

Trường Đại học Bách Khoa Hà Nội

Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

Phát triển các tính năng tạo,
trực quan hóa và tìm kiếm
trên dữ liệu ngũ nghĩa trên
hệ thống tổng hợp thông tin
thể thao BKSport

Lê Huỳnh Đức

Hà Nội, 05/2019

Trường Đại học Bách Khoa Hà Nội
Viện Công nghệ Thông Tin và Truyền Thông

Đồ án Tốt nghiệp Đại học

Phát triển các tính năng tạo, trực quan hóa và tìm kiếm trên dữ liệu ngũ nghĩa trên hệ thống tổng hợp thông tin thể thao BKSport

Sinh viên thực hiện Lê Huỳnh Đức

Người hướng dẫn PGS/Ts. Cao Tuấn Dũng

Hà Nội, 05/2019

Lời cam kết

Họ và tên sinh viên: Lê Huỳnh Đức

Điện thoại liên lạc: 0339112123 Email: huynhduc96@gmail.com

Lớp: IS2 K59 Hệ đào tạo: Đại Học

Tôi – *Lê Huỳnh Đức* – cam kết Đồ án Tốt nghiệp (ĐATN) là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *PGS.TS Cao Tuân Dũng*. Các kết quả nêu trong ĐATN là trung thực, là thành quả của riêng tôi, không sao chép theo bất kỳ công trình nào khác. Tất cả những tham khảo trong ĐATN – bao gồm hình ảnh, bảng biểu, số liệu, và các câu từ trích dẫn – đều được ghi rõ ràng và đầy đủ nguồn gốc trong danh mục tài liệu tham khảo. Tôi xin hoàn toàn chịu trách nhiệm với dù chỉ một sao chép vi phạm quy chế của nhà trường.

Hà Nội, ngày 27 tháng 5 năm 2019

Tác giả ĐATN

Lê Huỳnh Đức

Lời cảm ơn

‘Tuổi thanh xuân giống như một con mưa rào, dù cho bạn từng bị cảm lạnh vì tăm mưa thì bạn vẫn muốn được đắm mình trong con mưa ấy lần nữa.’ – Trích You Are The Apple Of My Eye.

Với tôi, Bách Khoa đã là cả thanh xuân !

Người ta cũng nói rằng, phải khi sắp rời xa một thứ gì đó, ta mới cảm thấy thêm yêu, thêm trân trọng nó biết chừng nào. Và giờ đây, khi thời gian được mang trên mình ‘Sinh viên Bách Khoa’ chỉ còn lại những ngày tháng ngắn ngủi, tôi mới hiểu được điều đó. 5 năm của tuổi thanh xuân gắn liền với Bách Khoa, là từng phòng học nhà D9, từng bước chân lên Tầng 8 Thư Viện Tạ Quang Bửu đã như là ngôi nhà thân thương nhất trong trái tim mình. Và cũng 5 năm ấy, bao kỷ niệm là niềm vui gặp gỡ bạn bè, là những giọt mồ hôi bên trang giấy, là những lần học nhóm, động viên nhau trong những đợt thi cuối kỳ. Tất cả đã làm cho tuổi thanh xuân của tôi trở nên thật đẹp, thật ý nghĩa, một thanh xuân mà tôi đã tự hào vì được trải qua.

Và trước ngưỡng cửa cuối cùng của Bách Khoa, để hoàn thành được đồ án này, em xin được gửi lời cảm ơn chân thành nhất đến PGS.TS Cao Tuấn Dũng đã dịu dắt, hướng dẫn em rất tận tình. Đồng thời em cũng xin được cảm ơn anh Nguyễn Hoàng Công và anh Lê Vĩnh Thiện đã bên cạnh chỉ dạy và động viên em để em có thể hoàn thành được đồ án này.

Em cũng xin được cảm ơn các thầy cô giáo trong trường Đại học Bách Khoa Hà Nội cùng các thầy cô trong Viện Công Nghệ Thông Tin và Truyền Thông đã dạy dỗ, truyền đạt cho em những kiến thức bổ ích trong suốt 5 năm qua, cho em nền tảng quý báu để em bước vào sự nghiệp và cuộc sống.

Và cuối cùng, em xin chân thành cảm ơn gia đình và bạn bè, thầy cô đã luôn tạo điều kiện, giúp đỡ, động viên em trong suốt quá trình học tập và hoàn thành đồ án.

Tóm tắt

Web ngữ nghĩa đang trở thành xu thế nghiên cứu và phát triển trong những năm gần đây. Bằng Web ngữ nghĩa, không chỉ đơn thuần là con người đọc, lưu trữ và sử dụng tin tức mà còn giúp máy tính có thể hiểu được, tạo ra nền tảng để con người và máy móc có thể giao tiếp và làm việc với nhau. Năm 2015 được coi là năm bắt đầu của Semantic Web. Nhóm nghiên cứu Semantic Innovation Group do PGS.TS Cao Tuấn Dũng đã xây dựng và phát triển hệ thống tổng hợp tin tức trong lĩnh vực thể thao BKSport. Tư tưởng của hệ thống này là sử dụng Web ngữ nghĩa làm nền tảng với các chức năng như: Tạo ngữ nghĩa, quản lý và khai thác chủ đề ngữ nghĩa được sinh ra. Trong nghiên cứu này, tôi tập trung trình bày thực hiện một số nội dung sau :

- Phát triển, cải thiện tính năng sinh chủ đề ngữ nghĩa trong module Semantic Annotation.
- Phát triển, nâng cao độ chính xác của module Semantic Search giúp chuyển câu hỏi dạng ngôn ngữ tự nhiên sang ngôn ngữ truy vấn ngữ nghĩa SPARQL.
- Thêm tính năng trực quan hóa dữ liệu ngữ nghĩa và hiển thị cho người dùng.

Giải pháp tôi đề ra là bộ các giải pháp bao gồm: Làm giàu cơ sở tri thức, nâng cao độ chính xác trong việc xác định chủ đề trong văn bản và câu hỏi, xử lý nhận diện và sinh ngữ nghĩa cho một số quan hệ mới, mở rộng thêm một số lớp câu hỏi mới, đánh dấu, gắn tag cho dữ liệu ngữ nghĩa để hiển thị cho người dùng.

Tiến hành thực nghiệm giải pháp hệ thống trên một bộ câu hỏi với nhiều lớp câu hỏi, nhiều văn bản kết quả cho thấy giải pháp mà tôi đã thực hiện đã nâng cao độ chính xác của toàn bộ hệ thống.

Mục lục

Lời cam kết	iii
Lời cảm ơn	iv
Tóm tắt	v
Mục lục	vi
Danh mục hình vẽ.....	ix
Danh mục bảng.....	x
Danh mục các từ viết tắt.....	xi
Chương 1 Giới thiệu đề tài	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài	3
1.3 Định hướng giải pháp.....	3
1.4 Bố cục đồ án	3
Chương 2 Cơ sở lý thuyết và công nghệ sử dụng	5
2.1 Cơ sở lý thuyết	5
2.1.1 Semantic Web	5
2.1.2 Ontology	6
2.1.3 Chú thích ngữ nghĩa	7
2.1.4 RDF/RDFS và OWL.....	7
2.1.5 Cây cấu trúc ngữ pháp	9

2.1.6 Ngôn ngữ SPARQL	10
2.1.7 Ngôn ngữ lập trình JAPE	11
2.2 Các công nghệ và platform sử dụng	11
2.2.1 Giới thiệu về KIM Platform	11
2.2.2 Allegrograph	12
2.2.3 Ruby on Rails Framework	13
Chương 3 Hoàn thiện hệ thống tổng hợp tin tức thể thao BKSport với khả năng xử lý ngữ nghĩa	14
3.1 Tổng quan về hệ thống tích hợp tin tức BKSport	14
3.2 Chức năng tạo dữ liệu ngữ nghĩa trong hệ thống tin tức BKSport	16
3.2.1 Kiến trúc Module Semantic Annotation đã đề xuất trước đây	16
3.2.2 Phát triển tính năng sinh ngữ nghĩa cho Module Semantic Annotation ..	20
3.3 Chức năng chuyển đổi câu hỏi từ ngôn ngữ tự nhiên sang truy vấn SPARQL	25
3.3.1 Kiến trúc module Semantic Search đã đề xuất trước đây	25
3.3.2 Phát triển tính năng sinh câu hỏi từ ngôn ngữ tự nhiên của Module Semantic Search	29
3.4 Phát triển tính năng trực quan hóa dữ liệu ngữ nghĩa trong hệ thống tích hợp thông tin BKSport	30
3.4.1 Kiến trúc tổng thể Module Web cung cấp giao diện người dùng	30
3.4.2 Thiết kế chi tiết Module Web	33
Chương 4 Các giải pháp và đóng góp nổi bật.....	37
4.1 Làm giàu và cập nhật cơ sở tri thức	37
4.1.1 Bổ sung dữ liệu về các thực thể	38
4.1.2 Bổ sung dữ liệu về các Ontology	40

4.2 Bổ sung nhận biết các quan hệ mới trong Module Semantic Annotation.....	43
4.2.1 Nhận biết quan hệ Trích dẫn gián tiếp	43
4.2.2 Nhận biết quan hệ Kết quả trận đấu và sinh ngữ nghĩa đối lập	46
4.2.3 Nhận biết quan hệ là biến thể của dạng S – V – O	48
4.3 Bổ sung nhận biết các dạng câu hỏi mới trong Module Semantic Search	50
4.3.1 Lớp câu hỏi về trích dẫn	50
4.3.2 Lớp câu hỏi Kết hợp hoặc Chọn lựa	54
4.3.3 Lớp câu hỏi về một người với một sự kiện và câu hỏi so sánh hơn	58
4.4 Trực quan hóa hiển thị dữ liệu ngữ nghĩa	59
Chương 5 Kết quả thực nghiệm và Đánh giá	63
5.1 Kết quả thực nghiệm về khả năng sinh chú thích ngữ nghĩa	63
5.2 Kết quả thực nghiệm về khả năng sinh câu hỏi và khả năng trả lời câu hỏi	65
5.3 Kết quả xây dựng trang web tổng hợp tin tức BKSport phiên bản 2.0 (Module Web giao diện người dùng).....	66
5.3.1 Kết quả về khả năng tìm kiếm ngữ nghĩa.	67
5.3.2 Kết quả về tính năng trực quan hóa dữ liệu ngữ nghĩa.	72
Chương 6 Kết luận và hướng phát triển	76
Tài liệu tham khảo	78

Danh mục hình vẽ

Hình 1 : Ví dụ về tìm kiếm kết quả không ra theo ý muốn	2
Hình 2: Ví dụ về một mẫu quan hệ bộ ba trong RDF	8
Hình 3: Sơ đồ tổng quan kiến trúc hệ thống BKSport	14
Hình 4: Kiến trúc Module Semantic Annotation	16
Hình 5: Mô hình quan hệ - thực thể(E-R) trong thiết kế CSDL module Crawler	21
Hình 6: Sơ đồ kiến trúc module Semantic Search	25
Hình 7: Sơ đồ kiến trúc module web giao diện người dùng	31
Hình 8: Sơ đồ thiết kế các gói trong module Web giao diện người dùng	33
Hình 9: Trang tin Transfermarkt	38
Hình 10: Sơ đồ làm giàu cơ sở tri thức	39
Hình 11: Mô hình hóa thuật toán xử lý lớp câu hỏi	53
Hình 12: Mô hình hóa thuật toán gắn nhãn cho dữ liệu ngữ nghĩa	60
Hình 13: Ví dụ về trang web xem tin thông thường	74
Hình 14: Ví dụ về trang web sau khi được trực quan hóa thực thể	74
Hình 15: Ví dụ về trực quan hóa quan hệ bộ ba ngữ nghĩa	75

Danh mục bảng

Bảng 1: Các từ khóa xuất hiện phổ biến trong các mối quan hệ thể thao	19
Bảng 2: Ánh xạ quan hệ từ bảng Player trong Database sang thuộc tính ngữ nghĩa của Player trong cơ sở tri thức	21
Bảng 3: Ánh xạ quan hệ từ bảng Manager trong Database sang thuộc tính ngữ nghĩa của Manager trong cơ sở tri thức.....	22
Bảng 4: Ánh xạ quan hệ từ bảng Club trong Database sang thuộc tính ngữ nghĩa của Club trong cơ sở tri thức	22
Bảng 5: Ánh xạ quan hệ từ bảng Stadium trong Database sang thuộc tính ngữ nghĩa của Stadium trong cơ sở tri thức	22
Bảng 6: Ánh xạ quan hệ từ bảng Stadium trong Database sang thuộc tính ngữ nghĩa của Stadium trong cơ sở tri thức	22
Bảng 7: Ánh xạ quan hệ từ các thuộc tính trong database với thuộc tính ngữ nghĩa.....	22
Bảng 8: Các thuộc tính ngữ nghĩa bổ sung vào Ontology lớp Player	42
Bảng 9: Các thuộc tính ngữ nghĩa bổ sung vào Ontology lớp Event -1	42
Bảng 10: Các thuộc tính ngữ nghĩa bổ sung vào Ontology lớp Event -2	43
Bảng 11: Thực nghiệm kết quả sinh câu hỏi từ ngôn ngữ tự nhiên sang ngôn ngữ truy vấn SPARQL và khả năng trả lời câu hỏi của hệ thống	66

Danh mục các từ viết tắt

API	Application Programming Interface Giao diện lập trình ứng dụng
Dataset	Bộ dữ liệu
Service	Dịch vụ
HTML	HyperText Markup Language Ngôn ngữ đánh dấu siêu văn bản
Module	Mô đun
Crawl	Thu thập thông tin
Offset	Vị trí bắt đầu tương đối
Database	Cơ sở dữ liệu

Chương 1 Giới thiệu đề tài

1.1 Đặt vấn đề

Ngày nay, trong bối cảnh toàn cầu hóa với sự phát triển vượt bậc của khoa học kĩ thuật, có nhiều dịch vụ công nghệ truyền thông ra đời nhằm đáp ứng nhu cầu ngày càng cao của con người. Một trong những nhu cầu lớn hiện tại là nhu cầu tìm kiếm thông tin. Với đại đa số người dùng, việc nắm bắt tin tức hàng ngày là vô cùng lớn và ngày càng tăng. Từ đó, các website tin tức ngày càng nhiều, lượng truy cập ngày một tăng cao, đem lại lượng tin tức và tri thức lớn cho người dùng.

