Kỷ yếu Hội nghị Quốc gia lần thứ VIII về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR); Hà Nội, ngày 9-10/7/2015
DOI: 10.15625/vap.2015.000210

THUẬT TOÁN MỚI VỀ SO KHỚP ONTOLOGY

Huỳnh Nhứt Phát[1], Hoàng Hữu Hạnh[2], Phan Công Vinh[3]
[1]Đại học Huế
[2]Đại học Huế
[3]Trường Đại học Nguyễn Tất Thành TP. HCM
huynhnhutphat@yahoo.com, hhhanh@hueuni.edu.vn, pcvinh@ntt.edu.vn

TÓM TẮT – *So khớp ontology (ontology matching) là một phần quan trọng trong kỹ nghệ ontology của Web ngữ nghĩa với mục tiêu tìm kiếm các so khớp (alignment) giữa các thực thể của các ontology đã cho. Trong nghiên cứu này chúng tôi đề xuất thuật toán mới và một công cụ dựa trên thuật toán này để tìm sự tương đồng giữa các thực thể của các ontology đầu vào. Thuật toán đề xuất này sử dụng độ đo mới về sự tương đồng của từ vựng và cũng sử dụng thông tin về cấu trúc của các ontology để xác định thực thể tương ứng của chúng. Độ đo sự tương đồng về từ vựng tạo ra một tập từ cho mỗi thực thể dựa trên nhãn và thông tin mô tả của chúng. Cách tiếp cận về cấu trúc tạo thành một mạng lưới cho mỗi nút trong các ontology. Sự kết hợp của phương pháp tiếp cận về từ vựng và cấu trúc tạo thành ma trận đồng dạng giữa ontology nguồn và ontology đích. Thuật toán đề xuất này đã được thử nghiệm dựa trên các chuẩn đã được công nhận và cũng được so sánh với các thuật toán khác hiện nay. Kết quả thực nghiệm của chúng tôi cho thấy thuật toán đề xuất rất hiệu quả và nhanh hơn so với các thuật toán khác.*

Từ khoá – *lexical similarity, ontology matching, similarity measure, structure similarity.*

I. GIỚI THIỆU

Các ontology là các mô hình khái niệm chia sẻ về một miền ứng dụng, cho nhiều người tham gia. Các ontology cho Web ngữ nghĩa được biểu diễn bởi RDFS hoặc OWL, đóng một vai trò rất quan trọng trong Web ngữ nghĩa. So khớp ontology được đề xuất như là một giải pháp tốt cho việc chia sẻ và tái sử dụng kiến thức, bằng cách cung cấp một cơ chế hình thức để xác định ngữ nghĩa của dữ liệu. Nó đóng một vai trò quan trọng trong việc mở rộng và sử dụng các ứng dụng dựa trên Web ngữ nghĩa [1]. Những năm gần đây có rất nhiều ontology được tạo ra với các miền khác nhau. So khớp ontology cho phép dữ liệu và tri thức được biểu diễn trong các ngôn ngữ khác nhau và các định dạng được chia sẻ. Chẳng hạn, với hai ontology ở đầu vào và tạo ra các so khớp (alignment) ở đầu ra. Các so khớp này là một tập hợp liên quan giữa các thực thể tương ứng về ngữ nghĩa, như là các lớp được đặt tên, các thuộc tính đối tượng và các thuộc tính dữ liệu của các ontology đầu vào.

Trong bài báo này chúng tôi trình bày phương pháp tổ hợp để tìm sự tương ứng giữa các thực thể (ví dụ: các lớp được đặt tên, các thuộc tính đối tượng và các thuộc tính dữ liệu) của các ontology, dựa trên sự tương đồng về từ vựng và cấu trúc của chúng. Trước tiên, chúng tôi xác định sự tương đồng về từ vựng trong số các lớp được đặt tên (các nút trong ontology), các thuộc tính đối tượng (các cạnh trong ontology) và các thuộc tính dữ liệu (các giá trị), sử dụng độ đo khoảng cách mới, nó tạo ra một túi từ (bag of words) cho mỗi thực thể trong các ontology đã cho. Các túi sau đó, được so sánh với nhau để tính toán sự tương đồng về từ vựng giữa hai thực thể với kiểu giống nhau (ví dụ như hai lớp được đặt tên) từ hai ontology. Vì vậy, việc sử dụng độ đo khoảng cách này, chúng tôi đưa ra ba ma trận khác nhau về từ vựng, nó tương ứng với các đặc điểm giống nhau của các lớp được đặt tên, các thuộc tính đối tượng và các thuộc tính dữ liệu của ontology nguồn và ontology đích. Thứ hai, chúng tôi tìm kiếm điểm tương đồng giữa các nút (các lớp được đặt tên) của ontology nguồn và ontology đích dựa trên các cấu trúc ontology của chúng. Nói cách khác, chúng tôi tạo ra một mạng lưới gồm nhiều nút, với mỗi nút sử dụng các nút lân cận của nó trong ontology nguồn và ontology đích để so sánh về mặt cấu trúc giữa chúng. Cuối cùng, trong giai đoạn này, ba ma trận có được ở các giai đoạn trước đó tương đồng về từ vựng và tương đồng về cấu trúc sẽ được kết hợp và sử dụng thêm kỹ thuật để tạo ra ma trận tương đồng toàn diện.

Chúng ta biết rằng so khớp ontology tạo ra sự tương ứng giữa các thực thể của hai ontology. Trong bài báo này chúng tôi trình bày thêm công cụ OMReasoner, nó tạo ra một khung ứng dụng có thể mở rộng về sự kết hợp của nhiều công cụ so khớp riêng lẻ và từ điển WordNet cũng như logic mô tả được sử dụng trong việc phân tích so khớp ontology. Nó xử lý so khớp ontology ở cả hai cấp độ từ và ngữ nghĩa, và nó sử dụng phần ngữ nghĩa cũng như cấu trúc của OWL-DL. Chúng tôi trình bày kết quả đạt được của OMReasoner với OAEI 2014 theo ba phương pháp: Benchmark, Conference và MultiFarm.

Bài báo được tổ chức như sau: Phần I giới thiệu. Phần II khảo sát các nghiên cứu liên quan. Phần III thảo luận với thuật toán đề xuất chi tiết. Phần IV đưa ra một ví dụ minh họa. Phần V trình bày các kết quả thực nghiệm và công cụ được triển khai dựa trên thuật toán đã mô tả. Phần VI trình bày công cụ OMReasoner có thể mở rộng về sự kết hợp nhiều công cụ so khớp riêng lẻ, Phần VII các kết luận và nhận xét.

II. CÁC NGHIÊN CỨU LIÊN QUAN

Trong phần này chúng tôi khảo sát các phương pháp tiếp cận đã được đề xuất cho việc so khớp ontology [3-10]. Những phương pháp tiếp cận này có thể được chia thành bốn loại: từ vựng, ngữ nghĩa, cấu trúc và tổ hợp.

Phương pháp tiếp cận về từ vựng là phương pháp dựa trên chuỗi để nhận dạng các thực thể tương đồng nhau trong các ontology đã cho. Phương pháp này có thể được dùng để nhận dạng các lớp tương đồng trong các ontology nguồn và ontology đích dựa trên sự giống nhau về nhãn hoặc về việc miêu tả của chúng [3]. Các kỹ thuật này xem các chuỗi có trình tự của các chữ cái. Chúng dựa trên sự nhận biết sau đây: các chuỗi giống nhau nhiều hơn, nhiều khả năng chúng biểu thị các khái niệm giống nhau. Việc so sánh các kỹ thuật so khớp chuỗi khác nhau, từ các vấn đề về khoảng cách đến các vấn đề dựa trên token có thể tìm thấy trong [4]. Một số ví dụ của các kỹ thuật dựa trên chuỗi được sử dụng rộng rãi trong các hệ thống so khớp là tiền tố, hậu tố, chỉnh sửa khoảng cách và n-gram.

Các phương pháp tiếp cận tiền tố và hậu tố của hai chuỗi đầu vào và kiểm tra xem chuỗi thứ nhất có bắt đầu (kết thúc) so với chuỗi thứ hai hay không. Phương pháp tiếp cận này hiệu quả trong việc so khớp các chuỗi có cùng nguồn gốc và các từ viết tắt tương tự nhau (ví dụ int và integer). Các phương pháp tiếp cận chỉnh sửa khoảng cách của hai chuỗi đầu vào được tính toán để chỉnh sửa khoảng cách giữa chúng. Chỉnh sửa khoảng cách là số các ký tự được chèn vào, xóa đi hay thay thế để chuyển đổi một chuỗi này thành một chuỗi khác, được chuẩn hóa theo chiều dài của chuỗi dài hơn. Ví dụ, MLMA+algorithm [9] sử dụng khoảng cách Levenshtein [6] để chỉnh sửa khoảng cách và tính toán sự tương đồng về từ vựng giữa hai thực thể. Phương pháp tiếp cận dựa trên N-gram của hai chuỗi đầu vào và tính toán số lượng n-grams (tức là trình tự của n ký tự) giữa chúng. Ví dụ, 3-grams của chuỗi ‘nikon’ là ‘nik’, ‘iko’, ‘kon’. Vậy, khoảng cách giữa ‘nkon’ và ‘nikon’ dựa trên 3-grams sẽ là 1/3.

