khoảng trắng. Phần tìm kiếm nâng cao được định nghĩa thông qua các cặp thẻ có cú pháp như sau:

  <tên khóa dữ liệu (metadata_element)> : <danh sách các từ khóa tìm kiếm>

  Cú pháp của câu truy vấn có dạng:

  query ::= (word " ")+ (metadata_search_block)*

  metadata_search_block ::= metadata_element: (word " ")+

  metadata_element ::= filetype | format | language | date | publisher | author | country | related terms

  word ::= (a..z | A..Z | 0..9 | + | # | / | .)+

Ví dụ: programming OOP in C++ authorname:ritchie type:book format: pdf.

Ontology "document representation" "conceptual graph" type:paper.

4.3.2. Quy trình xử lý câu truy vấn

Quá trình này gồm các giai đoạn chính như sau:

Input: câu truy vấn người dùng

Output: đồ thị keyphrase biểu diễn câu truy vấn và các thông tin lọc.

Các bước thực hiện chính:

Bước 1: Phân tách phần nội dung tìm kiếm chính và ghi nhận các thông tin mô tả liên quan (giúp khoanh vùng, giới hạn phạm vi tìm kiếm hay lọc kết quả)

Bước 2: Rút trích các keyphrase mô tả nội dung chính muốn tìm kiếm

<2.1> Phân tích và rút trích tự động các từ, cụm từ trong câu truy vấn.

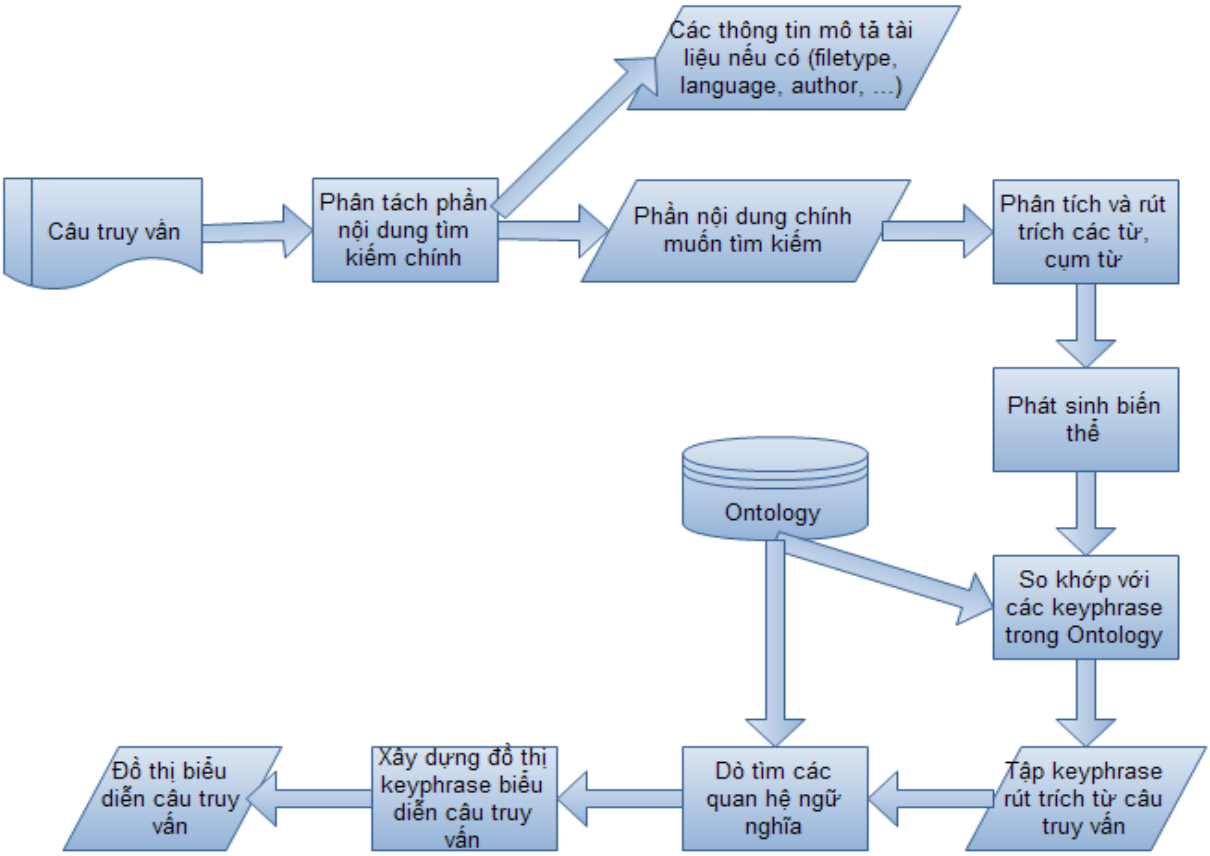
<2.2> Phát sinh các biến thể từ các thành phần trên