Cùng với sự phát triển cao của các web tin tức thì lượng tin tức càng ngày càng tăng cao. Vấn đề đặt ra lúc này là làm sao để khai thác tin tức này sao cho hiệu quả, bởi nếu không được lưu trữ và sử dụng thì lượng kiến thức đó sẽ rơi vào quên lãng. Và trong đó, em đề cập 2 vấn đề quan trọng nhất là:

1. *Tìm kiếm tin tức*

Hiện tại, việc tìm kiếm tin tức ở hầu hết các website đều dừng lại ở mức độ tìm kiếm theo từ khóa – và đây cũng là phương pháp tìm kiếm đơn giản và phổ biến nhất. Tuy nhiên để tìm kiếm bằng từ khóa, phải đảm bảo chính xác nội dung nhập vào. Đồng thời khi không tìm thấy kết quả mong muốn, người dùng buộc phải có một số kỹ năng nhất định để xử lý việc tìm kiếm. Ví dụ chỉ tìm kiếm trên một trang web nào đó thì thêm từ khóa site:<tên trang web>, ...

2. *Lưu trữ và sử dụng dữ liệu*

Cũng tương tự như vấn đề tìm kiếm tin tức, việc lưu trữ tin tức hiện nay hầu hết là lưu trữ dưới dạng thông tin dạng text hoặc html thuần túy. Do đó việc khai thác thông tin này là vô cùng khó, đặc biệt là khi có nhu cầu khai thác và xử lý thông tin thì việc lưu trữ thông tin chỉ dưới dạng text hay html tỏ ra vô cùng kém hiệu quả.

[Tất cả](#)[Tin tức](#)[Hình ảnh](#)[Video](#)[Thêm](#)[Cài đặt](#)[Công cụ](#)

Khoảng 1.920.000 kết quả (0,65 giây)

Sân vận động Bách Khoa - Facebook

<https://vi-vn.facebook.com/pages/Sân-vận-động-Bách-Khoa/222906877780352>

Sân vận động Bách Khoa, Hanoi, Vietnam. 1.1K likes. Stadium, Arena & Sports Venue.

Sân vận động Bách Khoa - Hanoi, Vietnam - Stadium, Arena & Sports ...

<https://www.facebook.com> › ... › Hanoi, Vietnam › Stadium, Arena & Sports Venue

Phone, +84 4 3868 0186 · Address. Lê Thanh Nghị, Hanoi, Vietnam 100000 ... Tuan Pham Anh Giò này vẫn ra đá dc thi đúng là đam mê cháy rực rõ rồi bạn tôi 🎉 1 tiếng đồng hồ ở mặt đường các b cảm thấy thế nào có mát như thẳng trong ...

Sân vận động Bách Khoa - 37, Lê Thanh Nghị, P. Bách Khoa, Q. Hai ...

<https://map.cococ.com/map/4018733809545973>

★★★★★ Xếp hạng: 4,3 - 3 đánh giá

Sân vận động Bách Khoa - Giải trí. 37, Lê Thanh Nghị, P. Bách Khoa, Q. Hai Bà Trưng, Tp. Hà Nội. Đánh giá của người dùng, Chỉ đường, Thêm đánh giá.

Bị thiếu: mở giờ

Sân vận động Hàng Đẫy – Wikipedia tiếng Việt

https://vi.wikipedia.org/wiki/Sân_vận_động_Hàng_Đẫy

Bách khoa toàn thư mở Wikipedia ... Sân vận động Hàng Đẫy là một sân vận động nằm ở đường Trịnh Hoài Đức, Hà Nội, Việt Nam với sức chứa khoảng 22 500 chỗ ngồi. thời huy hoàng sân Hàng

Hình 1 Ví dụ về tìm kiếm kết quả không ra theo ý muốn

3. Hiển thị tin tức

Việc hiển thị tin tức hiện tại cũng đang tỏ ra kém hiểu quả trong việc tóm gọn nội dung tin tức. Hầu hết các bài viết hiện nay có nội dung khá dài, tuy nhiên lượng tin tức khá ít, ta có thể tóm tắt lượng tin tức đó trong một vài dòng. Vậy nên gây ra lãng phí thời gian và công sức của người đọc.

BKSport là một hệ thống tổng hợp tin tức trong lĩnh vực thể thao do PSG.TS Cao Tuấn Dũng cùng nhóm Semantic Innovation Group phát triển nhằm một phần giải quyết các vấn đề trên. Phương pháp đã được công bố với tựa đề “A Method for the Generation of Semantic Annotation from Sport News Using Ontology Based Patterns- Quang-Minh NGUYEN, Tuan-Dung CAO, Thanh-Hien PHAN, Hoang-Cong NGUYEN and Tatsuya HAGINO”. Với cách tiếp cận là sử dụng Web ngữ nghĩa làm nền tảng, BKSport cung cấp các khả năng như: Tạo-Sinh chủ thích ngữ nghĩa, quản lý thông tin, bài toán gợi ý, bài toán tìm kiếm tin tức...

Trong BKSport, kết quả bước đầu hệ thống đã có thể sinh chủ thích ngữ nghĩa, trích xuất ra các thông tin quan trọng, đồng thời có thể chuyển đổi được một số lớp câu hỏi tự nhiên sang

truy vấn SPARQL. Tuy vậy, hệ thống vẫn cần khắc phục một số hạn chế còn tồn tại như: Cơ sở tri thức còn ít và đã cũ, các quan hệ nhận diện còn ít và chưa chính xác, các lớp câu hỏi còn chưa bao quát hết nhu cầu của người dùng.

1.2 Mục tiêu và phạm vi đề tài

Mục tiêu đồ án:

- Phát triển, cải thiện khả năng sinh chú thích ngữ nghĩa trong module sinh chú thích ngữ nghĩa
- Phát triển, mở rộng thêm các lớp câu hỏi và nâng cao độ chính xác trong việc sinh câu hỏi từ ngôn ngữ tự nhiên sang ngôn ngữ truy vấn ngữ nghĩa SPARQL.
- Trực quan hóa hiển thị nội dung của dữ liệu ngữ nghĩa cho người dùng

Phạm vi nghiên cứu của đồ án nằm trong lĩnh vực thể thao đặc biệt là lĩnh vực bóng đá. Các dữ liệu, tập tri thức, tập câu hỏi được sử dụng ở ngôn ngữ tiếng anh.

1.3 Định hướng giải pháp

Trong đồ án này, em đi theo phương pháp mà nhóm Semantic Innovation Group đã tiếp cận. Với 2 Module là Module sinh chú thích ngữ nghĩa – Semantic Annotation và Module chuyển đổi câu hỏi tự nhiên sang câu hỏi truy vấn ngữ nghĩa – Semantic Search em lần lượt xem xét mô hình kiến trúc của hệ thống đã thực hiện từ trước. Phương pháp này hiện tại đã có đạt được những thành công nhất định. Tuy nhiên cũng còn nhiều hạn chế về vùng thu nhận thông tin và vùng câu hỏi có thể chuyển đổi. Do đó đồ án này sẽ tập trung giải quyết những vấn đề được nêu ra nhằm nâng cao độ chính xác và phong phú trong việc sinh chú thích ngữ nghĩa cũng như việc mở rộng lớp câu hỏi có thể chuyển đổi được. Sau đó tiến hành thực nghiệm để so sánh kết quả đạt được của hệ thống trước và sau khi tiến hành tối ưu hóa.

Đồng thời trong đồ án này em cũng đề xuất phương án gắn nhãn để thực hiện trực quan hóa hiển thị cho người dùng.

1.4 Bố cục đồ án

Bố cục của đồ án được trình bày như sau:

Chương 2 trình bày về một số lý thuyết và công nghệ sử dụng trong đồ án, cụ thể là lý thuyết về: web ngữ nghĩa, lý thuyết về ontology, chủ thích ngữ nghĩa, ngôn ngữ RDF/RDFS và OWL, ngôn ngữ truy vấn ngữ nghĩa SPARQL và ngôn ngữ đặc tả JAPE. Các công nghệ bao gồm KIM Platform, Allegrograph và Ruby on Rails.

Chương 3 trình bày về kiến trúc hệ thống BKSport và Kiến trúc Module Semantic Annotation, Semantic Search đã được xây dựng trước đây, đồng thời các đề xuất để phát triển các module này. Tiếp đó là đề xuất nhằm trực quan hóa ngữ nghĩa hiển thị cho người dùng.

Chương 4 trình bày các đóng góp nổi bật của bản thân đối với hệ thống bao gồm các mô hình và thuật toán xử lý nhằm nâng cao hiệu quả với hệ thống

Chương 5 trình bày kết quả và thực nghiệm của các phương pháp đã thực hiện. Trong đó so sánh kết quả đạt được của hệ thống trước và sau khi được cải thiện. Từ các kết quả đã đạt được nêu ra điểm tốt và điểm còn hạn chế, từ đó chỉ ra các bước để phát triển, tối ưu trong tương lai.

Chương 6 trình bày về kết luận của đồ án, những nhiệm vụ em đã hoàn thành, đồng thời nêu hướng phát triển trong tương lai.

Chương 2 Cơ sở lý thuyết và công nghệ sử dụng

2.1 Cơ sở lý thuyết

Việc nâng cao độ chính xác của nhận diện ngữ nghĩa và tạo câu hỏi cần một số lý thuyết về Web ngữ nghĩa cũng như các thành phần khác tạo nên hệ thống như ngôn ngữ SPARQL, Ontology, các ngôn ngữ mô tả Ontology... Sau đây em xin được trình bày sơ lược về khái niệm cũng như các kiến thức cơ bản về các cơ sở lý thuyết được sử dụng trong đồ án.

2.1.1 Semantic Web

Semantic Web hay còn gọi là Web ngữ nghĩa. Đây là một dự án mã nguồn mở trên lĩnh vực website được khởi xướng từ tổ chức W3C. Semantic Web định nghĩa các chuẩn dữ liệu chung trên World Wide Web. Semantic Web hướng việc chuyển đổi các web hiện tại từ các văn bản không có cấu trúc hoặc bán cấu trúc thành dữ liệu có ngữ nghĩa. Semantic Web được tạo từ ngôn ngữ RDF thuộc W3C.

Theo W3C thì Semantic Web cung cấp một framework chung. Framework này cho phép dữ liệu được chia sẻ và tái sử dụng thông qua các ứng dụng, các công ty và cộng đồng.

Theo Tim Berners-Lee, Semantic Web được định nghĩa là một sự mở rộng Web hiện tại. Nó xây dựng các dữ liệu một cách hợp lý, cho phép máy tính và con người làm việc với nhau. [1]

Một số ứng dụng của sử dụng công nghệ Web Ngữ nghĩa như:

- Search Engine: Hiện tại, các công cụ search engine hầu hết là Keyword Search Engine. Tuy nhiên, nếu sử dụng Web ngữ nghĩa với nền tảng là cơ sở tri thức thì việc search sẽ dựa vào khái niệm, các mối liên quan đến các thực thể, từ đây kết quả tìm kiếm sẽ chính xác và thông minh hơn rất nhiều lần
- FrameWork quản lý tri thức: Thúc đẩy khả năng tìm kiếm tri thức với độ chính xác cao, tăng khả năng truy cập câu tạo các nguồn tri thức cần thiết cho việc giải quyết một vấn đề nào đó.