Với phương pháp tiếp cận ngữ nghĩa theo cách thông thường là một hoặc nhiều tài nguyên về ngôn ngữ như từ vựng và từ điển chuyên ngành được sử dụng để xác định các thực thể đồng nghĩa [3]. Những phương pháp tiếp cận này thường sử dụng kiến thức phổ thông hoặc từ điển thuộc miền cụ thể để so khớp các từ dựa trên các mối quan hệ ngôn ngữ giữa chúng (ví dụ: các từ đồng nghĩa, các từ có nghĩa hẹp so với từ khái quát). Trong trường hợp này, tên các thực thể của ontology được xem như các từ của ngôn ngữ tự nhiên. Một số phương pháp tiếp cận sử dụng từ điển kiến thức phổ thông để có được ý nghĩa của các thuật ngữ sử dụng trong các ontology. Ví dụ, WordNet [7] là một cơ sở dữ liệu điện tử về từ vựng tiếng Anh (và các ngôn ngữ khác), trong đó các từ có nghĩa riêng biệt được đặt vào các bộ từ đồng nghĩa. Các quan hệ giữa các thực thể ontology có thể được tính toán liên quan đến các các ràng buộc về nghĩa trong WordNet [8-9]. Các phương pháp tiếp cận khác sử dụng từ điển về tên miền cụ thể thường lưu trữ một số kiến thức về miền cụ thể, nó không có sẵn trong bộ từ điển kiến thức phổ thông (ví dụ như tên riêng), như truy cập với từ đồng nghĩa, các từ có nghĩa hẹp so với từ khái quát và các mối quan hệ khác.

Các phương pháp tiếp cận về cấu trúc nhận dạng các lớp giống nhau (các nút) bằng cách quan sát các đối sánh của chúng với các lớp khác dựa vào ontology được đề cập và các thuộc tính của chúng nữa. Ý tưởng chính với hai lớp của ontology nguồn và ontology đích là tương đồng nếu chúng có những lân cận giống nhau (các cấu trúc) và các thuộc tính giống nhau [2]. Ví dụ, GMO là một thuật toán về cấu trúc, nó sử dụng đồ thị hai bên tách biệt (bipartite graphs) để miêu tả các ontology. Nó đo các đồ thị tương đồng về cấu trúc bởi một phép đo mới. Tuy nhiên, GMO có một tập các cặp đối sánh, chúng thường được tìm thấy trước bởi các phương pháp tiếp cận khác, với dữ liệu nhập vào từ bên ngoài trong quá trình so khớp của nó. Một số so khớp khác về cấu trúc được sử dụng để so sánh các ontology, chúng so khớp dựa trên các nút con, các nút lá và các mối quan hệ. Trong trường hợp đối sánh, hai thực thể không phải nút lá được xem là tương đồng nếu chúng có các nút con hoặc các nút lá lần lượt tương đồng nhau. Trong quan hệ đối sánh, việc tính toán giữa các nút tương đồng cũng có thể dựa vào các mối quan hệ (thuộc tính đối tượng) của chúng [10].

Các phương pháp tiếp cận về tổ hợp, chúng kết hợp hai hoặc nhiều các phương pháp tiếp cận nói trên (tức là sự kết hợp về từ vựng, ngữ nghĩa và phương pháp tiếp cận về cấu trúc) để có được kết quả tốt hơn. MLMA+algorithm [5] và phiên bản cải tiến của nó là tổ hợp các phương pháp tiếp cận, nó sử dụng một kỹ thuật tìm kiếm lân cận, nó thực hiện ở hai cấp độ. Ở cấp độ đầu tiên, đo sự tương đồng về hai từ vựng của hai ontology đầu vào. Trường hợp này là đo sự tương đồng theo tên, nó sử dụng khoảng cách Levenshtein [6] và đo sự tương đồng về từ vựng, nó sử dụng WordNet [7]. Ở cấp độ thứ hai, các thuật toán được áp dụng đo sự tương đồng về cấu trúc để tìm các giải pháp so khớp tốt nhất.

III. PHƯƠNG PHÁP TỔ HỢP ĐỂ ĐỐI SÁNH CÁC ONTOLOGY

Trong phần này chúng tôi miêu tả phương pháp tổ hợp của chúng tôi trong so khớp các ontology dựa trên sự tương đồng về từ vựng và sự tương đồng về cấu trúc của chúng. Phương pháp đề xuất này bao gồm ba giai đoạn để tìm các đối sánh giữa ontology nguồn và ontology đích. Ở giai đoạn thứ nhất và giai đoạn thứ hai, các ontology đầu vào được so khớp tương ứng về từ vựng và về cấu trúc. Sau đó, ở giai đoạn thứ ba các kết quả tiếp nhận từ hai giai đoạn trước được kết hợp lại để tạo ra kết quả tổng thể. Các chi tiết của ba giai đoạn này được giải thích trong các mục tiếp theo.

A. Các thực thể so khớp về từ vựng giữa hai ontology

Các phương pháp tiếp cận so khớp tương đồng về từ vựng là các phương pháp dựa trên chuỗi của các thực thể tương đồng được xác định trong các ontology đã cho. Ở đây, chúng tôi tìm kiếm sự tương đồng về từ vựng được tách biệt giữa các thực thể (các lớp được đặt tên, các thuộc tính đối tượng và các thuộc tính dữ liệu) của ontology nguồn và ontology đích. Vì vậy, giai đoạn này chúng tôi sẽ đưa ra ba ma trận riêng biệt tương đồng về từ vựng như là đầu ra của nó.

Chúng tôi giới thiệu độ đo mới tương đồng về khoảng cách để xác định sự tương đồng về từ vựng của các ontology đầu vào. Giả sử chúng tôi muốn tính toán sự tương đồng về từ vựng giữa chuỗi s và chuỗi t. Các chuỗi này có thể là một từ hoặc một văn bản chứa một số các phát biểu (statements). Lúc đầu, chúng tôi chuyển mỗi chuỗi ký tự

thành chuỗi các token bằng cách sử dụng các dấu phân cách, sau khi chuyển đổi thành các token sẽ đưa vào một túi từ. Bất kỳ ký tự trong chuỗi đã cho không thuộc bảng chữ cái sẽ được xem như là một dấu phân cách. Ví dụ, nếu chuỗi ký tự s chứa hai ký tự không thuộc bảng chữ cái thì chúng tôi xem hai ký tự như là hai dấu phân cách và loại bỏ chúng ra khỏi chuỗi s. Kết quả là, chuỗi s sẽ chuyển đổi thành ba token tức là ba từ. Mỗi chuỗi s và t sau khi được chuyển đổi thành các token sẽ cho vào mỗi túi từ tương ứng, mỗi từ mà chung cho hai túi sẽ bị loại bỏ khỏi hai túi. Sau đó, nếu không còn gì trong túi thứ nhất và túi thứ hai, khoảng cách giữa hai chuỗi đầu vào sẽ là zero. Mặt khác, tất cả các từ còn lại trong mỗi túi sẽ được kết nối và dẫn đến khoảng cách tương đồng Levenshtein [6] được tính toán giữa hai từ. Sau khi tính toán khoảng cách giữa chuỗi s và t, thì sự tương đồng của chúng sẽ là phương trình sau đây:

$$\text{Lexical\_Similarity(s,t)} = 1 - distance(s,t)/max\_len(s,t) \quad (1)$$

trong đó distance là khoảng cách giữa chuỗi s và t, và max_len là độ dài tối đa của chuỗi s và t.

Hãy xét ví dụ sau đây:

s = “Part Of”
t = “is_part_of”
bag_of_words(s) = {“Part”, “Of”}
bag_of_words(t) = {“is”, “part”, “of”}

Sau khi tạo các túi từ, chúng tôi loại bỏ hai từ “part” và “of” ra khỏi hai túi từ. Đến đây, chúng tôi sẽ có các túi sau:

bags_of_words(s) = {}
bags_of_words(t) = {“is”}

Cuối cùng, sự tương đồng giữa chuỗi s và t sẽ là: Levenshtein_distance(“”, “is”) = 2

$$Lexical\_similarity(s,t) = 1 - \left[\frac{distance(s,t)}{\max\_len(s,t)}\right] = 1 - \left[\frac{Levenshtein\_distance("","is")}{\max(7,10)}\right] = 1 - {}^{2}/_{10} = 0.80$$

Chúng tôi tính toán riêng biệt sự tương đồng về từ vựng trong số các lớp được đặt tên, các thuộc tính đối tượng và các thuộc tính dữ liệu của hai ontology đầu vào, sử dụng độ đo nói trên và sau đó tạo ra ba ma trận riêng biệt tương đồng về từ vựng.