<2.3> So khớp (gần đúng) từ, cụm từ với keyphrase trong Ontology và chọn

lọc keyphrase ứng viên.

Bước 3: Dò tìm các quan hệ ngữ nghĩa trên tập keyphrase dựa trên ontology

Bước 4: Xây dựng đồ thị keyphrase biểu diễn câu truy vấn tương tự như đối với tài liệu.



Hình 4.3 : Qui trình xử lý câu truy vấn

4.4. BÀI TOÁN TÌM KIẾM THEO NGỮ NGHĨA

Như đã giới thiệu trong chương 2, một hệ thống truy tìm tài liệu (Document Retrieval System - DRS) là một hệ thống sẽ truy tìm những tài liệu trong số các tài liệu có trong cơ sở dữ liệu lưu trữ có nội dung liên quan, phù hợp, đáp ứng với nhu cầu thông tin của người dùng. Sau đó người dùng sẽ tìm kiếm thông tin họ cần trong các tài liệu liên quan đó. Hệ thống DRR có hai khối chức năng chính, đó là lập chỉ mục và tra

cứu hay tìm kiếm. Lập chỉ mục là giai đoạn phân tích tài liệu để rút trích các đơn vị thông tin từ tài liệu và biểu diễn lại tài liệu bởi các đơn vị thông tin đó. Theo hướng tiếp cận của đề tài, đơn vị thông tin được xét đến là các keyphrase đặc trưng của tài liệu, mang ý nghĩa thể hiện nội dung chính của tài liệu. Tra cứu là giai đoạn tìm kiếm trong cơ sở dữ liệu những tài liệu phù hợp với nội dung câu truy vấn. Trong giai đoạn tra cứu, nhu cầu thông tin của người sử dụng được đưa vào hệ thống dưới dạng một câu truy vấn theo dạng thức qui ước như đã nêu trong 4.3. Câu truy vấn và tập tài liệu sẽ được phân tích và biểu diễn thành các đồ thị keyphrase. Hệ thống sẽ sử dụng một hàm so khớp để so khớp đồ thị keyphrase biểu diễn câu hỏi với các đồ thị keyphrase biểu diễn tài liệu để đánh giá độ tương quan về ngữ nghĩa của các tài liệu với câu truy vấn, trả về danh sách tài liệu có liên quan được sắp hạng cùng với đề xuất tinh chỉnh câu truy vấn.

4.4.1. Mô hình tổng quát của hệ truy tìm tài liệu theo ngữ nghĩa

Mô hình tổng quát của hệ truy tìm tài liệu theo ngữ nghĩa là một hệ thống gồm có bốn thành phần, được ký hiệu bởi bộ bốn:

(Q, KG(Q), SDB, rank)

trong đó các thành phần được mô tả như sau :

- Q là tập các câu truy vấn.
- KG(Q) là mô hình biểu diễn ngữ nghĩa cho câu truy vấn.
- SDB = (D, FS, DB, ONTO, SDB_R) là mô hình cơ sở tài liệu có ngữ nghĩa
- $rank : Q \times D \rightarrow \square^{+}$ là hàm xếp hạng theo độ đo tương quan ngữ nghĩa giữa các câu truy vấn trong Q và các tài liệu có trong D. Giá trị xếp hạng rank(q_i, d_j) với $q_i \in Q$ và $d_j \in D$ xác định một thứ tự về mức độ liên quan của tài liệu d_j với câu truy vấn q_i trong tập tài liệu D.

4.4.2. Thuật toán tìm kiếm theo ngữ nghĩa tổng quát

Input:

- Kho tài liệu được tổ chức theo mô hình SDB.

- Câu truy vấn q của người dùng.