2.1.2 Ontology

1. Khái niệm

Trong khoa học máy tính, một ontology là một mô hình dữ liệu biểu diễn khái niệm về một lĩnh vực. Ontology cũng được sử dụng để mô tả và biểu diễn về các đối tượng trong lĩnh vực đó và mối quan hệ giữa chúng. Ontology cung cấp một bộ từ vựng bao gồm các khái niệm, các thuộc tính quan trọng và các định nghĩa về các thuộc tính này. Ngoài ra, ontology còn cung cấp các ràng buộc, đôi khi các ràng buộc này được coi như các giả định cơ sở về ý nghĩa mong muốn của bộ từ vựng. Nó được sử dụng để con người cung cấp cơ sở dữ liệu để máy có thể hiểu.

Theo **Gruber, Thomas R** [2] một ontology là "một đặc tả chính thức và rõ ràng của một khái niệm chung".

Theo **Arvidsson, F.; Flycht-Eriksson** [3] một ontology cung cấp một ngữ nghĩa chung (semantic), đi kèm với các đối tượng và/hoặc các khái niệm tồn tại và các mối quan hệ, các thuộc tính của các đối tượng đó.

Tóm lại, ontology gồm những khái niệm về một lĩnh vực cụ thể và các mối quan hệ giữa chúng. Một ontology về một lĩnh vực sẽ mô tả rõ ràng những thực thể giúp con người và máy tính có thể hiểu và suy luận được theo ngữ nghĩa trong phạm vi lĩnh vực đó.

Ontology là một thành phần quan trọng của Semantic Web trong việc giúp máy tính có thể hiểu được những gì mà con người biểu diễn bằng ngôn ngữ thông thường thông qua tập từ vựng đã định nghĩa. Các vai trò chính của ontology như:

- Hình thành ngôn ngữ chung để chia sẻ, tái sử dụng tri thức
- Giúp giao tiếp giữa các ứng dụng tốt hơn
- Là nền tảng để biểu diễn tri thức

2. Các thành phần của Ontology

Một Ontology bao gồm các thành phần sau:

- Các cá thể (Individuals) - Thể hiện
- Các lớp (Classes) - Khái niệm
- Các thuộc tính (Properties)
- Các mối quan hệ (Relation)

2.1.3 Chú thích ngữ nghĩa

Theo [4], Chú thích ngữ nghĩa là quy trình gắn thông tin bổ sung vào các khái niệm khác nhau (như con người, đồ vật, địa điểm, tổ chức...) trong văn bản hoặc bất kỳ nội dung nào khác được đưa ra. Các chú thích ngữ nghĩa được con người cung cấp cho máy tính các máy tính sử dụng để tham chiếu.

Khi tài liệu được chú thích ngữ nghĩa thì nó trở thành nguồn dữ liệu để các máy tính truy vấn, sử dụng và suy luận.

Ví dụ, để chú giải về ngữ nghĩa các khái niệm được chọn trong câu “Manchester defeat Chelsea last night”, máy tính sẽ được hiểu Manchester, Chelsea là một thực thể, thực thể này là một đội bóng chứ không phải con người. Để hiểu được đội bóng là gì ta lại phải định nghĩa cho máy tính hiểu. Tương tự như thế cho “defeat”, “last night”... để máy tính có thể hiểu được vấn đề. Từ đây máy tính có thể suy luận và đưa ra kết quả cho các câu hỏi không chỉ là trực tiếp...

2.1.4 RDF/RDFS và OWL

1. RDF và RDFS

Nội dung thông tin Web được phục vụ chủ yếu cho con người, và máy móc không thể đọc và hiểu được nội dung này. Do đó, rất khó để tự động hóa bất cứ nội dung nào trên Web, ít nhất trên quy mô lớn. Hơn nữa, với lượng thông tin khổng lồ trên Web, chúng ta không thể xử lý chúng chỉ bằng phương pháp thủ công. Vì vậy, W3C đề xuất một giải pháp để mô tả dữ liệu trên Web và có thể được hiểu bởi máy móc, đó chính là RDF.

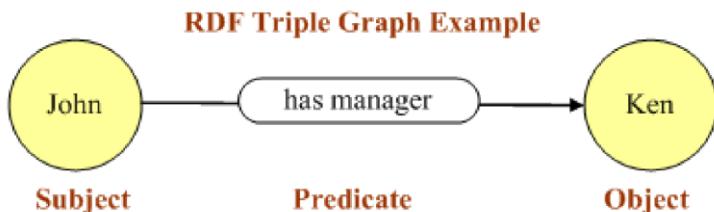
RDF (viết tắt từ **Resource Description Framework**, tạm dịch là **Framework Mô tả Tài nguyên**) có nguồn gốc tạo ra từ đầu năm 1999 bởi tổ chức W3C như là 1 tiêu chuẩn để mã hóa siêu dữ liệu (metadata).

RDF sử dụng một mô hình trừu tượng để phân rã thông tin/kiến thức thành những mảnh con, với 1 số luật cơ bản về ngữ nghĩa cho các mảnh này. Mục tiêu là cung cấp 1 phương thức chung mà đủ đơn giản và linh hoạt để diễn giải bất kỳ sự thật (fact) nào, nhưng có cấu trúc để các ứng dụng máy tính có thể hiểu và diễn giải cấu trúc đó.

Mô hình trừu tượng gồm các bộ 3 triples:

- Nguồn tài nguyên subject (chủ ngữ): URI
 - Nguồn tài nguyên object (tân ngữ, bổ ngữ): URI
 - Predicate (vị ngữ): URI hoặc literal

Ví dụ:



Hình 2: Ví dụ về một mẫu quan hệ bô ba trong RDF

Dưới đây là ví dụ về cú pháp RDF/XML biểu diễn quan hệ như bảng sau:

Title	Artist	Country	Company	Price	Year
Empire Burlesque	Bob Dylan	USA	Columbia	10.90	1985
Hide your heart	Bonnie Tyler	UK	CBS Records	9.90	1988
...					

```
1. <?xml version="1.0"?>
2. <rdf:RDF
3.   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
4.   xmlns:cd="http://www.recshop.fake/cd#">
5.   <rdf:Description
6.     rdf:about="http://www.recshop.fake/cd/Empire Burlesque">
7.       <cd:artist>Bob Dylan</cd:artist>
8.       <cd:country>USA</cd:country>
9.       <cd:company>Columbia</cd:company>
10.      <cd:price>10.90</cd:price>
11.      <cd:year>1985</cd:year>
12.    </rdf:Description>
13.    <rdf:Description
14.      rdf:about="http://www.recshop.fake/cd/Hide your heart">
15.        <cd:artist>Bonnie Tyler</cd:artist>
16.        <cd:country>UK</cd:country>
17.        <cd:company>CBS Records</cd:company>
18.        <cd:price>9.90</cd:price>
19.        <cd:year>1988</cd:year>
20.      </rdf:Description>
21.      .
22.      .
23.      .
24.    </rdf:RDF>
```

RDFS (RDF Schema) cũng là ngôn ngữ để mô tả ngữ nghĩa như RDF. RDFS biểu diễn các từ vựng RDF đơn giản trên Web (theo [5]). Các công nghệ định nghĩa từ vựng như OWL

hoặc SKOS, xây dựng trên RDFS và cung cấp ngôn ngữ để định nghĩa các ontology dựa trên Web.

RDFS có thể coi như là bản mở rộng ngôn ngữ nghĩa cho ngôn ngữ RDF. RDF cung cấp mô tả các nhóm nguồn dữ liệu và mối quan hệ giữa các nguồn dữ liệu này.

2. OWL(*Web Ontology Language*)

Theo W3C [6], OWL là một ngôn ngữ web ngữ nghĩa được thiết kế để thể hiện dữ liệu bao gồm định nghĩa, và các quan hệ liên quan. OWL là một ngôn ngữ web ngữ nghĩa có thể được khai thác bằng máy tính. OWL được thiết kế như một tiêu chuẩn mới của ngôn ngữ biểu diễn ontology trên web, nó được xây dựng dựa trên RDF/RDFS. OWL cung cấp các chuẩn để quản lý tài nguyên, để chia sẻ cũng như tái sử dụng dữ liệu ngữ nghĩa trên web.

2.1.5 Cây cấu trúc ngữ pháp

Phần này, em sẽ trình bày về cây cấu trúc trong tiếng anh, đặc biệt quan tâm đến gắn thẻ cho cây cấu trúc. Hay còn gọi là Part-of-speech tags trong Penn Treebank Tags.

Part-of-speech tags là gắn cho mỗi từ trong câu một thẻ duy nhất theo vai trò của nó trong câu. Trong ngữ pháp cơ bản tiếng anh, câu được tạo thành ở 8 thành phần cơ bản là: động từ (the verb -VB), danh từ (the noun -NN), đại từ (the pronoun - PR + DT), tính từ (the adjective -JJ), trạng từ (the adverb - RB), giới từ (the preposition -IN), Điều điện (the conjunction - CC) và xen kẽ (the interjection - UH).

Theo [13], tác giả đã phát triển chú thích cho câu đầy đủ với các thẻ cụ thể. Sau đây, em xin phép được tổng hợp một số thẻ hay sử dụng trong đồ án này.

Clause Level

S - simple declarative clause, i.e. one that is not introduced by a (possibly empty) subordinating conjunction or a *wh*-word and that does not exhibit subject-verb inversion.

SBAR - Clause introduced by a (possibly empty) subordinating conjunction.

SBARQ - Direct question introduced by a *wh*-word or a *wh*-phrase. Indirect questions and relative clauses should be bracketed as SBAR, not SBARQ.

SINV - Inverted declarative sentence, i.e. one in which the subject follows the tensed verb or modal.

SQ - Inverted yes/no question, or main clause of a *wh*-question, following the *wh*-phrase in SBARQ.

Phrase Level

ADJP - Adjective Phrase.

ADVP - Adverb Phrase.

CONJP - Conjunction Phrase.

FRAG - Fragment.

INTJ - Interjection. Corresponds approximately to the part-of-speech tag UH.

LST - List marker. Includes surrounding punctuation.

NAC - Not a Constituent; used to show the scope of certain prenominal modifiers within an NP.

NP - Noun Phrase.
NX - Used within certain complex NPs to mark the head of the NP. Corresponds very roughly to N-bar level but used quite differently.
PP - Prepositional Phrase.
PRN - Parenthetical.
PRT - Particle. Category for words that should be tagged RP.
QP - Quantifier Phrase (i.e. complex measure/amount phrase); used within NP.
RRC - Reduced Relative Clause.
UCP - Unlike Coordinated Phrase.
VP - Verb Phrase.
WHADJP - Wh-adjective Phrase. Adjectival phrase containing a wh-adverb, as in *how hot*.
WHAVP - Wh-adverb Phrase. Introduces a clause with an NP gap. May be null (containing the 0 complementizer) or lexical, containing a wh-adverb such as *how* or *why*.
WHNP - Wh-noun Phrase. Introduces a clause with an NP gap. May be null (containing the 0 complementizer) or lexical, containing some wh-word, e.g. *who*, *which book*, *whose daughter*, *none of which*, or *how many leopards*.
WHPP - Wh-prepositional Phrase. Prepositional phrase containing a wh-noun phrase (such as *of which* or *by whose authority*) that either introduces a PP gap or is contained by a WHNP.
X - Unknown, uncertain, or unbracketable. X is often used for bracketing typos and in bracketing *the...the*-constructions.

CC - Coordinating conjunction	
CD - Cardinal number	
DT - Determiner	
EX - Existential there	
FW - Foreign word	
IN - Preposition or subordinating conjunction	
JJ - Adjective	RBS - Adverb, superlative
JJR - Adjective, comparative	RP - Particle
JJS - Adjective, superlative	SYM - Symbol
LS - List item marker	TO - to
MD - Modal	UH - Interjection
NN - Noun, singular or mass	VB - Verb, base form
NNS - Noun, plural	VBD - Verb, past tense
NNP - Proper noun, singular	VBG - Verb, gerund or present participle
NNPS - Proper noun, plural	VBN - Verb, past participle
PDT - Predeterminer	VBP - Verb, non-3rd person singular present
POS - Possessive ending	VBZ - Verb, 3rd person singular present
PRP - Personal pronoun	WDT - Wh-determiner
PRPS - Possessive pronoun (prolog version PRP-S)	WP - Wh-pronoun
RB - Adverb	WPS - Possessive wh-pronoun (prolog version WP-S)
RBR - Adverb, comparative	WRB - Wh-adverb

2.1.6 Ngôn ngữ SPARQL

SPARQL là một ngôn ngữ truy vấn ngữ nghĩa bằng RDF. SPARQL là một chuẩn để truy cập dữ liệu RDF của World Wide Web Consortium, và được coi là một trong những công nghệ chủ chốt của web semantic. Ngày 15 tháng 2008, SPARQL đã trở thành một ngôn ngữ truy vấn khuyễn dùng của W3C. SPARQL cho phép cho một truy vấn bao gồm mô hình ngữ nghĩa triple, mẫu không cố định[7].