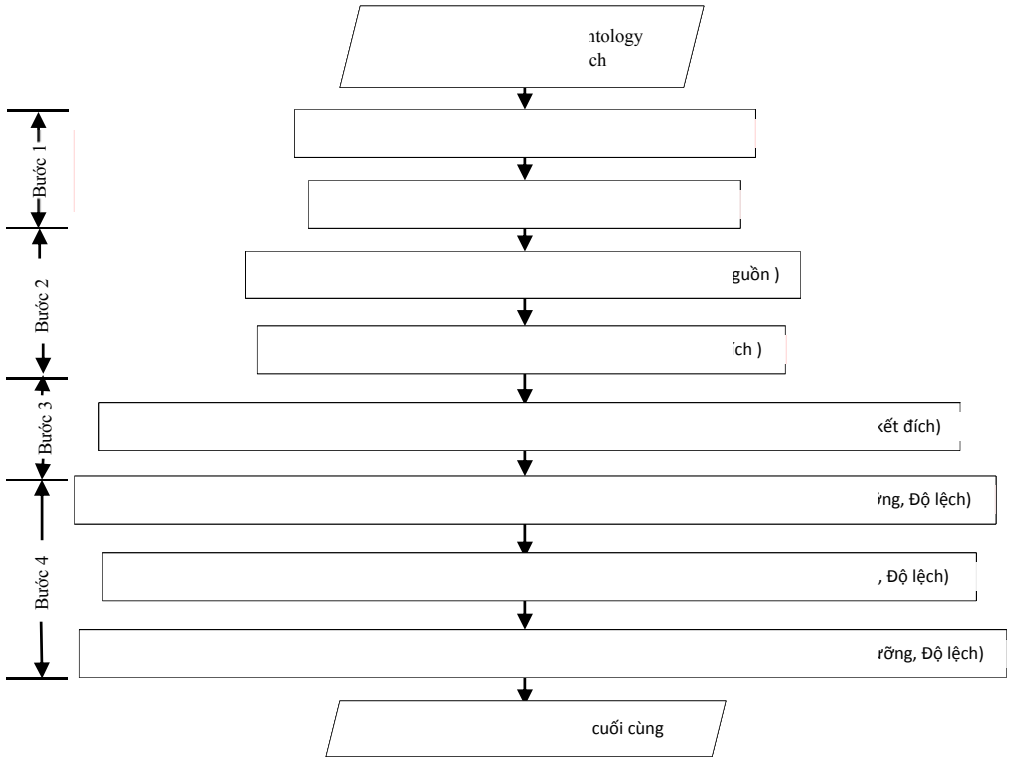
B. Các thực thể so khớp về mặt cấu trúc giữa hai ontology

Trong phần này, chúng tôi đưa ra phương pháp so khớp về mặt cấu trúc giữa hai ontology đầu vào, trong Hình 1 cho thấy sơ đồ của phương pháp này (có độ phức tạp thuật toán $O(n^2)$). Sau đây là các bước để tạo ra cấu trúc ma trận tương đồng giữa ontology nguồn và ontology đích:

1. Tạo một ma trận lân cận đối với mỗi ontology.
2. Tạo một dãy các danh sách liên kết đối với mỗi ontology dựa trên ma trận lân cận của nó.
3. Tính toán sự tương đồng về cấu trúc trong số các nút của ontology nguồn và ontology đích bằng cách sử dụng danh sách liên kết của chúng và tạo ra ma trận khởi tạo (ban đầu) tương đồng về cấu trúc.
4. Cải thiện ma trận khởi tạo tương đồng về cấu trúc bằng cách sử dụng ba thao tác bổ sung (bước 4, Hình 1).

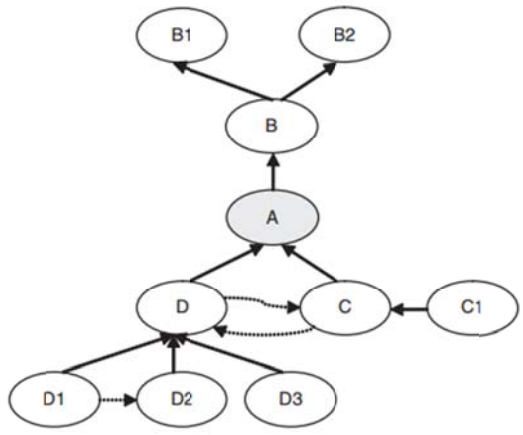
Các bước trên sau đây được mô tả một cách chi tiết. Để so sánh cấu trúc hai ontology đầu vào, chúng tôi tạo ra một mạng lưới gồm các nút của mỗi ontology. Những nút này sẽ được so sánh với mỗi nút của ontology khác dựa trên mạng lưới của chúng. Lưới này được mô phỏng bằng cách sử dụng một dãy các danh sách liên kết (tức là một dãy hai chiều). Điểm quan trọng của phương pháp này là số nút lân cận của một nút, cách thức chúng liên quan với nhau và với nút này (Hình 2).

Chúng tôi xem hai nút trong một ontology là lân cận nhau nếu chúng có liên quan với nhau thông qua quan hệ ‘is-a’ (lớp con hoặc lớp cha), ‘equivalent to’ hoặc ‘disjoint with’ hoặc thông qua quan hệ về thuộc tính đối tượng. Ví dụ, nếu lớp A là một lớp con của lớp A’, thì A và A’ là lân cận nhau. Một ví dụ khác, một đối tượng có p thuộc tính, các nút miền và vùng của nó sẽ là lân cận nhau (ví dụ nút A có hai thuộc tính D và C, hai thuộc tính này có quan hệ vùng miền, nên D và C là lân cận nhau). Dựa trên giả định này, chúng tôi tính toán ma trận lân cận cho một ontology bất kỳ. Mỗi phần tử của ma trận lân cận là 1 hoặc 0, nó cho thấy các nút tương ứng về hàng và cột của ma trận là có lân cận hay không (Hình 3).



Hình 1. Sơ đồ thuật toán tính toán sự tương đồng về cấu trúc.

Từ ontology nguồn hoặc ontology đích, chúng tôi tính toán mạng lưới của mỗi nút bằng cách sử dụng một ma trận lân cận của ontology đó. Mạng lưới của mỗi nút được miêu tả bởi một dãy các danh sách liên kết. Mỗi hàng trong ma trận lân cận của một ontology tương ứng với một nút của ontology đó. Số 1 trong mỗi hàng cho biết số nút lân cận với nó. Trường hợp nếu nút A trong ma trận lân cận có n nút lân cận thì dãy các danh sách liên kết tương ứng của nó sẽ có n hàng (theo Hình 2, nút A có 3 nút lân cận đó là nút B, nút C và nút D, nên dãy các danh sách liên kết tương ứng của nút A sẽ có 3 hàng, hàng 1 là của nút B, hàng 2 là của nút C và hàng 3 là của nút D, xem Hình 3b). Mỗi hàng trong dãy các danh sách liên kết này miêu tả các lân cận của nút đó.



Hình 2. Ví dụ ontology cho thấy cách thức so khớp sự tương đồng về cấu trúc

Cột đầu tiên của dãy các danh sách liên kết của nút A cho biết số lân cận của nút lân cận với A (ví dụ cột đầu tiên của ma trận trong Hình 3b gồm 3 3 5 nghĩa là nút B có 3 lân cận, nút C có 3 lân cận và nút D có 5 lân cận). Cột 2 trở đi cho biết số lân cận của mỗi lân cận tương ứng với lân cận ở hàng thứ j (với j = 1,2, ..., n) của dãy các danh sách liên kết của nút A (ví dụ cột 2 3 4 của hàng 1 tương úng là 1 1 3 nghĩa là lân cận thứ nhất của B là B1 và B1 này có 1 lân cận với nó, lân cận thứ hai của B là B2 và B2 này có 1 lân cận với nó, cuối cùng lân cận thứ ba của B là A và A này có 3 lân cận với nó).