Output: danh sách các tài liệu (được sắp hạng) có liên quan đến thông tin truy vấn.

Các bước thực hiện chính:

Bước 1: Ghi nhận thông tin truy vấn của người dùng.

Bước 2: Xử lý và biểu diễn câu truy vấn q thành đồ thị keyphrase KG(q).

Bước 3: Thực hiện một quá trình dò tìm các tài liệu có trong kho phù hợp với thông tin truy vấn của người dùng và trả về tập tài liệu kết quả đã được sắp hạng.

Các tài liệu có trong D được biểu diễn bởi tập các đồ thị keyphrase KG(D) = $\{G_1, G_2, \ldots, G_k\}$, nghĩa là ta đánh index cho các tài liệu bằng một ngôn ngữ index dựa trên đồ thị keyphrase

<3.1> Tìm trong KG(D) những đồ thị "trùng khớp" với KG(q) bằng cách tính toán so khớp giữa các đồ thị

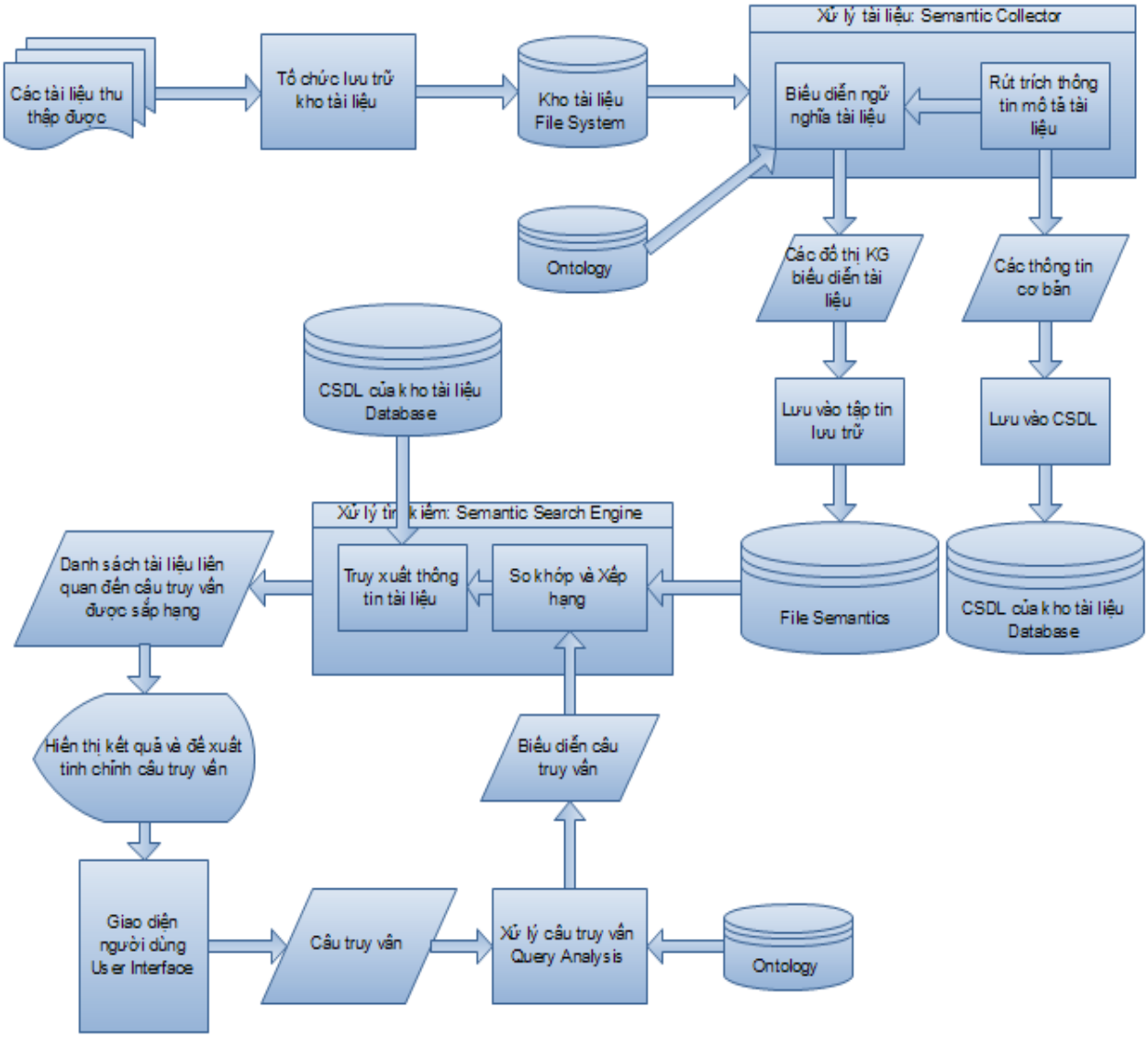
```
for g in KG(D)
        if  Match(g, KG(q)) then
                Result ← (g, Rel(g, KG(q))
```