SPARQL cho phép người dùng viết các truy vấn rõ ràng. Ví dụ sau trả về các tiêu đề của cuốn sách lưu trong dataset.

```

1. SELECT ?title
2. WHERE
3. {
4.   <http://example.org/book/book1> <http://purl.org/dc/elements/1.1/title> ?title
5. }
  
```

Ngôn ngữ SPARQL gồm bốn loại truy vấn khác nhau cho các mục đích khác nhau:

- Truy vấn SELECT: Sử dụng để lấy các giá trị từ SPARQL endpoint. Các kết quả được trả về trong một định dạng bảng.

- Truy vấn CONSTRUCT: Sử dụng để trích xuất thông tin từ SPARQL Endpoint. Các kết quả thành dạng RDF hợp lệ.
- Truy vấn ASK: Là truy vấn cho các câu hỏi đúng/sai. Cung cấp cho các truy vấn trên SPARQL endpoint
- Truy vấn DESCRIBE: Sử dụng để trích xuất một đồ thị RDF từ SPARQL endpoint.

2.1.7 Ngôn ngữ lập trình JAPE

JAPE là từ viết tắt của Java Annotation Patterns Engine. Đặc điểm quan trọng của JAPE là JAPE cung cấp tái nạp trạng thái ưu tiên trên các chú thích dựa trên các biểu thức lập trình. JAPE cung cấp các biểu thức phục vụ cho viễn nhận dạng ngữ nghĩa.

Về cơ bản, JAPE khá giống với ngôn ngữ JAVA, điểm khác là nó bao gồm các biểu thức chính quy và quy phạm mới để biểu diễn ngữ nghĩa. Điểm đặc trưng cho JAPE là các luật mô tả. Luật là một tập hợp gồm nhiều thành phần (phase), mỗi phần chứa tập các luật được định nghĩa trong từng pattern. JAPE Grammar gồm có hai thành phần là về trái (LHS) và về phải (RHS). Hai về được phân biệt bởi ký hiệu \rightarrow . Có cú pháp như sau : LHS \rightarrow RHS.

2.2 Các công nghệ và platform sử dụng

Ngoài các kiến thức sử dụng ở trên, trong phạm vi đồ án, có sử dụng các công nghệ như KIM Platform, Allegrograph và Ruby and Rails. Em xin được mô tả lần lượt các công nghệ này một cách sơ lược nhất.

2.2.1 Giới thiệu về KIM Platform

KIM (Knowledge and Information Management)[8] là một hệ thống rút trích thông tin được tăng cường ngữ nghĩa, cung cấp chú thích ngữ nghĩa tự động bằng các tham khảo đến các lớp trong ontology và đến các thể hiện. Hệ thống này đã được chạy trên một tập hợp tài liệu phát triển liên tục.

Nền tảng KIM cung cấp cơ sở hạ tầng thông tin và tri thức mới, cùng với các dịch vụ để chú thích ngữ nghĩa tự động, lập chỉ mục, và tìm kiếm tài liệu. Nó cung cấp một cơ sở hạ tầng để rút trích thông tin tùy biến và mở rộng cũng như chú thích và quản lý tài liệu

dựa trên GATE. Để cung cấp mức độ cơ bản về hiệu suất và cho phép dễ dàng khởi động các ứng dụng, KIM được trang bị với một ontology mức trung và một cơ sở tri thức về các thực thể chung quan trọng. Các ontology và cơ sở tri thức liên quan được xử lý dùng công nghệ Semantic Web với các chuẩn như kho chứa RDFS, trung gian ontology và lý luận. Theo quan điểm kỹ thuật, nền tảng này cho phép các ứng dụng dựa trên KIM sử dụng nó để chú thích ngữ nghĩa tự động, tìm kiếm nội dung dựa trên các hạn chế ngữ nghĩa, truy vấn và sửa đổi các ontology cơ bản và cơ sở tri thức.

Nền tảng KIM bao gồm KIM Ontology (KIMO), cơ sở tri thức, KIM server (với API để truy cập từ xa, nhúng, và tích hợp), và các lối vào (plug-in trình duyệt cho Internet Explorer, giao diện người dùng web KIM với nhiều phương pháp truy cập khác nhau, và bộ thăm dò tri thức để điều hướng cơ sở tri thức). API KIM cung cấp chú thích ngữ nghĩa, các dịch vụ lập chỉ mục và tìm kiếm, và cơ sở hạ tầng. Các ontology KIM và các cơ sở tri thức được giữ trong các kho chứa ngữ nghĩa dựa trên công nghệ Web có ngữ nghĩa và các chuẩn gồm kho chứa RDF(RDFS), và ontology. KIM cung cấp một cơ sở hạ tầng trưởng thành cho rút trích thông tin có thể tùy biến và mở rộng, cũng như chú thích và quản lý tài liệu, dựa trên GATE.

2.2.2 Allegrograph

Theo [9], AllegroGraph là một mã nguồn đóng, được thiết kế để lưu trữ dạng dữ liệu triple RDF, một định dạng chuẩn cho liên kết dữ liệu. Nó cũng hoạt động như một kho lưu trữ tài liệu được thiết kế để lưu trữ, truy xuất và quản lý thông tin cho dữ liệu ngữ nghĩa.

AllegroGraph được phát triển để đáp ứng các tiêu chuẩn W3C cho RDF, do đó, nó được coi là một Cơ sở dữ liệu RDF. Đây là một triển khai tham chiếu cho giao thức SPARQL.

Các chức năng cơ bản của AllegroGraph bao gồm:

- Semantic Engine: AllegroGraph cung cấp công cụ tìm kiếm ngữ nghĩa bằng ngôn ngữ SPARQL, truy vấn các dữ liệu ngữ nghĩa trong thời gian thực.
- Store RDF triples: Như đã mô tả ở trên, AllegroGraph được coi như một kho lưu trữ dữ liệu RDF có thể truy vấn theo thời gian thực.

Hiện tại AllegroGraph có interfaces cho Java, Python, Ruby, Perl, C#, Clojure, and Common Lisp

2.2.3 Ruby on Rails Framework

Ruby on Rails, hay Rails, là một web framework được viết bằng Ruby. Về cơ bản, Ruby On rails là một Framework cho phép phát triển ứng dụng Web gồm 2 phần cơ bản:

- Phần ngôn ngữ Ruby[11]
- Phần Framework Rails bao gồm nhiều thư viện liên kết.

Rails cũng là framework tuân theo mô hình MVC[12] và bao gồm các công cụ cho phép làm nhiều chức năng thông dụng cho web, như thêm, xóa sửa thông qua scaffold.

Ruby on Rails (RoR) là một web framework được viết bằng ngôn ngữ Ruby và tất cả các ứng dụng trong Rails sẽ được viết bằng Ruby. Ruby on Rails được tạo ra để hỗ trợ các lập trình viên việc phát triển các phần mềm nền web một cách nhanh nhất có thể.

Rails framework tận dụng các đặc điểm của ngôn ngữ Ruby. Yukishiro Matsumoto viết ra ngôn ngữ này vào năm 1995, nó khá giống với các ngôn ngữ thông dịch khác như Perl, Eiffel, Python.... Ruby là ngôn ngữ script, định nghĩa kiểu động và là một ngôn ngữ hướng đối tượng, nó được thiết kế với một cú pháp trong sáng, tạo cảm giác dễ đọc, và viết code ngắn gọn nhất có thể đối với người dùng, ví dụ như nó không cần dấu chấm phẩy khi kết thúc câu lệnh, không cần các dấu ngoặc đơn khi khai báo các phương thức, có những đoạn code thậm chí được viết giống như việc chúng ta viết tiếng anh vậy.

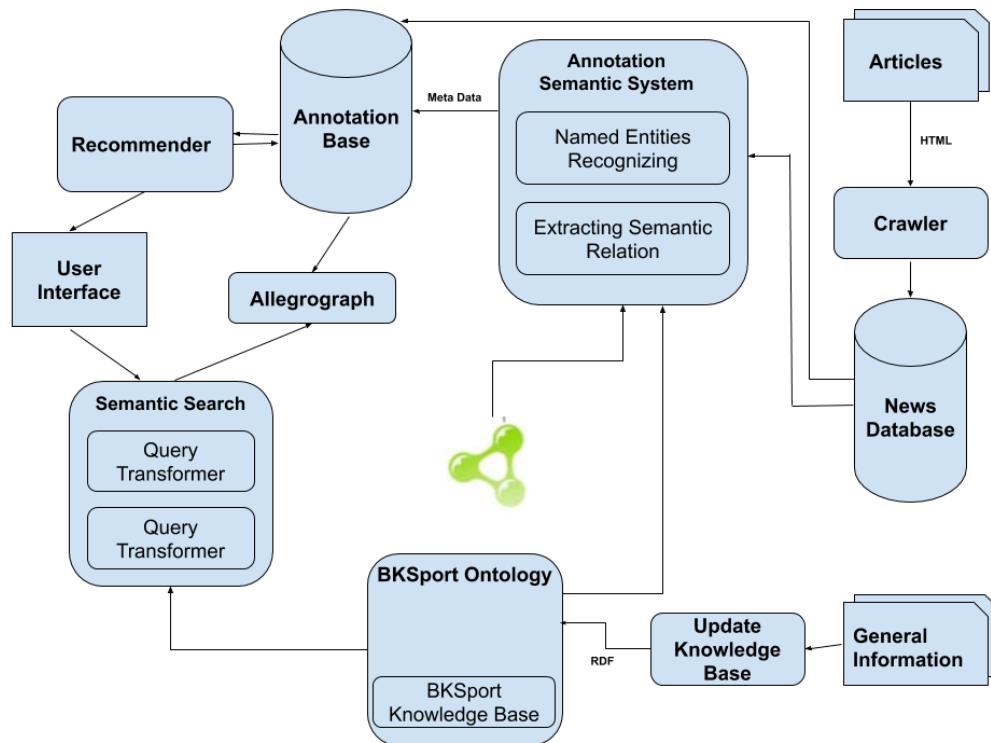
Ruby on Rails được tạo ra như là một câu trả lời đối với các web frameworks khác như J2EE, .NET. Để quá trình phát triển phần mềm diễn ra nhanh hơn, RoR sử dụng các qui ước triệt để và đảm nhận xử lý rất nhiều các task khiến người lập trình viên không phải bận tâm về nó nữa như : mail management, object- database mappers, file structures, code generation..., đây chính là hai đặc điểm nổi bật nhất của RoR, điều này không chỉ giúp các lập trình viên viết code ít hơn, phát triển ứng dụng nhanh hơn mà còn làm ứng dụng dễ hiểu và dễ bảo trì hơn.

Chương 3 Hoàn thiện hệ thống tổng hợp tin tức thể thao BKSport với khả năng xử lý ngữ nghĩa

Trong chương này, em xin được trình bày các hoàn thiện cho hệ thống tổng hợp tin tức thể thao với công nghệ ngữ nghĩa. Đặc biệt trong các Module Semantic Anotation, Module Semantic Search và Module Webstie FontEnd.

3.1 Tổng quan về hệ thống tích hợp tin tức BKSport

Hệ thống tích hợp tin tức thể thao BKSport là một hệ thống đã được xây dựng từ lâu do nhóm Semantic Innovation Group phát triển, bao gồm nhiều chức năng và thành phần phức tạp do đó trước khi đi vào chi tiết kiến trúc của các module riêng lẻ trong hệ thống BKSport ta cần xem xét vị trí của nó trong một hệ thống. Đây là mô hình kiến trúc hệ thống BKSport đã được sử dụng.



Hình 3: Sơ đồ tổng quan kiến trúc hệ thống BKSport

Từ sơ đồ tổng quan trên, luồng hoạt động của hệ thống được mô tả như sau:

BKSport Ontology được coi là Cơ sở tri thức của hệ thống bao gồm: Các nhóm từ vựng cơ bản để mô hình hóa tri thức thể thao và các thực thể cụ thể trong lĩnh vực thể thao, từ đây các module **Semantic Annotation System** và **Semantic Search** lấy được thông tin trích chọn.

Module Ontology Crawler thu thập thông tin từ các website tổng hợp thông tin về các lĩnh vực liên quan. Dữ liệu Crawler sau khi được chuẩn hóa sẽ lưu về dưới dạng RDF và được lưu vào **BKSport Ontology**. Đây được coi như bước làm giàu cơ sở tri thức cho hệ thống.

Module Crawler thu thập tin tức từ các website tin tức. Từ đây Module giữ lại các thành phần quan trọng như tiêu đề, nội dung bài báo. Sau khi tách dữ liệu sẽ lưu vào cơ sở dữ liệu (**News Database Module**).

Với dữ liệu trong cơ sở dữ liệu (**News Database Module**), **Module Semantic Annotation System** sẽ dựa vào **Module BKSport Ontology** để nhận dạng và trích xuất thông tin ngữ nghĩa, chuyển về dạng MetaData và lưu vào **Annotation Base** và lưu lại một phần ở **News Database Module** để phục vụ việc hiện thị cho người dùng. Kiến trúc cụ thể của module này sẽ được mô tả kỹ ở phần sau.