Trong bước thứ ba của mục này, chúng tôi tính toán ma trận khởi tạo tương đồng về cấu trúc giữa các ontology đầu vào bằng cách sử dụng dãy các danh sách liên kết của chúng. Mỗi phần tử của ma trận khởi tạo tương ứng với mức độ tương đồng về cấu trúc giữa một nút của ontology nguồn và một nút của ontology đích. Chúng tôi so sánh nút A của

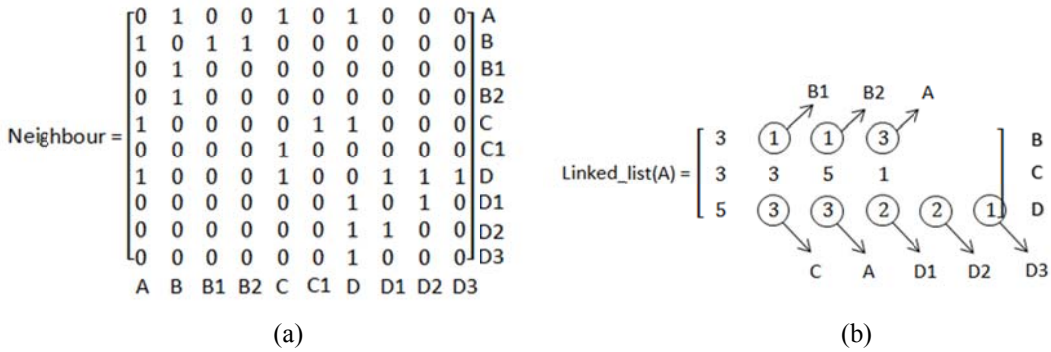
ontology nguồn với tất cả các nút A' của ontology đích với num_of_neighbors(A) – num_of_neighbors(A') ≤ 1. Với giải pháp thay thế, chúng tôi có thể so sánh tất cả các nút của ontology nguồn với tất cả các nút của ontology đích.

Để so sánh một nút A của ontology nguồn có n lân cận với một nút A' của ontology đích, chúng tôi tính toán khả năng so sánh có thể xảy ra là n + 1. Khả năng đầu tiên, được gọi là p_0, được tính toán dựa trên sự so sánh các lân cận của hai nút này (cột đầu tiên của mỗi dãy nút). Khả năng p_i (với i = 1, 2, ..., n) được tính bằng cách so sánh các lân cận của những lân cận này. Mỗi hàng i của dãy đầu tiên (nút A) được so sánh với tất cả các hàng của dãy thứ hai (nút A') và theo đó hàng so khớp tốt nhất có khả năng cao nhất được xem là khả năng p_i. Trong thực tế, p_i cho biết so khớp có lân cận tốt nhất với nút A' là lân cận thứ i của nút A. Khi tất cả các khả năng được tính toán, trung bình cộng của khả năng so sánh n + 1 này hình thành sự tương đồng về cấu trúc của hai nút.

Sau khi tất cả các phần tử của ma trận tương đồng được tính toán, ba thao tác dưới đây được sử dụng để tính toán cải thiện ma trận khởi tạo:

- Nếu hai nút từ ontology nguồn và ontology đích là tương đồng, thì sự tương đồng về lân cận của chúng sẽ được tăng lên bởi độ sai lệch (bias) được xác định trước.
- Nếu hai nút từ ontology nguồn và ontology đích có n nút lân cận được đối sánh, thì sự tương đồng của chúng đối với độ sai lệch được xác định trước sẽ được tăng lên bởi n × độ sai lệch.
- Nếu hai nút từ ontology nguồn và ontology đích có n thuộc tính với kiểu dữ liệu phổ biến, thì sự tương đồng của chúng đối với độ sai lệch được xác định trước sẽ được tăng lên bởi n × độ sai lệch.

Sau khi tiến hành một số thí nghiệm chúng tôi tìm được giá trị thích hợp cho yếu tố độ lệch là 0.1. Vì vậy chúng tôi sử dụng giá trị này trong các thí nghiệm của chúng tôi. Việc áp dụng giá trị ngưỡng (threshold) trên ma trận khởi tạo tương đồng về cấu trúc, chúng tôi xác định được các nút của ontology nguồn so khớp với các nút của ontology đích. Nói chung, trong quá trình tạo ma trận cuối cùng tương đồng về cấu trúc, bốn ma trận tương đồng khác nhau được tạo ra. Ma trận khởi tạo tương đồng về cấu trúc là ma trận đầu tiên và ba ma trận khác được tạo ra bằng cách áp dụng ba thao tác được xác định trên từng bước để cải thiện ma trận khởi tạo. Ma trận được tạo ra cuối cùng sẽ được xem là ma trận cuối cùng tương đồng về cấu trúc.



(a) (b)

Hình 3 (a) Ma trận lân cận, và (b) dãy các danh sách liên kết của nút A của ontology

C. Sự kết hợp về từ vựng và các kết quả tương đồng về cấu trúc

Trong mục 3.1 và 3.2 các chi tiết của phương pháp đề xuất cho việc tính toán những điểm tương đồng về từ vựng và cấu trúc giữa các thực thể của ontology nguồn và ontology đích đã được bàn đến.

Như đã trình bày, phương pháp đề xuất đưa ra ba ma trận tương đồng về từ vựng của các lớp được đặt tên, các thuộc tính đối tượng và các thuộc tính dữ liệu và ma trận tương đồng về cấu trúc cho các lớp được đặt tên. Bây giờ, chúng ta tìm hiểu việc tiếp nhận tất cả các kết quả tương đồng của các ma trận kết hợp này.

Để xác định các lớp tương đồng được đặt tên của ontology nguồn và ontology đích, chúng tôi lấy trung bình có trọng số của ma trận đầu tiên tương đồng về từ vựng và ma trận tương đồng về cấu trúc (tức là ma trận tương đồng của các lớp được đặt tên) như sau:

$$\text{NamedClasses\_Similarity} = \alpha \times Lexical\_NC\_Matrix + \beta \times Structural\_Matrix \Big/ \alpha + \beta \quad (2)$$

trong đó NamedClasses_Similarity là tất cả sự tương đồng trong số các lớp được đặt tên của ontology nguồn và ontology đích, Lexical_NC_Matrix là ma trận tương đồng về từ vựng của các lớp được đặt tên, Structural_Matrix là ma trận tương đồng về cấu trúc, α và β là các trọng số được gán tương ứng cho ma trận về từ vựng và ma trận về cấu trúc. Trong thí nghiệm của chúng tôi, nếu hai ontology đã cho với tương đồng về từ vựng nhiều hơn tương đồng về cấu trúc thì hệ số α và β sẽ có giá trị tương ứng là 0.6 và 0.4, trường hợp còn lại thì α = β = 0.6.

Ma trận thứ hai và thứ ba tương đồng về từ vựng tương ứng với các thuộc tính đối tượng và thuộc tính dữ liệu. Với mỗi cặp thuộc tính đối tượng hoặc thuộc tính dữ liệu của ontology nguồn và ontology đích, nếu chúng có giá trị tương đồng về từ vựng bằng hoặc lớn hơn 0.50 thì các thao tác dưới đây được sử dụng để cải thiện hai ma trận này:

- Nếu chúng có các miền được đối sánh, sự tương đồng của chúng sẽ được tăng lên bởi độ sai lệch được xác định trước.
- Nếu chúng có các vùng được đối sánh (hoặc tương đương với các thuộc tính dữ liệu), sự tương đồng của chúng sẽ được tăng lên bởi độ sai lệch được xác định trước.

Sau tác vụ này, các ma trận sẽ phản ánh mọi sự tương đồng tương ứng trong số các thuộc tính đối tượng và các thuộc tính dữ liệu. Chúng tôi dựa vào ma trận NamedClasses_Similarity để quyết định các miền và vùng của thuộc tính đối tượng và thuộc tính dữ liệu được đối sánh.

IV. VÍ DỤ MINH HỌA

Ví dụ minh họa trình bày trong phần này để làm rõ hơn phương pháp so khớp tương đồng về cấu trúc với thuật toán được đề xuất. Hãy xem xét ontology mẫu trong Hình 2.

Trong Hình 2 mũi tên liên tục hiển thị các mối quan hệ ‘is-a’, ‘equivalent’ hay ‘disjoint’ và các mũi tên không liên tục hiển thị các quan hệ về thuộc tính đối tượng. Khi ontology có 10 nút, ma trận lân cận của ontology này sẽ là một ma trận 10 × 10, xem Hình 3(a). Việc sử dụng ma trận lân cận để tính toán, một dãy các danh sách liên kết được tạo ra đối với mỗi nút để chỉ ra các lân cận với nó và những lân cận của các lân cận đó và các kết nối của chúng. Với ma trận lân cận trình bày trong Hình 3(a), ví dụ, hàng đầu tiên tương ứng với nút A của ontology. Nó cho thấy nút A có ba nút lân cận B, C và D tương ứng với ba hàng trong dãy danh sách liên kết. Hình 3(b) cho thấy đây là dãy danh sách liên kết được tính toán đối với nút A. Cột đầu tiên của dãy danh sách liên kết này hàm ý rằng ba nút B, C và D tương ứng có số nút lân cận là 3 3 và 5. Hàng đầu tiên của dãy danh sách liên kết này hàm ý rằng lân cận đầu tiên của nút A đó là nút B, nút B này có ba lân cận đó là nút B1, B2 và A, với mỗi nút này, có số nút lân cận tương ứng là 1 1 và 3. Tương tự cho hàng thứ hai và thứ ba của dãy danh sách liên kết này.