<3.2> Xếp hạng các tài liệu trong tập kết quả Result theo giá trị Rel tương ứng

Bước 4: Hiển thị kết quả và đề xuất tinh chỉnh câu truy vấn

Kết quả thu được bao gồm một danh sách các tài liệu có liên quan đến thông tin tìm kiếm của người dùng đã được sắp hạng và một danh sách các chủ đề con, các keyphrase có liên quan với từ khóa tìm kiếm ban đầu, qua đó hỗ trợ người dùng có thể sửa đổi truy vấn và tìm lại một lần nữa.

Bước 5: Điều chỉnh câu truy vấn và lặp lại từ bước 2 cho đến khi thỏa yêu cầu của người dùng .



Hình 4.4: Sơ đồ hoạt động của hệ thống tìm kiếm tài liệu theo ngữ nghĩa

4.5. XÁC ĐỊNH THƯ MỤC LƯU TRỮ CHO TÀI LIỆU

Xác định thư mục lưu trữ cho một tài liệu là quá trình gán tài liệu vào một thư mục tương ứng với chủ đề đã xác định trước. Như vậy, việc xác định thư mục tài liệu được thực hiện dựa trên các thao tác phân tích nhằm xác định lĩnh vực hay chủ đề mà nội dung tài liệu đề cập đến và lưu tài liệu vào thư mục tương ứng với chủ đề đó. Vì các kho tài nguyên thường có khối lượng khá lớn nên ngay từ đầu tổ chức kho ta không thể phân loại một cách thủ công được. Trong trường hợp phải cập nhật vào kho một số

lượng lớn tài liệu mà các thông tin mô tả kèm theo không được cung cấp sẵn thì việc lưu trữ thủ công bằng cách duyệt qua nội dung chính của từng tài liệu đó là rất khó khăn. Do đó một chương trình tự động được yêu cầu.

Nhằm tận dụng ưu điểm của các mô hình biểu diễn cũng như kỹ thuật tính toán độ đo tương tự ngữ nghĩa kể trên, chúng tôi xây dựng một giải thuật xác định thư mục lưu trữ tài liệu tự động dựa trên ý tưởng: mỗi thư mục trong hệ thống thư mục lưu trữ có thể được biểu diễn bởi một keyphrase thể hiện thông tin ngữ nghĩa liên quan và do đó, việc tìm ra độ đo tương quan ngữ nghĩa giữa thư mục và tài liệu bằng cách so khớp keyphrase biểu diễn thư mục với đồ thị keyphrase biểu diễn tài liệu cho ta một phép phân loại tài liệu vào thư mục tương ứng.

Thuật toán xác định thư mục lưu trữ cho một tài liệu d trong hệ thống thư mục FS bao gồm các bước chính như sau:

Input: Hệ thống thư mục FS

Một tài liệu d

Output: Thông tin thư mục lưu trữ tài liệu d

Bước 1: Ghi nhận thông tin về cây thư mục FS

Directories: = [<danh sách keyphrase biểu diễn thư mục trong FS>];

Bước 2: Xác định đồ thị keyphrase biểu diễn ngữ nghĩa của tài liệu.

Bước 3: Thực hiện vòng lặp for để tính toán độ tương quan ngữ nghĩa giữa từng thư mục trong Directories với tài liệu d sử dụng kỹ thuật so khớp như đã giới thiệu trong phần 4.1

for dir in Directories do

Tính giá trị Rel(dir, KG(d))

Bước 4: Gán tài liệu vào thư mục nào có giá trị Rel(dir, KG(d)) lớn nhất.