Khi người dùng xem một tin tức trong hệ thống, **Module Recommender** sẽ dựa vào các metadata trong **Annotation Base** để phân tích và đưa ra các tin liên quan nhắc mục đích giới thiệu cho người dùng.

Module Semantic Search cung cấp dịch vụ trên giao diện Website. Người dùng có thể đọc tin và tìm kiếm ngữ nghĩa bằng ngôn ngữ tự nhiên. Tại đây, câu hỏi dưới dạng ngôn ngữ tự nhiên được module tiếp nhận, từ đây với sự trợ giúp của **Module BKSport Ontology**, module sẽ chuyển câu hỏi về dưới dạng truy vấn SPARQL. Module tiếp tục truy vấn vào **Annotation Base** và **Database News** chứa các tin tức và các thông tin liên quan về mặt ngữ nghĩa và hiển thị lại cho người dùng. Kiến trúc cụ thể của module này em xin được mô tả kỹ ở phần tiếp theo.

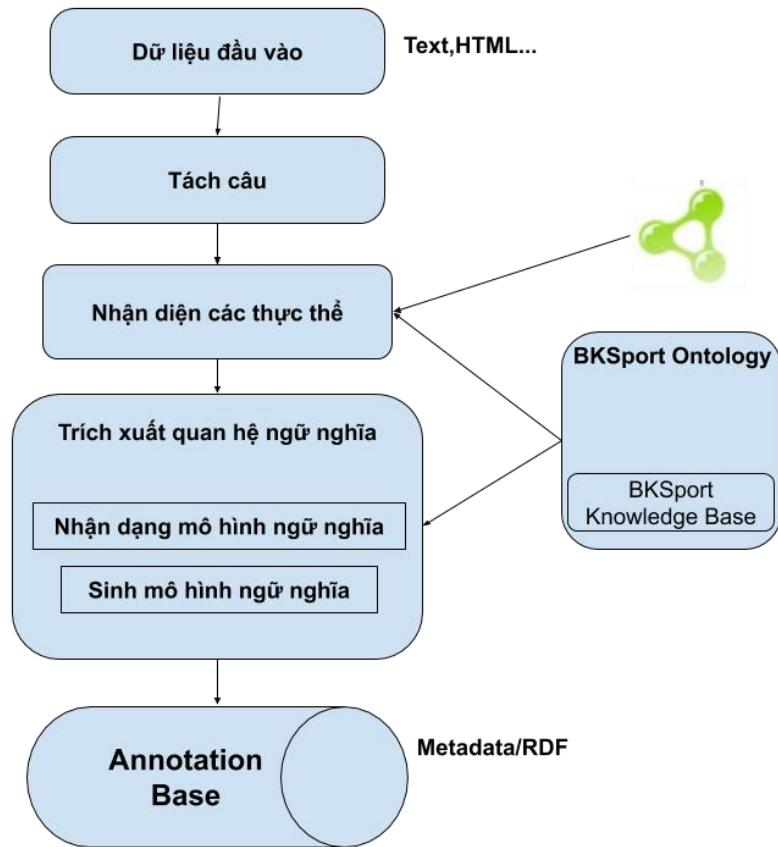
Trong phạm vi đồ án này, các nghiên cứu và đóng góp của em thực hiện trên **Module Semantic Search** và **Module Semantic Annotation** và **Module Webstie FontEnd**.

3.2 Chức năng tạo dữ liệu ngữ nghĩa trong hệ thống tin tức BKSport

Về cơ bản, **Module Semantic Annotation** là module quan trọng nhất của hệ thống. Module sẽ tóm tắt và tổng hợp ngữ nghĩa từ văn bản dạng text/html sang dạng dữ liệu metadata. Mô tả và cấu trúc dữ liệu để máy có thể hiểu được.

3.2.1 Kiến trúc Module Semantic Annotation đã đề xuất trước đây

Dưới đây là kiến trúc **Module Semantic Annotation** đã được mô hình hóa từ trước bởi nhóm nghiên cứu.



Hình 4: Kiến trúc Module Semantic Annotation

- **Bước 1: Xử lý đầu vào và tách câu**

Ở bước này, module sẽ nhận đầu vào là các tin tức dưới dạng text hoặc html.

Module sẽ kiểm tra và đưa ra nếu đầu vào không ở dưới dạng chuẩn này. Sau khi đầu vào được chuẩn hóa, module sẽ tiến hành tách câu.

Việc tách câu cũng vô cùng quan trọng. Ngoài việc phân vùng để tiến hành thực hiện các bước tiếp theo, tại đây các thành phần được phân chia. Đặc biệt là tiêu đề, bởi trong bài báo, đôi khi tiêu đề đã mang hàm ý tóm tắt nội dung, do đó hệ thống đánh giá tiêu đề có trọng số cao hơn để xác định ngữ nghĩa.

Tóm lại module này thực hiện 2 nhiệm vụ:

- Tách câu
 - Phân vùng: Vùng nội dung văn bản – Vùng tiêu đề bài báo
- **Bước 2 : Nhận diện các thực thể**

Để nhận diện được ngữ nghĩa của văn bản, trước hết hệ thống phải hiểu được ngữ nghĩa của các thực thể có tên xuất hiện trong văn bản. Các thực thể được đặt tên trong lĩnh vực thể thao bao gồm tên của các cầu thủ, huấn luyện viên, câu lạc bộ, sân vận động, sự kiện thể thao...

Ví dụ: “Cordoba has completed the loan signing of Brazilian Winger Ryder Matos”

Hệ thống lúc này cần phải hiểu **Cordoba** là tên của một câu lạc bộ bóng đá và **Ryder Matos** là một cầu thủ chạy cánh (a winger). Để làm được điều này, phải có một bước nhận diện thực thể có tên.

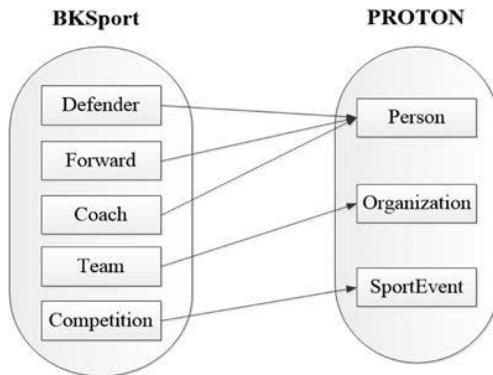
Để thực hiện được điều này, hệ thống đã tái sử dụng **KIM platform** để xác định thực thể có tên. **KIM platform** được xây dựng để nhận diện trong một phạm vi rất rộng, nó không dành cho một lĩnh vực cụ thể nào. Vậy nên để phục vụ cho lĩnh vực thể thao, hệ thống cần thêm một tập hợp các khái niệm và thuộc tính chi tiết hơn bổ sung vào cơ sở tri thức của KIM.

Ví dụ: Ở **KIM platform** thực thể được đặt tên được thể hiện ở mức độ chung chung (ví dụ như Người –Person) mà không chi tiết (ví dụ: cầu thủ chạy cánh- winger, tiền đạo- forward, v.v.).

Do đó, hệ thống đã tích hợp cơ sở tri thức BKSport với PROTON, theo nghĩa là các khái niệm chuyên biệt hơn trong cái trước sẽ thay thế khái niệm trừu tượng trong cái sau trong

quá trình nhận dạng. Việc tích hợp có thể được thực hiện bằng cách ánh xạ các khái niệm giữa chúng.

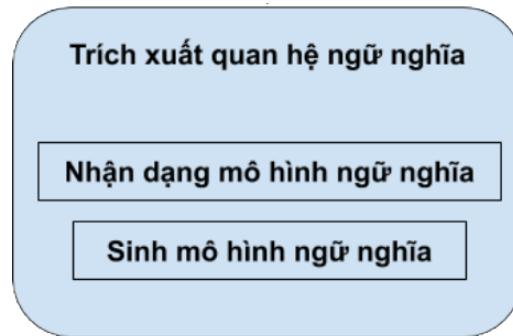
Ví dụ, các lớp của cơ sở tri thức BKSport ánh xạ với các khái niệm nguyên thủy trong KIM Platform.



Để thực hiện được điều này, **BKSport Ontology** đóng vai trò rất lớn, là cơ sở tri thức chứa dữ liệu tổng hợp các dữ liệu về các thực thể có tên. Do đó việc tổng hợp, phân loại và xử lý dữ liệu của cơ sở tri thức là vô cùng quan trọng và cần được thực hiện thường xuyên.

- *Bước 3: Trích xuất quan hệ ngữ nghĩa*

Với đầu ra của bước 2 là các thực thể được đánh dấu, thì đầu vào ở bước này bao gồm các thực thể có tên, đồng thời với tập các câu được tách ra từ bước 1. Giờ đây hệ thống tiến hành trích xuất các quan hệ ngữ nghĩa. Đây cũng là thành phần quan trọng nhất của hệ thống này.



Bước đầu tiên của quy trình này, là **nhận dạng các mô hình ngữ nghĩa**.

Hiện tại, hệ thống đã mô hình hóa 3 nhóm chính là:

- Mọi quan hệ giữa người và người:

<Person><relationship><Person>

Ví dụ: <Wayne Rooney> <be against> <Alex Ferguson>

- Mọi quan hệ giữa người và tổ chức nào đó:

<Person> <relation> <Organization>

Ví dụ: <Wayne Rooney> <transfer to> <Manchester United>

- Mọi quan hệ giữa tổ chức và tổ chức:

<Organization><relation> <Organization>

Ví dụ: <Manchester United><defeat><Chelsea FC>

Để nhận dạng các mô hình này, hệ thống đã mô hình ngữ nghĩa bằng các luật nhận dạng, viết bằng ngôn ngữ JAPE. Mỗi mối quan hệ được đặc trưng bởi một tập hợp các từ cụ thể khi xuất hiện trong văn bản. Nhóm từ khóa này làm cơ sở để xây dựng các mẫu nhận dạng mối quan hệ.

Ví dụ:

Relationship	Keyword	Pattern
defeat	“win”, “beat”, “defeat”, “overpower”	{SportTeam} [key] {SportTeam}
faceWith	“face with”, “against”	{SportPerson}[key] {SportPerson}
leftOnBenchFor	“left on bench for”	{SportPerson}[key] {SportCompetition}
hasRetired	“retires”, “retired”	{SportPerson} [key]

Bảng 1: Các từ khóa xuất hiện phổ biến trong các mối quan hệ thể thao

Từ đây các luật được hình thành bằng cách quy định chặt chẽ về ngữ nghĩa và khoảng cách ngữ nghĩa. Mỗi mối quan hệ được xác định sẽ được ánh xạ tới một mối quan hệ tương ứng trong **BKSport Ontology** đã được xác định ở bước 2.

Từ đây ta thấy rằng việc bổ sung các từ khóa cho hệ thống cũng là một việc quan trọng. Nó sẽ giúp hệ thống nhận diện tốt hơn và chính xác hơn các quan hệ cần thiết.

Bước tiếp theo là **Mô hình hóa các quan hệ ngữ nghĩa**.

Từ việc mô hình hóa, sinh luật, nhận dạng và bắt các quan hệ ngữ nghĩa, bước này sẽ mô hình hóa các quan hệ ngữ nghĩa đã được tìm ra ở bước nhận dạng mô hình ngữ nghĩa ở trên.

Hệ thống sinh ra các bộ 3 để mô tả các quan hệ ngữ nghĩa dưới ngôn ngữ RDF.