Bây giờ, nếu đem so sánh cấu trúc nút A của ontology nguồn với một nút A’ của ontology đích, chúng tôi cần phải tính toán bốn khả năng so sánh p_i có thể xảy ra (với i = 0, 1, 2, 3). Để tính toán p_0, cột đầu tiên của dãy danh sách liên kết của nút A được so sánh với cột đầu tiên của dãy danh sách liên kết của nút A’ của ontology đích. Về cơ bản đây là sự so sánh về cấu trúc của hai nút dựa trên các lân cận của chúng. Để tính p_i (với i = 1, 2, 3), hàng thứ i (không bao gồm phần tử đầu tiên) của dãy danh sách liên kết của A được so sánh với tất cả các hàng (không bao gồm các phần tử đầu tiên) của dãy danh sách liên kết của nút A’, hàng so khớp tốt nhất với dãy danh sách liên kết của nút A’ sẽ xác định giá trị của p_i. Cuối cùng, với mức trung bình của bốn khả năng so sánh được tính toán để xác định sự tương đồng về cấu trúc giữa nút A của ontology nguồn và nút A’ của ontology đích.

Sự tương đồng về cấu trúc trong số tất cả các nút của ontology nguồn và tất cả các nút của ontology đích tạo thành ma trận khởi tạo tương đồng về cấu trúc. Sau khi tính toán ma trận này, ba thao tác được giới thiệu trong mục 3.2 được áp dụng để tạo ra ma trận cuối cùng tương đồng về cấu trúc.

V. CÁC THÍ NGHIỆM VÀ KẾT QUẢ

Việc đánh giá hiệu suất của các thuật toán so khớp ontology đối với các ontology thử nghiệm và các kết quả so sánh, người ta đưa ra hai độ đo precision and recall có nguồn gốc từ truy hồi thông tin [2].

Định nghĩa 1 (Đối sánh - Alignment)

Cho hai ontology O và O’, một đối sánh giữa O và O’ là một tập các bộ tương ứng (gồm 4 phần tử): <e, e’, r, n> với e ∈ O và e’ ∈ O’ của hai thực thể so khớp nhau, r là mối quan hệ ràng buộc giữa e và e’ và n là độ tin cậy [0..1] tương ứng.

Thuật toán so khớp trả về đối sánh A, nó được so sánh với đối sánh tồn tại R.

Ví dụ trong Hình 4 trình bày hai ontology cùng với hai đối sánh A và R. Vì mục đích của việc đơn giản hóa, mối quan hệ luôn luôn là ‘=’ và độ tin cậy luôn luôn là 1.0.

đối sánh A được xác định như sau:

<o1:Vehicle,o2:Thing,=,1.0>
<o1:Car,o2:Porsche,=,1.0>
<o1:hasSpeed,o2:hasProperty,=,1.0>
<o1:MotorKA1,o2:MarcsPorsche,=,1.0>
<o1:250kmh,o2:fast,=,1.0>

đối sánh tồn tại (R) được xác định như sau:

<o1:Object,o2:Thing,=,1.0>
<o1:Car,o2:Automobile,=,1.0>
<o1:Speed,o2:Characteristic,=,1.0>
<o1:250kmh,o2:fast,=,1.0>
<o1:PorscheKA123,o2:MarcsPorsche,=,1.0>

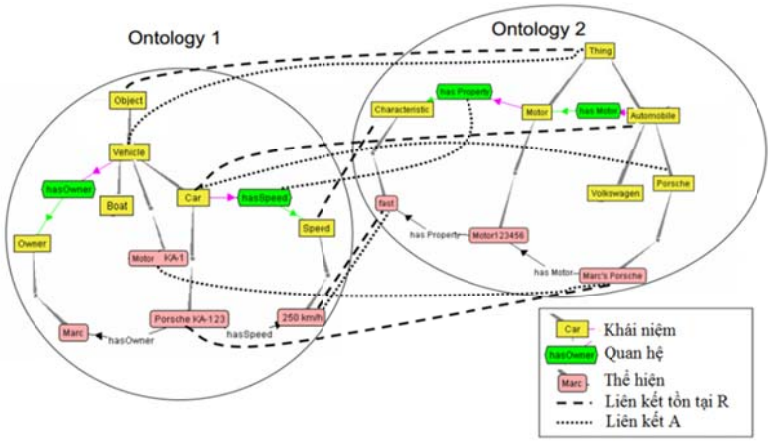
Định nghĩa 2 (precision, recall)

Với đối sánh tồn tại R, độ đo precision của đối sánh A cho bởi:

$$P(A,R) = \frac{|R \cap A|}{|A|}$$

và độ đo recall cho bởi:

$$R(A,R) = \frac{|R \cap A|}{|R|}$$



Hình 4. Hai ontology đối sánh nhau

Chúng tôi đánh giá hiệu suất thuật toán do chúng tôi đề xuất bằng việc sử dụng hệ số đặc trưng. Về các kết quả đối sánh, chúng tôi so sánh các kết quả của chúng tôi với sáu thuật toán được trình bày trong Ontology Alignment Evaluation Initiative 2008 (OAEI-08) và cùng với MLMA + algorithm. OAEI là một hoạt động thường niên cho các hệ thống so khớp ontology để xác định điểm mạnh và điểm yếu của chúng. Đây là một sáng kiến quốc tế phối hợp tổ chức đánh giá về quá trình phát triển của các hệ thống so khớp ontology. Mục tiêu chính của nó là so sánh các hệ thống và các thuật toán trên cơ sở tương đồng và cho phép bất kỳ ai đưa ra kết luận về các chiến lược so khớp tốt nhất. Sự đánh giá của các tổ chức đưa ra bộ thử nghiệm chuẩn có hệ thống với cặp ontology đối sánh cũng như các kết quả dự kiến. Các ontology này được mô tả bởi OWL-DL và được định dạng bằng RDF/XML. Các đối sánh dự kiến đưa ra định dạng chuẩn được diễn tả bằng RDF/XML. Chúng tôi đã phát triển một công cụ dựa trên thuật toán được đề xuất và áp dụng nó vào bộ thử nghiệm chuẩn OAEI-08. Chúng tôi sử dụng các số liệu thống kê chuẩn về truy hồi thông tin để đánh giá các kết quả thử nghiệm:

$$precision = \frac{|R \cap A|}{|A|} = \frac{số\ bộ\ tương\ ứng\ giữa\ R\ và\ A}{số\ bộ\ của\ Liên\ kết\ A} \tag{3}$$

$$recall = \frac{|R \cap A|}{|R|} = \frac{số\ bộ\ tương\ ứng\ giữa\ R\ và\ A}{số\ bộ\ của\ Liên\ kết\ tồn\ tại\ R}$$

$$f - measure = \frac{2 \times precision \times recall}{precision + recall}$$

Chúng tôi phân loại các trường hợp thử nghiệm vào năm nhóm: đó là các nhóm sau #101-104, #201-210, #221-247, #248-266 và #301-304. Các giá trị trung bình về độ đo của precision, recall và f-measure nhận được từ mỗi nhóm bằng cách sử dụng thuật toán đã đề xuất thể hiện trong Bảng 1.

Bảng 1. Hiệu suất trung bình thuật toán được đề xuất trên bộ thử nghiệm chuẩn OAEI-08

	#101–104	#201–210	#221–247	#248–266	#301–304	Trung bình
Precision	0.98	0.96	0.93	0.46	0.87	0.84
Recall	0.95	0.92	0.88	0.41	0.84	0.80
F-measure	0.96	0.93	0.90	0.43	0.85	0.81

Chúng tôi sử dụng công cụ Jena để phân tích các ontology của bộ thử nghiệm chuẩn OAEI-08 và xét giá trị ngưỡng th = 0.70 cũng như độ sai lệch bias = 0.1. Trong trường hợp thử nghiệm #101-104, dữ liệu về từ vựng và cấu trúc phù hợp, vì vậy chúng tôi đã nhận được kết quả tốt nhất. Trong trường hợp thử nghiệm #201-210 ontology nguồn và ontology đích có cấu trúc tương đồng và trong trường hợp thử nghiệm #221-247 dữ liệu về từ vựng phù hợp giữa ontology nguồn và ontology đích. Vì vậy, chúng tôi có kết quả tốt trong trường hợp thử nghiệm này. Với trường hợp

thử nghiệm #248-266 dữ liệu về từ vựng và cấu trúc không đầy đủ, chúng tôi đã thu được các kết quả không thích hợp. Trong trường hợp thử nghiệm #301-304 với bốn ontology, cùng với thuật toán được đề xuất, nó cho ra kết quả tốt. Chúng tôi so sánh nhiều cấp độ và cách tiếp cận so khớp ontology tổ hợp của chúng tôi với một số hệ thống như CIDER, DSSim, GeRoMe, MapPSO, SPIDER, TaxoMap, gồm những người tham gia bộ thử nghiệm chuẩn OAEI-08 cùng với MLMA + algorithm [5]. Các kết quả của sáu thuật toán trên bộ thử nghiệm chuẩn OAEI-08. Kết quả so sánh được trình bày trong Bảng 2.

Bảng 2. So sánh các giá trị trung bình của precision và recall bởi phương pháp tiếp cận của chúng tôi với một số hệ thống đã tham gia vào tổ chức OAEI-08

System test	TaxoMap		MapPSO		GeRoMe		SPIDER		CIDER		DSSim		MLMA+		Our approach	
	Prec	Rec.	Prec	Re.	Prec	Rec.	Prec	Rec.	Pre.	Rec	Prec	Rec.	Pre.	Rec.	Pre.	Rec.
1xx	1.0	0.34	0.92	1.0	0.96	0.79	0.99	0.99	0.99	0.99	1.0	1.0	0.91	0.89	0.98	0.95
2xx	0.95	0.21	0.48	0.53	0.56	0.52	0.97	0.57	0.97	0.57	0.97	0.64	0.57	0.52	0.78	0.74
3xx	0.92	0.21	0.49	0.25	0.61	0.40	0.15	0.81	0.90	0.75	0.90	0.71	0.68	0.65	0.87	0.84
Average	0.91	0.22	0.51	0.54	0.60	0.58	0.81	0.63	0.97	0.62	0.97	0.67	0.69	0.65	0.86	0.83
F-measure	0.35		0.52		0.58		0.70		0.75		0.79		0.66		0.84	

Các thử nghiệm chuẩn trên hệ thống được phân thành ba loại: 1xx, 2xx và 3xx. Bảng 2 cho thấy giá trị trung bình precision và recall của từng loại, trung bình tổng (hay trung bình điều hòa) và f-measure của ba loại này. Xem phương trình (3).

Khi tham khảo từ các kết quả được trình bày trong Bảng 2, thuật toán đề xuất của chúng tôi có độ đo f-measure tốt hơn các hệ thống khác và hàm ý rằng nó hiệu quả hơn các hệ thống khác. Thuật toán đề xuất cũng đã đạt được độ đo recall tốt hơn so với các hệ thống khác. Nhưng, nó có độ đo precision thấp hơn các hệ thống TaxoMap, CIDER và DSSim. Tuy nhiên, các hệ thống này gần như đạt tới độ đo recall và độ đo f-measure không thích hợp trong tất cả các thuật toán. Trong thực tế chúng đã loại bỏ độ đo recall để có được độ đo precision tốt hơn.

VI. CÔNG CỤ OMREASONER

A. Trình bày về hệ thống

So khớp ontology tìm kiếm sự tương ứng giữa các thực thể liên quan đến ngữ nghĩa của các ontology. Nó đóng một vai trò quan trọng trong nhiều miền ứng dụng.

Các phương pháp so khớp ontology đã được đề xuất: việc thực hiện so khớp có thể sử dụng nhiều thuật toán so khớp hoặc các công cụ đối sánh, và các tiêu chí phân loại chủ yếu sau đây được xem xét [11-13].

Nhiều phương pháp tập trung vào các khía cạnh cú pháp thay thế cho ngữ nghĩa. OMReasoner thực hiện việc so khớp bởi quy trình sử dụng một số từ điển bên ngoài và các kỹ thuật suy diễn. Tuy nhiên, phương pháp này bao gồm chiến lược của việc phối hợp (chủ yếu cú pháp) nhiều công cụ so khớp (ví dụ, công cụ so khớp EditDistance).

1. Định nghĩa và phân tích hệ thống

Quá trình so khớp có thể được định nghĩa là một hàm f.

$A' = f(O_1, O_2, A, p, r)$

Trong đó O_1 và O_2 là một cặp của các ontology như là đầu vào để đối sánh, A là đối sánh đầu vào giữa các ontology và A' là đối sánh mới ở đầu ra giữa các ontology, p là một tập các thông số (ví dụ, trọng số w và ngưỡng τ) và r là một tập các nguồn tài nguyên.

Các đối sánh biểu thị sự tương ứng giữa hai thực thể. Một tương ứng phải thể hiện hai thực thể và mối quan hệ giữa chúng. Cho hai ontology, một sự tương ứng là bộ 5 phần tử: <id, e_1, e_2, R, n>, trong đó:

- id là một định danh duy nhất của sự tương ứng;
- e1 và e2 là các thực thể của ontology thứ nhất và ontology thứ hai tương ứng;
- R là một quan hệ (ví dụ, tương đồng (=), lớn hơn (>), nhỏ hơn (<), không tương đồng (⊥)) giữa e1 và e2. Theo OAEI, quan hệ tương đồng là cốt lõi;
- n là độ tin cậy (thường trong khoảng [0 1]) với sự tương ứng giữa e1 và e2.

OMReasoner thực hiện đối sánh ontology với ba bước như sau (Hình 5):

1. Phân tích cú pháp: chúng tôi có thể thu được các lớp và các thuộc tính của các ontology bằng cách sử dụng API ontology: Jena.

2. Kết hợp giữa các công cụ so khớp riêng lẻ: tương đồng về từ có thể được sinh ra bằng cách sử dụng nhiều thuật toán so khớp hoặc các công cụ đối sánh, ví dụ phương pháp tương đồng về chuỗi (tiền tố, hậu tố, chỉnh sửa khoảng cách) bằng cách dựa trên chuỗi, các kỹ thuật dựa trên ràng buộc. Trong khi đó, một số ngữ nghĩa tương ứng có

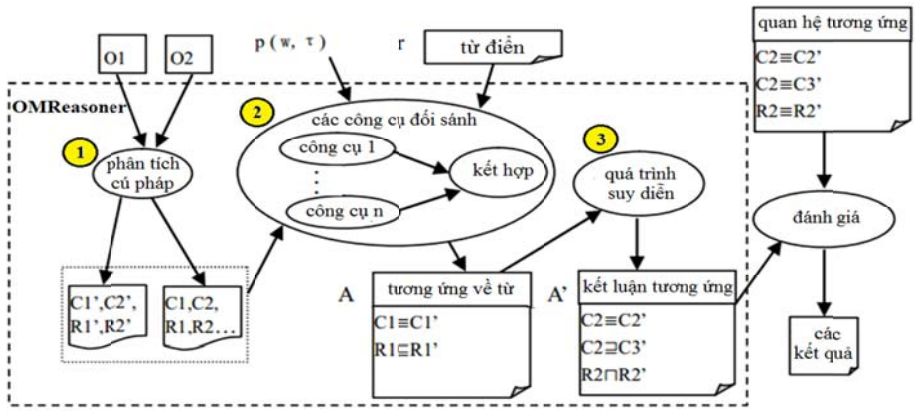
thể thực hiện bằng cách sử dụng từ điển bên ngoài như WordNet. Sau đó, nhiều kết quả so khớp được kết hợp bằng cách sử dụng chiến lược cụ thể.

Khung ứng dụng sẽ hỗ trợ về việc kết hợp các công cụ đối sánh, tạo điều kiện thuận lợi cho các công cụ so khớp riêng lẻ.

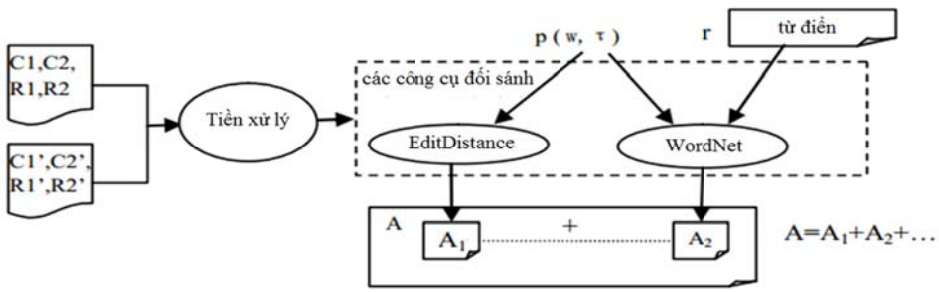
3. Quá trình suy diễn: ngữ nghĩa tương ứng có thể được suy diễn bằng cách sử dụng logic mô tả DL (Description Logic), trong đó các tương đồng về từ được sinh ra ở bước 2 được xem là đầu vào.