Đặc trưng bởi bộ 3 : <Subject> <Predicate> <Object>

Ví dụ về một bộ 3 được sinh ra:

```
1. <rdf:Description rdf:about="http://bk.sport.owl#stoke-city">
2.   <j:0:lose rdf:resource="http://bk.sport.owl#crystal-palace"/>
3. </rdf:Description>
```

3.2.2 Phát triển tính năng sinh ngữ nghĩa cho Module Semantic Annotation

Để phát triển tính năng sinh chú thích ngữ nghĩa cho hệ thống, em đề xuất các phương pháp sau:

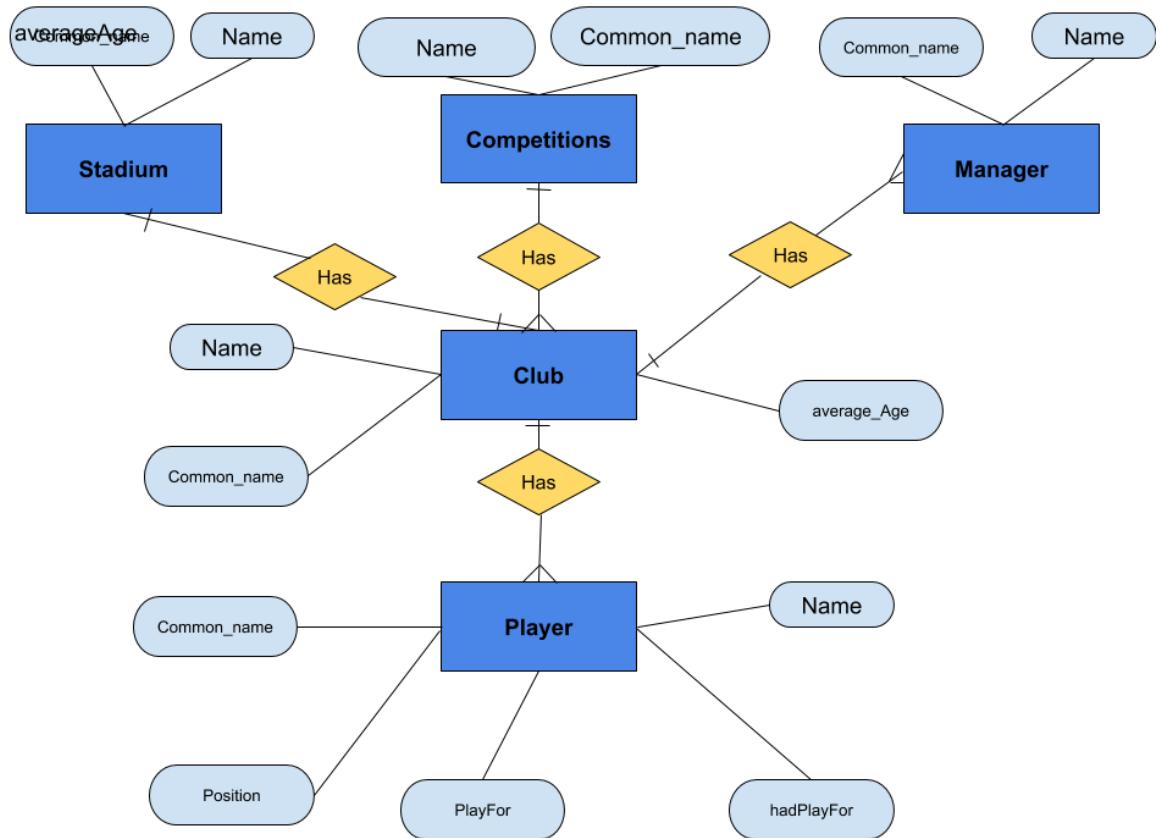
- Làm giàu cơ sở tri thức
- Bổ sung các lớp quan hệ hệ thống chưa nắm bắt

Trong phần này, em xin được trình bày sơ lược về thiết kế dữ liệu, thiết kế chương trình cùng một vài method chính ảnh hưởng đến việc sinh thêm chú thích ngữ nghĩa. Các logic về thuật toán, các thức thực hiện xem xin trình bày ở chương kế tiếp.

1. Thiết kế cấu trúc database làm giàu cơ sở tri thức

Để làm giàu cơ sở tri thức, em tiến hành thu thập thêm dữ liệu các thực thể có trong lĩnh vực bóng đá. Ở đây em tập trung vào các lớp : Cầu thủ bóng đá, Huấn Luyện Viên, Câu lạc bộ, Sân vận động, Giải đấu tham gia.

Dưới đây là sơ đồ cấu trúc database chứa thông tin được lưu trữ:



Hình 5: Mô hình quan hệ - thực thể(E-R) trong thiết kế CSDL module Crawler

Từ biểu đồ thực thể liên kết trên, em thiết kế CSDL và ánh xạ sang thuộc tính ngữ nghĩa như sau:

Thuộc tính trên database	Thuộc tính ngữ nghĩa
Mặc định	Type: Player
Name	Label, mainLabel
Common_name	hasAlias
HasPlayFor	hasPlayed
PlayFor	PlayFor
Position	Type: Striker , Left-Back...

Bảng 2: Ánh xạ quan hệ từ bảng Player trong Database sang thuộc tính ngữ nghĩa của Player trong cơ sở tri thức

Thuộc tính trên database	Thuộc tính ngữ nghĩa
Mặc định	Type: Coach
Name	Label, mainLabel
Common_name	hasAlias

Bảng 3: Ánh xạ quan hệ từ bảng Manager trong Database sang thuộc tính ngữ nghĩa của Manager trong cơ sở tri thức

Thuộc tính trên database	Thuộc tính ngữ nghĩa
Mặc định	Type: Player
Name	Label, mainLabel
Common_name	hasAlias
AverageAge	AverageAge

Bảng 4: Ánh xạ quan hệ từ bảng Club trong Database sang thuộc tính ngữ nghĩa của Club trong cơ sở tri thức

Thuộc tính trên database	Thuộc tính ngữ nghĩa
Mặc định	Type: Player
Name	Label, mainLabel
Common_name	hasAlias
location	location

Bảng 5: Ánh xạ quan hệ từ bảng Stadium trong Database sang thuộc tính ngữ nghĩa của Stadium trong cơ sở tri thức

Thuộc tính trên database	Thuộc tính ngữ nghĩa
Mặc định	Type: Player
Name	Label, mainLabel
Common_name	hasAlias
location	location

Bảng 6: Ánh xạ quan hệ từ bảng Stadium trong Database sang thuộc tính ngữ nghĩa của Stadium trong cơ sở tri thức

Thuộc tính trên database	Thuộc tính ngữ nghĩa
Player - Club	<Player><PlayFor><Club>
Stadium - Club	<Stadium><homeOf><Club>
Manager - Club	<Manager><managerOf><Club>
Club - Competition	<Club><PlayIn><Competition>

Bảng 7: Ánh xạ quan hệ từ các thuộc tính trong database với thuộc tính ngữ nghĩa

Ví dụ 1 thực thể Player sau khi sinh chú thích ngữ nghĩa:

```

1. <owl:NamedIndividual rdf:about="http://bk.sport.owl#lionel-messi">
2.   <rdfs:label xml:lang="en">lionel messi</rdfs:label>
3.   <protons:mainLabel>lionel messi</protons:mainLabel>
4.   <rdf:type rdf:resource="#bksport;Right-Winger"/>
5.   <protons:generatedBy rdf:resource="http://bk.sport.owl"/>
6.   <playFor rdf:resource="#bksport;fc-barcelona"/>
7.   <hasPlayed rdf:resource="#bksport;fc-barcelona"/>
8.   <hasPlayed rdf:resource="#bksport;fc-barcelona-b"/>

```

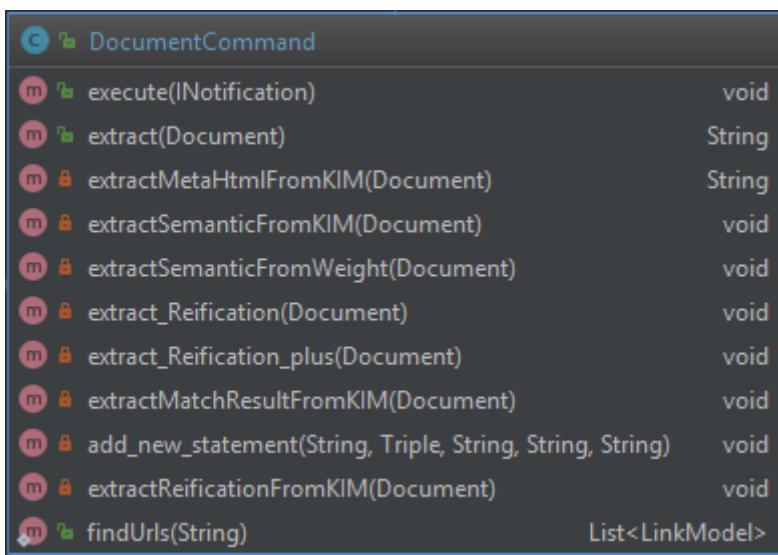
```

9.   <hasPlayed rdf:resource="#bksport;fc-barcelona-c"/>
10.  <hasPlayed rdf:resource="#bksport;fc-barcelona-u19"/>
11.  <hasPlayed rdf:resource="#bksport;fc-barcelona-u17"/>
12.  <hasPlayed rdf:resource="#bksport;fc-barcelona-youth"/>
13.  <hasPlayed rdf:resource="#bksport;club-atlético-newell's-old-boys-u19"/>
14.  </owl:NamedIndividual>

```

2. Thiết kế thêm lớp trong Module Semantic Annotation

Để nhận diện thêm các quan hệ trong chú thích ngữ nghĩa, em có viết thêm các lớp mới là lớp DocumentCommand – các thuộc tính và phương thức của lớp này như sau:

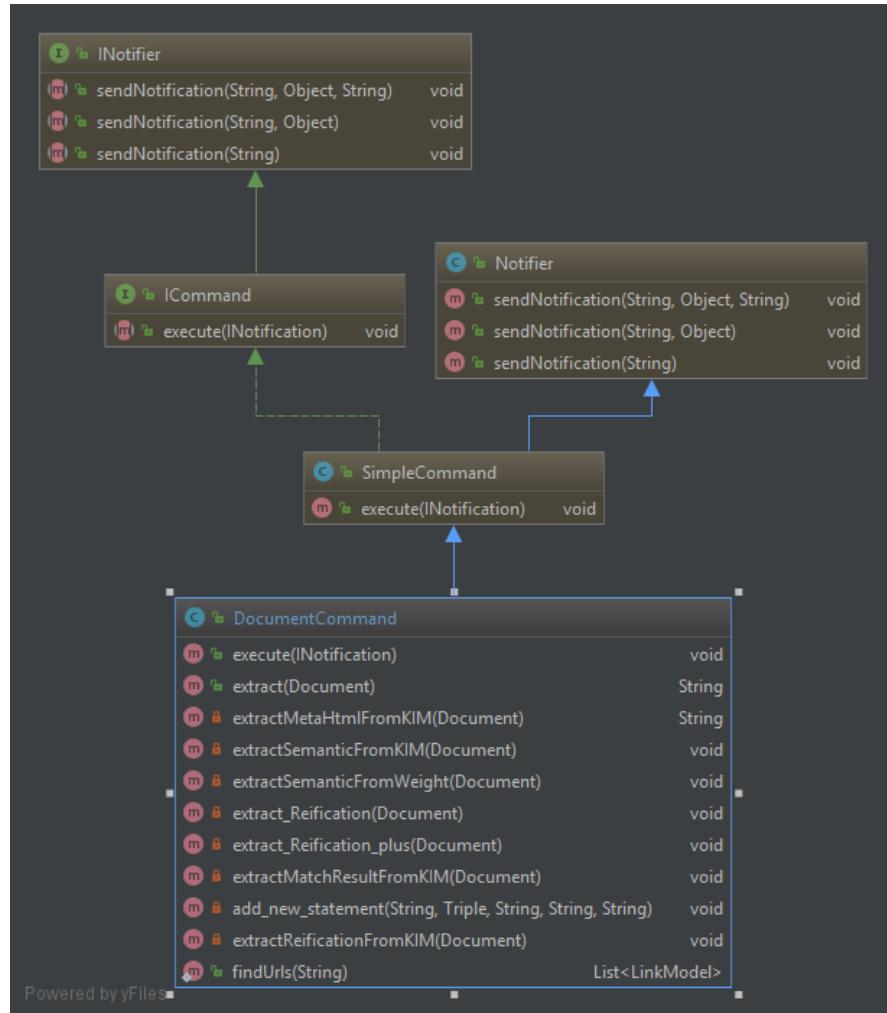


Các phương thức bao gồm:

- execute(Notification): Đầu vào là một thông báo chạy từ hệ thống, khi có lời gọi hàm này sẽ được thực hiện
- extract(Document): Đầu vào là một tài liệu, đầu ra là chuỗi String định dạng RDF – Chú thích ngữ nghĩa. Hàm này được hàm execute() gọi. Hàm này sẽ tiếp tục gọi các hàm sau:
 - extractMetaHTMLFromKIM(Document) : Chạy chú thích ngữ nghĩa 1 văn bản – đầu ra sẽ cho html đoạn văn bản được đánh dấu
 - extractSemanticFromWeight(Document): Chạy chú thích ngữ nghĩa 1 văn bản – đầu ra là các thực thể được đánh trọng số
 - extractSemanticFromKIM(Document) : Chạy chú thích ngữ nghĩa 1 văn bản – đầu ra cho các quan hệ thông thường

- `extract_Refification(Document)` : Chạy chú thích ngữ nghĩa 1 văn bản – đầu ra cho ngữ nghĩa của câu giàn tiếp, trích dẫn câu giàn tiếp.
- `extractMatchResultFromKIM(Document)` : chạy chú thích ngữ nghĩa 1 văn bản – đầu ra là ngữ nghĩa kết quả trích dẫn, hoặc suy dẫn từ trận đấu.