Cuối cùng, chúng tôi đánh giá các kết quả dựa vào các đối sánh liên quan, và tính toán hai độ đo: precision và recall.

Với OMReasoner, khung ứng dụng rất linh hoạt đối với các công cụ so khớp riêng lẻ. Hiện nay, nhiều công cụ so khớp riêng lẻ bao gồm EditDistance và WordNet (Hình 5).



Hình 5. So khớp ontology trong OMReasoner



Hình 6. Minh hoạ về các công cụ so khớp trong OMReasoner

2. Các kỹ thuật sử dụng cụ thể

a) Ngưỡng

Ngưỡng rất cần thiết đối với nhiều công cụ so khớp (đặc biệt là cú pháp) về sự tương đồng. Ví dụ, khoảng cách chỉnh sửa của "book" và "booklet" là 3/7 (tức là, các độ đo tin cậy tương đồng là 1-3/7=0.57). Nếu ngưỡng là 0.55, thì hai thực thể là tương đồng (với độ do tin cậy 0.57); ngược lại nếu ngưỡng là 0.6, thì chúng không tương đồng. Vì vậy, chúng tôi phải điều chỉnh công cụ so khớp thông qua ngưỡng.

b) Kết hợp độ đo tin cậy

Mỗi công cụ so khớp riêng lẻ có thể tạo ra các độ đo tin cậy tương ứng. Tất cả các độ đo tin cậy này sẽ được chuẩn hoá trước khi kết hợp. OMReasoner bao gồm các chiến lược linh hoạt sau đây để kết hợp các kết quả đối sánh:

- Thuật toán tổng hợp trọng số (WeightSum)

Độ tin cậy có thể được tổng hợp bằng thuật toán tương đồng về trọng số (công thức 1), trong đó w_k là trọng số cho một công cụ so khớp k cụ thể và $sim_k(e1, e2)$ là độ tin cậy của sự tương đồng.

$$sim(e_1, e_2) = \sum_{k=1}^{n} w_k \times sim_k(e_1, e_2), \text{ trong đó } \sum_{k=1}^{n} w_k = 1.0 \quad (1)$$

- Phương pháp cực đại (Max)

Độ đo tin cậy cực đại được chọn trong số n công cụ so khớp (công thức 2).

$$sim(e1, e2) = max(sim1(e1, e2),..., simn(e1, e2)) \quad (2)$$

c) So khớp ngữ nghĩa

OMReasoner sử dụng các phương pháp so khớp ngữ nghĩa như công cụ so khớp WordNet và việc suy diễn bởi logic mô tả (DL - Description Logic).

WordNet là một cơ sở dữ liệu điện tử về từ vựng tiếng Anh, trong đó các nghĩa khác nhau (các nghĩa có thể là một từ hay cụm từ) của các từ được sắp xếp tạo thành các bộ từ đồng nghĩa. Các quan hệ giữa các thực thể ontology được tính toán với các thuật ngữ ràng buộc về nghĩa của WordNet. Công cụ so khớp riêng lẻ này sử dụng một từ điển bên ngoài như WordNet để đạt được sự tương ứng về ngữ nghĩa.

OMReasoner sử dụng logic mô tả DL được cung cấp bởi Jena. OMReasoner bao gồm các luật suy diễn về việc so khớp ontology. Tuy nhiên, khả năng suy diễn mất nhiều thời gian và chỉ góp một phần nhỏ cho các kết quả. Trong phiên bản này, khả năng suy diễn được bỏ qua.

B. Kết quả của OMReasoner theo từng phương pháp thực hiện

Trong phần này, chúng tôi trình bày các kết quả đạt được từ OMReasoner với OAEI 2014. Nó thực hiện theo ba phương pháp: Benchmark, Conference và MultiFarm. Các thử nghiệm được tiến hành trên một máy tính đang chạy Windows Server 2008 R2 Standard với bộ vi xử lý Intel Core i5 chạy ở 2.8 Ghz và 16 GB RAM.

1. Phương pháp Benchmark

Với phương pháp này, các ontology có thể được chia thành 3 loại (Bảng 3). Trong nhóm 1, thông tin từ vựng đã được thay đổi để thay thế các nhãn hoặc định danh về chúng. Sự thay đổi này bao gồm việc thay thế các nhãn hoặc các định danh với các tên khác theo một quy ước đặt tên cụ thể, một tên ngẫu nhiên, một tên sai chính tả hoặc một từ nước ngoài. Trong nhóm 2 có các ontology thu hẹp hệ thống phân cấp, mở rộng hệ thống phân cấp hoặc tất cả đều không có phân cấp. Trong nhóm 3 các ontology được thách thức lớn nhất về đối sánh ontology. Ở đây các nhãn được trộn sao cho phép hoán vị của các từ có chiều dài cụ thể. Chúng tôi điều chỉnh công cụ bằng cách sử dụng ngưỡng T (τ_{wd}: ngưỡng của WordNet, τ_{ed}: ngưỡng của EditDitance) và kết hợp chiến lược S, sau đó nhận được các kết quả tốt hơn (τ_{wd} = 0.95, τ_{ed} = 0.9; S = Max). Các kết quả đạt được từ OMReasoner theo Benchmark được tóm tắt trong Bảng 4.

Bảng 3. Phân loại theo chuẩn 2014

phân loại	khái niệm	hệ thống phân loại ontology			ontology thực tế
các thử nghiệm	101-104	201-210	221-247	248-266	301-304

Bảng 4. Các kết qủa đạt được theo Benchmark 2014

	101-104	201-210	210-247	248-266	301-304	H-mean
precision	0.898	0.675	0.820	0.637	0.925	0.791
recall	1.000	0.414	1.000	0.517	0.437	0.647
F-measure	0.946	0.491	0.898	0.555	0.574	0.694

2. Phương pháp Conference

Tập dữ liệu tin cậy bao gồm các ontology thực tế. Chúng tôi sử dụng chiến lược kết hợp để thực thi công cụ hệ thống của chúng tôi theo phương pháp Conference. Các kết quả đạt được từ OMReasoner được tóm tắt trong Bảng 5 (τ_{wd} = 0.9, τ_{ed} = 0.8; S = Max).

Bảng 5. Kết quả đạt được theo Conference 2014

thử nghiệm	precision	recall	F-measure
Conference	0.778	0.518	0.647

3. Phương pháp MultiFarm

Phương pháp MultiFarm bao gồm một tập con của tập dữ liệu kết hợp, được dịch với tám ngôn ngữ khác nhau. Với phương pháp này, các ontology có thể được chia thành 2 loại. Trong nhóm 1 các đối sánh ontology đều giống nhau. Trong nhóm 2 các đối sánh ontology đều khác nhau.

Trước hết, chúng tôi sử dụng từ điển để dịch các ngôn ngữ khác nhau sang tiếng Anh. Sau đó, tiếng Anh đã được dịch sẽ đưa vào các công cụ so khớp bằng cách sử dụng chiến lược Max. Cuối cùng chúng tôi nhận được các kết quả. Chúng tôi điều chỉnh công cụ của chúng tôi bằng cách sử dụng ngưỡng và các kết quả có thể hiển thị trong Bảng 6 (τ_{wd} = 0.8, τ_{ed} = 0.6; S = Max), trong đó cho thấy các độ đo F-Measures của các đối sánh ontology ở nhóm 2 là rõ ràng kém hơn so với các đối sánh ontology trong nhóm 1. Chúng tôi thấy rằng những lý do mà OMReasoner không được thiết kế tốt để so khớp với các ontology khác là vì chúng được viết bằng các ngôn ngữ hoàn toàn khác nhau.

Bảng 6. Các kết quả đối với MultiFarm 2014

Trường hợp thử nghiệm	precision	recall	F-measure
Nhóm 1: Các ontology giống nhau	0.955	0.800	0.853
Nhóm 2: Các ontology khác nhau	0.584	0.438	0.471

Để chọn ngưỡng tốt hơn, chúng tôi so sánh các kết quả (Bảng 7) trên một số ngưỡng theo phương pháp Conference. Tuy nhiên, chúng tôi vẫn sử dụng chiến lược về phương pháp Max để thực hiện công cụ của chúng tôi.

Từ các kết quả, chúng tôi thấy rằng khi ngưỡng $\tau_{wd} = 0.9$, $\tau_{ed} = 0.8$, công cụ của chúng tôi thực hiện tốt nhất (F-measure = 0.647). Vì vậy mà chúng tôi sử dụng ngưỡng $\tau_{wd} = 0.9$, $\tau_{ed} = 0.8$ theo phương pháp Conference. Việc sử dụng phương pháp Conference, chúng tôi nhận được các ngưỡng tốt hơn so với phương pháp Benchmark và MultiFarm.

Bảng 7. So sánh kết quả với các ngưỡng khác nhau của Conference 2014

Ngưỡng		precision	recall	F-measure
τ_{wd}	τ_{ed}			
0.8	0.8	0.782	0.508	0.599
0.95	0.8	0.787	0.466	0.580
0.9	0.8	0.778	0.518	0.647
0.9	0.9	0.796	0.476	0.580

C. Nhận xét chung

1. Thảo luận về cách thức để cải thiện hệ thống đề xuất

Thực hiện việc suy diễn dựa trên các tương ứng về từ là rất khó khăn, vì vậy các kết quả chính xác được đưa ra từ các công cụ so khớp riêng lẻ sẽ nâng cao các kết quả của chúng tôi. Một số cách để cải thiện công cụ của chúng tôi được liệt kê như sau:

a) Áp dụng nhiều chiến lược linh hoạt hơn trong việc kết hợp nhiều công cụ so khớp thay vì chỉ tổng hợp dựa trên trọng số.

b) Thêm một số tiền xử lý (Hình 6), chẳng hạn như loại bỏ đặc tính cụ thể (ví dụ, '-', '_') hoặc tách các từ ghép, trước khi đưa vào các công cụ đối sánh.

c) Lấy các nhận xét và thông tin về nhãn của ontology để tính toán, cá biệt khi tên của khái niệm này là vô nghĩa.

d) Xem xét lại việc sử dụng giá trị ngưỡng thích hợp để tối ưu hóa độ chính xác.

e) Một vấn đề khác trong công cụ của chúng tôi là bỏ qua thông tin về cấu trúc bao gồm ontology ở giai đoạn hiện nay. Và chúng tôi sẽ cải thiện nó trong tương lai.

2. Đề xuất các biện pháp mới

Chúng tôi thấy rằng OMReasoner có thể cải tiến rất nhiều. Một số cách mới được đề xuất như sau:

a) Làm phong phú các từ điển ngữ nghĩa vì WordNet không phải là một từ điển chuyên nghiệp, nó không thể có được các khái niệm ngữ nghĩa toàn diện.

b) Tính đến sự phân cấp các khái niệm ngữ nghĩa thay vì chỉ tính đến tất cả các khái niệm và thuộc tính.

c) Tìm các từ đồng nghĩa theo phương pháp kết hợp.

d) Tìm các từ điển ngôn ngữ khác nhau cho MultiFarm.

e) Cải thiện thuật toán của một số công cụ đối sánh.

f) Bao gồm nhiều công cụ so khớp khác nhau.

VII. KẾT LUẬN

Trong bài báo này chúng tôi trình bày thuật toán so khớp ontology tìm ra sự tương đồng trong số các thực thể của các ontology đã cho dựa trên thông tin về từ vựng và cấu trúc của chúng. Thuật toán này thực hiện ở ba giai đoạn: từ vựng, cấu trúc, và tổ hợp. Đối với việc xác định sự tương đồng về từ vựng giữa các thực thể, chúng tôi giới thiệu một phép đo mới về sự tương đồng, trong đó các thông tin về từ vựng của mỗi thực thể, chẳng hạn như nhãn hoặc sự miêu tả, được chuyển đổi và cho vào một túi từ, sau đó chúng được sử dụng cho việc tìm kiếm sự tương đồng của các thực thể. Trong giai đoạn đầu tiên, chúng tôi thu được ba ma trận tương đồng về từ vựng bằng cách so sánh các lớp được đặt tên, các thuộc tính đối tượng và các thuộc tính dữ liệu của hai ontology. Trong giai đoạn thứ hai, để so sánh cấu trúc các ontology, chúng tôi tạo ra một mạng lưới cho mỗi nút trong ontology nguồn và ontology đích và sau đó so sánh chúng với nhau dựa trên mạng lưới của chúng. Mỗi mạng lưới của mỗi nút được tạo ra bằng cách sử dụng các lân cận của nút đó và các lân cận của các lân cận đó đồng thời được thể hiện bởi một mảng hai chiều. Ma trận khởi tạo tương đồng về cấu trúc được tính bằng cách so sánh các mảng này. Sau khi tạo ra ma trận này, nó được cải thiện bằng cách áp dụng ba thao tác được mô tả trong phần III mục B. Cuối cùng, trong giai đoạn thứ ba, chúng tôi tính toán giá trị trung bình có trọng số của các kết quả về từ vựng và cấu trúc. Chúng tôi đã thực hiện thuật toán của chúng tôi trên bộ thử nghiệm chuẩn của OAEI-08 và có các kết quả khả quan. Ngoài ra chúng tôi so sánh thuật toán của chúng tôi với một số hệ thống đã tham gia vào tổ chức OAEI-08 và như trong Bảng 2 cho thấy thuật toán của chúng tôi có độ đo f-measre tốt hơn. Ngoài ra, chúng tôi trình bày thêm các kết quả của hệ thống OMReasoner cho việc đối sánh các ontology theo ba phương pháp: Benchmark, Conference và MultiFarm. Chiến lược kết hợp của nhiều công cụ so khớp

riêng lẻ và sự suy diễn logic mô tả DL bao hàm cả trong cách tiếp cận của chúng tôi. Các kết quả đạt được chúng tôi thấy vẫn chưa thỏa mãn và sẽ tiếp tục cải tiến nó trong tương lai.

VIII. TÀI LIỆU THAM KHẢO

[1] N. Arch-Int and P. Sophatsathit, A semantic information gathering approach for heterogeneous information sources on WWW, Journal of Information Science 29 (2003) 357–374.

[2] M. Ehrig and J. Euzenat. Relaxed precision and recall for ontology matching, K-Cap 2005 Workshop on Integrating Ontologies2005 (Banff, Alberta, Canada) 25–32.

[3] L. S. Xiao and R. Ellen, Automated schema mapping techniques: an exploratory study, Research Letters Information Science4 (2003) 113–136.

[4] W. Cohen, P. Ravikumar and S. Fienberg, A comparison of string metrics for matching names and records, Proceedings of the Workshop on Data Cleaning and Object Consolidation at the International Conference on Knowledge Discovery and Data Mining (KDD)(2003).

[5] A. Alasoud, V. Haarslev and N. Shiri, An empirical comparison of ontology matching techniques, Journal of Information Science35(4) (2009) 379–397.

[6] V. I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals, Soviet Physics Doklady10 (1966) 707–710.

[7] G. A. Miller, WordNet: A lexical database for english, Communications of the ACM38 (1995) 39–41.

[8] P. Bouquet, L. Serafini and S. Zanobini, Peer-to-peer semantic coordination, Journal of Web Semantics 2(1) (2004) 81–97.

[9] G. Pirro, A semantic similarity metric combining features and intrinsic information content, Journal of Data and Knowledge Engineering 68 (2009) 1289–1308.

[10] A. Maedche and S. Staab, Measuring similarity between ontologies, In Proceedings of the International Conference on Knowledge Engineering and Knowledge Management(2002) 251–263.

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

[12] Shvaiko, P. and Euzenat, J.: A survey of schema-based matching approaches. Journal on Data Semantics (JoDS) IV, 146--171(2005).

[13] Kalfoglou, Y. and Schorlemmer, M.: Ontology mapping: the state of the art. The Knowledge Engineering Review Journal, 18(1):1--31, (2003).

A NEW ALGORITHM FOR ONTOLOGY MATCHING

Huynh Nhut Phat, Hoang Huu Hanh, Phan Cong Vinh

ABSTRACT *– Ontology matching is an importance in ontology technology of the Semantic Web with a goal of finding alignments among the entities of given ontologies. Ontology matching is a necessary step for establishing interoperation and knowledge sharing among Semantic Web applications. In this study we present an algorithm and a tool developed based on this algorithm to find correspondences among entities of input ontologies. The proposed algorithm uses a new lexical similarity measure and also utilizes structural information of ontologies to determine their corresponding entities. The lexical similarity measure generates a bag of words for each entity based on its label and description information. The structural approach creates a grid for each node in the ontologies. The combination of lexical and structural approaches creates the similarity matrix between the source and target ontologies. The proposed algorithm was tested on a well known benchmark and also compared to other algorithms presented in the literature. Our experimental results show the proposed algorithm is effective and outperforms other algorithms.*