Đây là sơ đồ lời gọi method giữa các lớp:



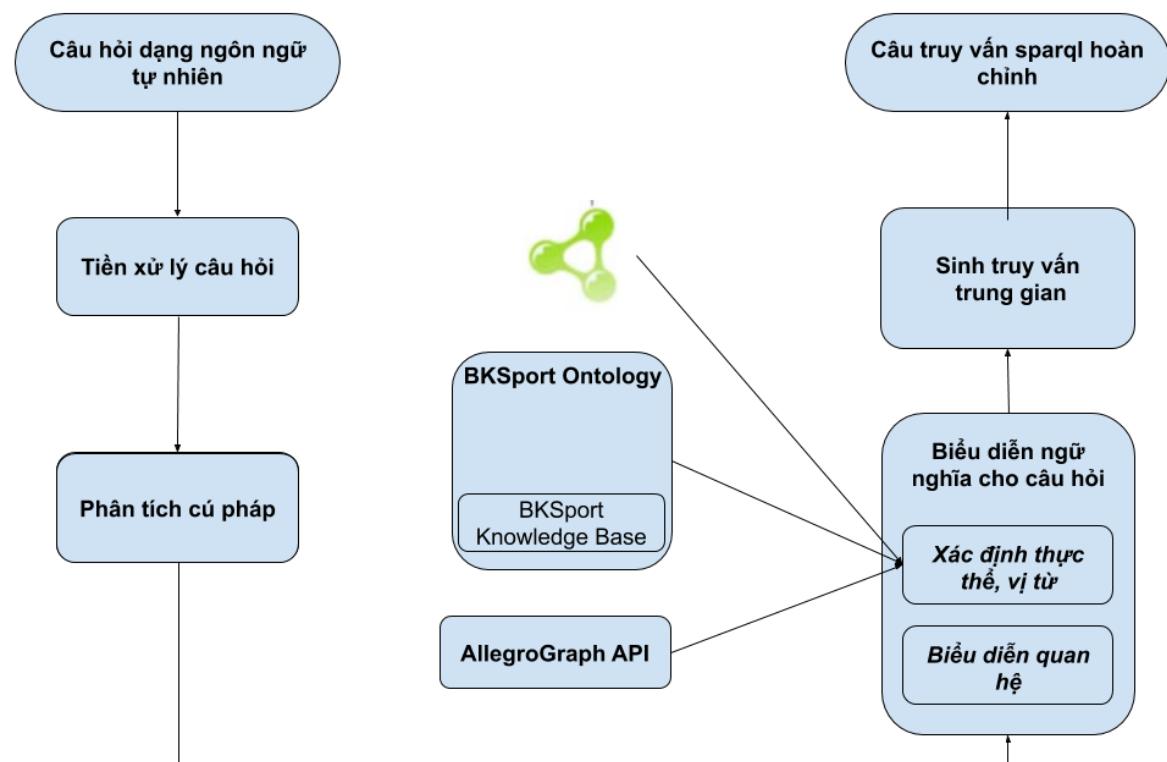
Khi có yêu cầu thực hiện chú thích ngữ nghĩa, hệ thống gửi thông báo cho `INotifier`, `INotifier` gọi tới `SimpleCommand`. `SimpleCommand` gọi lớp `DocumentCommand` để thực hiện. Thực hiện xong lớp `DocumentCommand` gửi kết quả và thông báo tới `Notifier`.

3.3 Chức năng chuyển đổi câu hỏi từ ngôn ngữ tự nhiên sang truy vấn SPARQL

Cũng như module Semantic Annotation, Module Semantic Search đóng vai trò quan trọng trong việc truy vấn tới các dữ liệu ngữ nghĩa. Sau đây em xin được trình bày sơ lược về kiến trúc mà hệ thống đã đề xuất và sau đó là đóng góp của em nhằm mục đích nhận diện nhiều dạng câu hỏi hơn trong module này.

3.3.1 Kiến trúc module Semantic Search đã đề xuất trước đây

Phương pháp chuyển đổi câu hỏi từ ngôn ngữ tự nhiên sang truy vấn SPARQL nằm trong **Module Semantic Search**. Quá trình thực hiện qua những bước được thể hiện qua sơ đồ sau:



Hình 6: Sơ đồ kiến trúc module Semantic Search

Do đây là kết quả của nghiên cứu đã được thực hiện trước đây và đồ án này tiếp tục kế thừa nên chỉ trình bày các bước cơ bản của chuyên đổi từ câu hỏi bằng ngôn ngữ tự nhiên sang truy vấn ngữ nghĩa SPARQL.

- **Bước 1: Tiền xử lý câu hỏi**

Module tiền xử lý câu hỏi có nhiệm vụ chuẩn hóa câu hỏi đầu vào ở dạng ngôn ngữ tự nhiên với các chức năng:

- Chuẩn hóa những từ ngữ không chuẩn về mặt cú pháp.
- Chuẩn hóa tên các thực thể chưa chính xác
- Xác định nhãn thời gian và thay thế bằng các giá trị cụ thể
- Chuyển đổi tương đương giữa các truy vấn để cho các phép xử lý các câu hỏi dạng rút gọn.

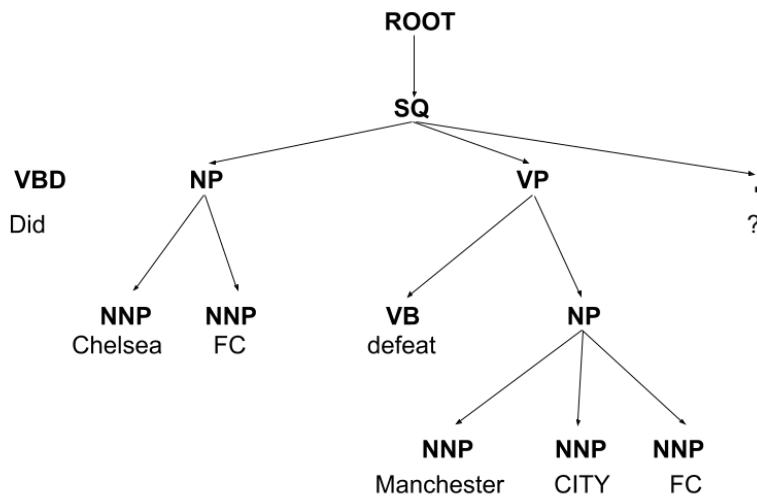
- **Bước 2: Phân tích cú pháp**

Nhiệm vụ của bước này là phân tích cấu trúc câu, các thành phần phụ thuộc trong câu hỏi người dùng. Module sử dụng phân tích cú pháp POSTagged là một phần của thư viện Stanford Parser để thực hiện công đoạn này. Đồng thời sử dụng các như bước 2 ở module Semantic Annotation để nhận dạng các thực thể có trong câu.

Kết quả của bước này cho ra sẽ xác định được:

- Cây cấu trúc câu: Biểu diễn trực quan đầu ra của quá trình phân tích cú pháp bao gồm:
 - ❖ Thứ tự tuyến tính của các từ trong câu
 - ❖ Các nhóm từ đi với nhau tạo thành cụm từ
 - ❖ Cấu trúc phân cấp của các cụm từ
- Các thành phần phụ thuộc trong câu:
Biểu diễn quan hệ ngữ pháp giữa các từ trong một câu.
Mỗi một thành phần phụ thuộc được biểu diễn theo dạng:
<<Tên quan hệ>> (Thực thể, Phụ thuộc)

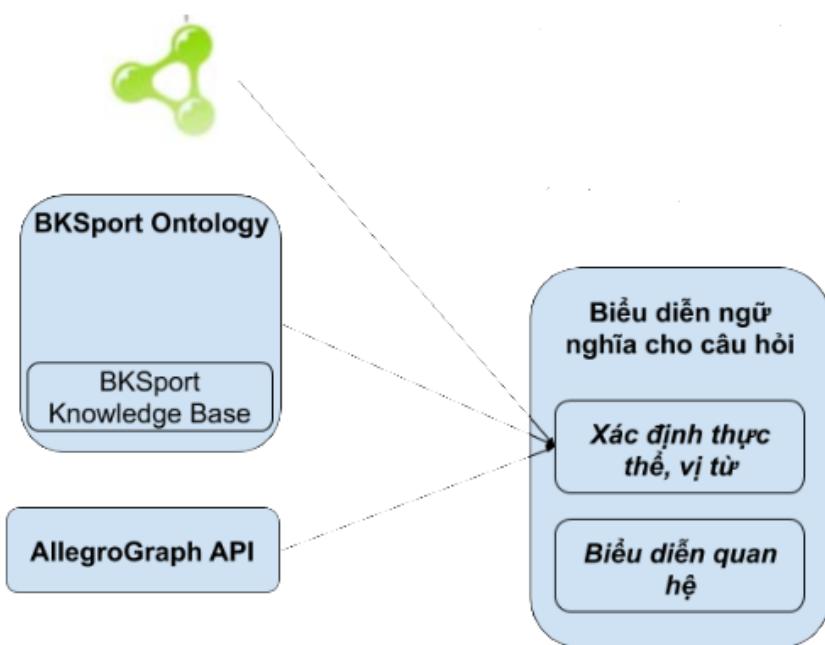
Ví dụ về kết quả phân tích một câu hỏi như sau:



Các ý nghĩa của các thẻ như VBD, NP, NNP đã được trình bày ở mục cơ sở lý thuyết.

- **Bước 3: Biểu diễn ngữ nghĩa cho câu hỏi**

Từ cấu trúc câu đã nhận dạng, cùng các bộ quan hệ đã nhận dạng được từ bước 2, ta sẽ sinh ra các bộ 3 quan hệ phù hợp.



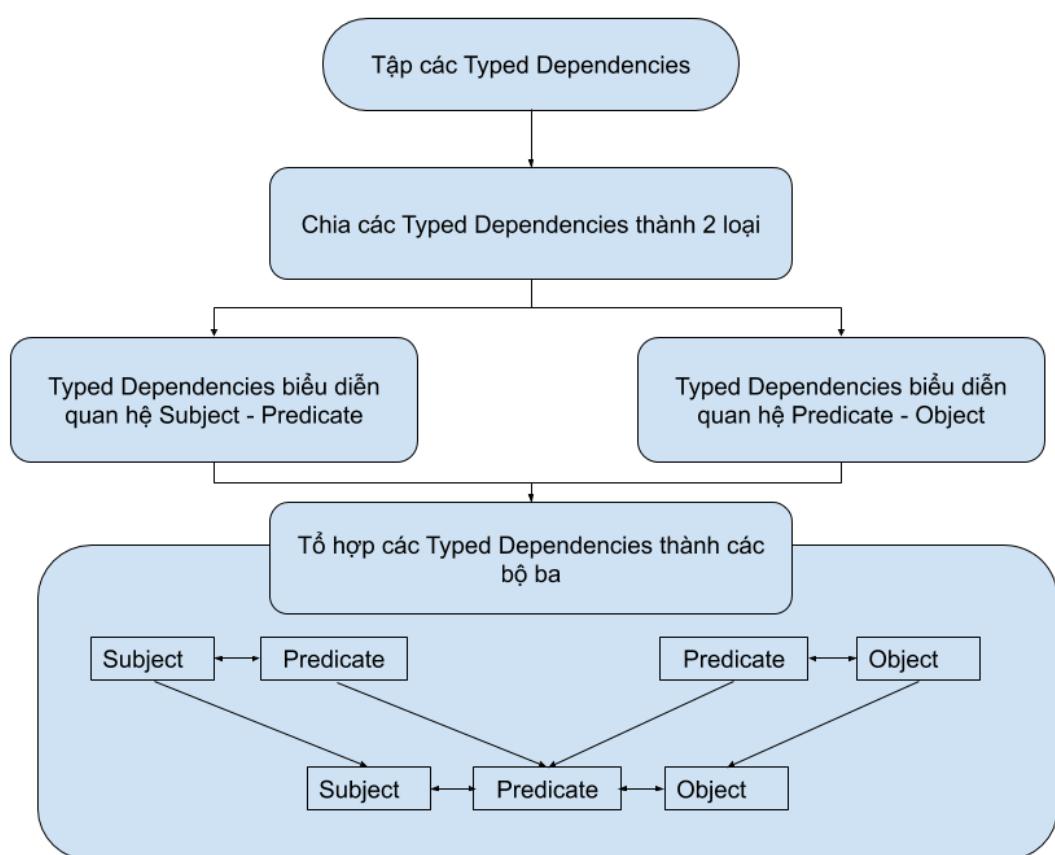
Để làm được bước này đầu tiên ta cần xác định nhãn và biến trong mô hình ngữ nghĩa. Module sẽ ánh xạ vào Cơ sở tri thức BKSport Ontology và kho lưu trữ AllegroGraph để xác định nhãn đó ứng với thực thể hay lớp nào.

Từ tập các kiểu đã được đánh dấu ở bước trên, ta tiến hành phân chia làm 2 loại:

Cặp kiểu quan hệ Subject – Predicate và Predicate – Object

Từ đây ta tổng hợp thành quan hệ hộ 3 bằng cách kết hợp từ 2 cặp trên:

Subject – Predicate – Object



▪ *Bước 4: Sinh câu truy vấn SPARQL trung gian*

Sau khi có quan hệ ngữ nghĩa là bộ 3 quan hệ, module tiếp theo sẽ sinh câu truy vấn SPARQL trung gian. Câu truy vấn gồm 3 phần như sau:

<Mệnh đề hỏi><Mệnh đề điều kiện><Mệnh đề tùy chọn ràng buộc>.

- Mệnh đề hỏi:

Có 2 loại câu hỏi là câu hỏi đúng/sai hoặc câu hỏi có từ đê hỏi.

Đê xác định một câu hỏi thuộc loại nào ta dựa vào kết quả của cây truy vấn đã được mô tả bước 2.

Và SPARQL đã mô tả 2 loại câu hỏi này bằng mệnh đề SELECT và ASK.

- Mệnh đề điều kiện:

Mệnh đề điều kiện chứa các bộ ba biểu diễn quan hệ của các đối tượng đã được phân tích từ bước 3 có dạng (?subject ?predicate ?object).

- Mệnh đề ràng buộc:

Các điều kiện ràng buộc của câu hỏi như về thời gian, về sự lựa chọn.

- **Bước 5 : Sinh câu truy vấn hoàn chỉnh**

Công việc sinh truy vấn SPARQL hoàn chỉnh sẽ từ câu truy vấn SPARQL trung gian.

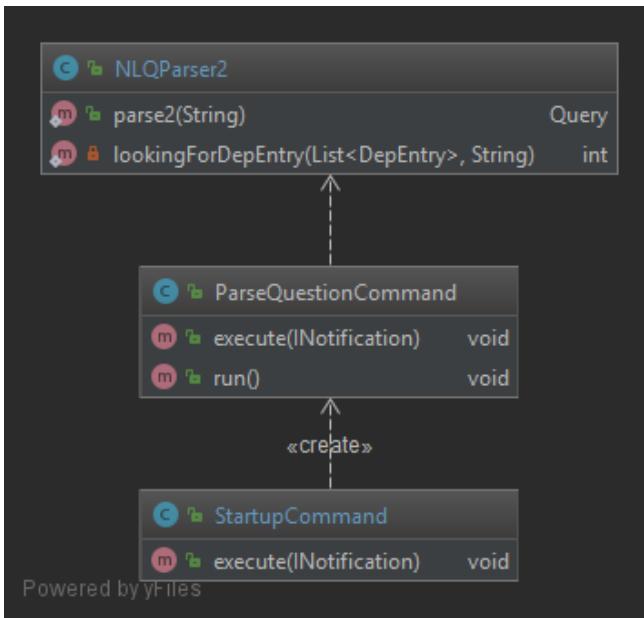
Từ câu trung gian, ta có mô hình bộ ba quan hệ. Kết hợp với vị từ, thực thể từ bước 2. Ta tiến hành sinh câu truy vấn hoàn chỉnh bằng các URI tương ứng.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX bksport: <http://bk.sport.owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX time: <http://www.w3.org/2006/time#>
ASK WHERE {
    GRAPH ?graph {
        . . . <http://bk.sport.owl#chelsea-fc> <http://bk.sport.owl#defeat> <http://bk.sport.owl#manchester-city-fc>.
    }
}
```

Ví dụ: Câu SPARQL tương ứng với câu hỏi “Did Chelsea FC defeat Manchester City FC?”

3.3.2 Phát triển tính năng sinh câu hỏi từ ngôn ngữ tự nhiên của Module Semantic Search

Để bắt thêm các dạng câu hỏi, em thiết kế thêm lớp NQLParse2(String nlQuestion) với đầu vào là câu hỏi dạng ngôn ngữ tự nhiên, đầu ra là câu hỏi dạng SPARQL. Đây là sơ đồ lời gọi Method giữa các lớp liên quan:



Từ hệ thống, khi nhận yêu cầu câu hỏi đáp của người dùng, StartupCommand được gọi và tại đây khởi tạo lớp ParseQuestionCommand, lớp này khi nhận được thông báo từ StartupCommand sẽ gọi tới lớp NLQParse2.

3.4 Phát triển tính năng trực quan hóa dữ liệu ngữ nghĩa trong hệ thống tích hợp thông tin BKSport

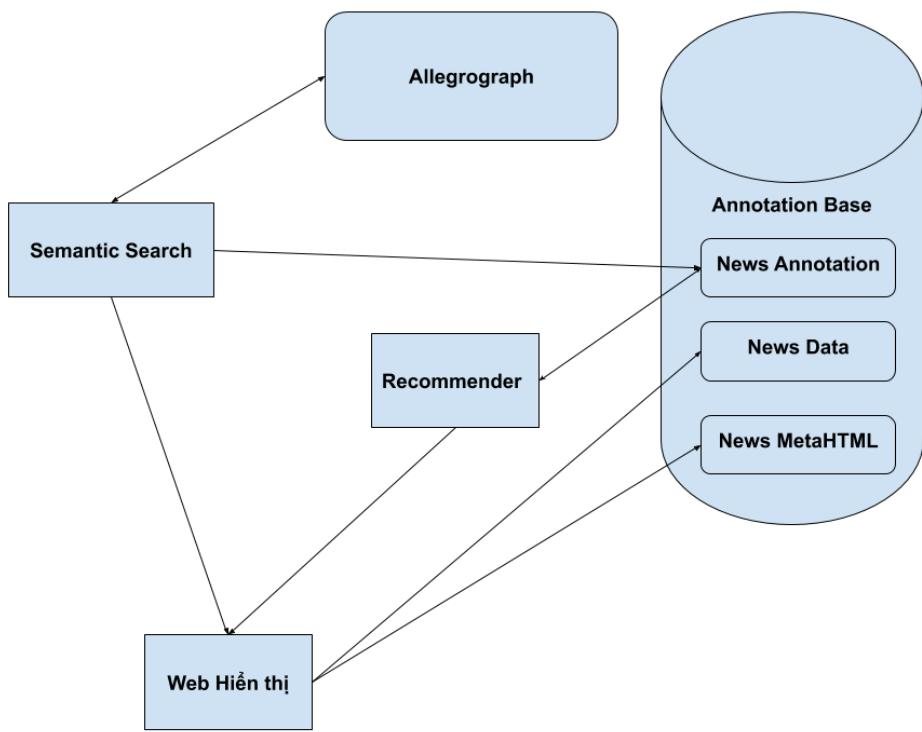
Trong hệ thống tích hợp thông tin BKSport, giao diện người dùng trước đây là màn hình dòng lệnh. Do đó để trực quan hóa, giúp người dùng dễ dàng thao tác với hệ thống, em đã xây dựng một ứng dụng Web nhỏ, gồm 3 chức năng chính:

- Đọc tin tức thể thao có trong hệ thống.
- Hỏi đáp ngữ nghĩa về các sự kiện, thông tin liên quan đến thể thao.
- Hiển thị, trực quan hóa dữ liệu ngữ nghĩa đã được sinh ra trong hệ thống cho người dùng xem.

Sau đây, em xin được trình bày vắn tắt về kiến trúc, thiết kế module Web giao diện người dùng. Thuật toán trực quan hóa ngữ nghĩa sẽ được trình bày ở chương tiếp theo.

3.4.1 Kiến trúc tổng thể Module Web cung cấp giao diện người dùng

Mô hình hóa giữa Module Web giao diện người dùng và các module còn lại được thể hiện như sau:



Hình 7: Sơ đồ kiến trúc module web giao diện người dùng

- Với chức năng hiển thị tin tức đơn thuần (như đọc báo, xem tin...), Web sẽ trực tiếp lấy dữ liệu và các tin tức đã được crawl về trong database (Data News).
- Với chức năng gợi ý tin tức liên quan, lúc này module Recommender sẽ thao tác với database là Annotation Base – chứa các ngữ nghĩa các bài báo để đưa ra các gợi ý, từ đây chuyển tới cho Web để hiển thị cho người dùng.
- Với chức năng tìm kiếm ngữ nghĩa, hệ thống sẽ gửi câu hỏi cho module Semantic Search. Từ đây Module Semantic Search sẽ chuyển câu hỏi từ dạng câu hỏi tự nhiên, trả về cho Module Web câu hỏi dạng SPARQL. Module Web sẽ gửi câu hỏi lên AllegroGraph. AllegroGraph ngoài nhiệm vụ là lưu trữ dữ liệu ngữ nghĩa online, còn cung cấp Engine Search ngữ nghĩa với đầu vào là câu hỏi SPARQL. Module Web nhận câu trả lời từ AllegroGraph là các id về thực thể hoặc bài báo. Từ đây Web Module sẽ truy vấn vào database là Data News để hiển thị câu trả lời cho người dùng.
- Với chức năng trực quan hóa dữ liệu ngữ nghĩa, dữ liệu ngữ nghĩa đã được đánh dấu vào văn bản dạng HTML, em gọi là MetaHTML đã được sinh ra khi chú thích ngữ nghĩa một bài báo và lưu ở database News MetaHTML. Việc sinh MetaHTML em xin phép được nêu ở chương tiếp theo.

Như vậy, ta thấy ngoài chức năng là hiển thị thông tin bài báo cho người dùng, Module Web còn cung cấp khá nhiều dịch vụ cho người dùng. Tiếp tới đây em xin được trình bày về thiết kế lớp và luồng hoạt động của module này.

3.4.2 Thiết kế chi tiết Module Web

1. Định hướng giải pháp tiếp cận

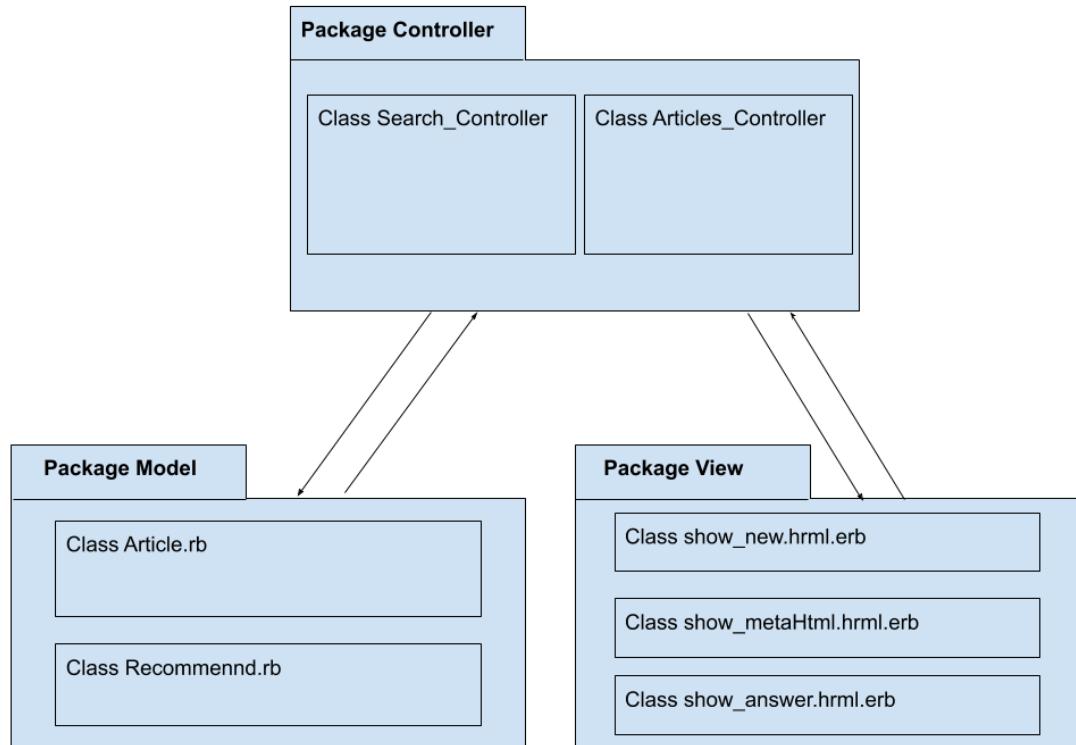
Để xây dựng trang web cho người dùng truy nhập và thực hiện các chức năng như ở phần kiến trúc đã nêu, em đã xây dựng trang web gồm 2 phần là Frontend và Backend.

Trong đó, BackEnd là phần xử lý của module web, được viết bằng ngôn ngữ Ruby, sử dụng framework Ruby on Rails. Em lựa chọn Ruby on Rails bởi ngôn ngữ này được thiết kế theo mô hình MVC dễ tiếp cận và phát triển. Sử dụng template giúp việc kết hợp giữa Rails và HTML dễ dàng, cho ta thời gian xây dựng module nhanh chóng.

FrontEnd hiển thị nội dung Web, em sử dụng ngôn ngữ HTML kết hợp CSS để căn chỉnh giao diện. Javascript, Angular Js để tạo hiệu ứng, nhận biết và xử lý thao tác người dùng tương tác với hệ thống.

2. Thiết kế chi tiết gói

Module Web – cụ thể là phần WebServer được thiết kế theo mô hình MVC ứng với 3 gói là Model, gói View và gói Controller.



Hình 8: Sơ đồ thiết kế các gói trong module Web giao diện người dùng

Gói Controller gồm 2 gói là : Search_Controller và Articles_Controller

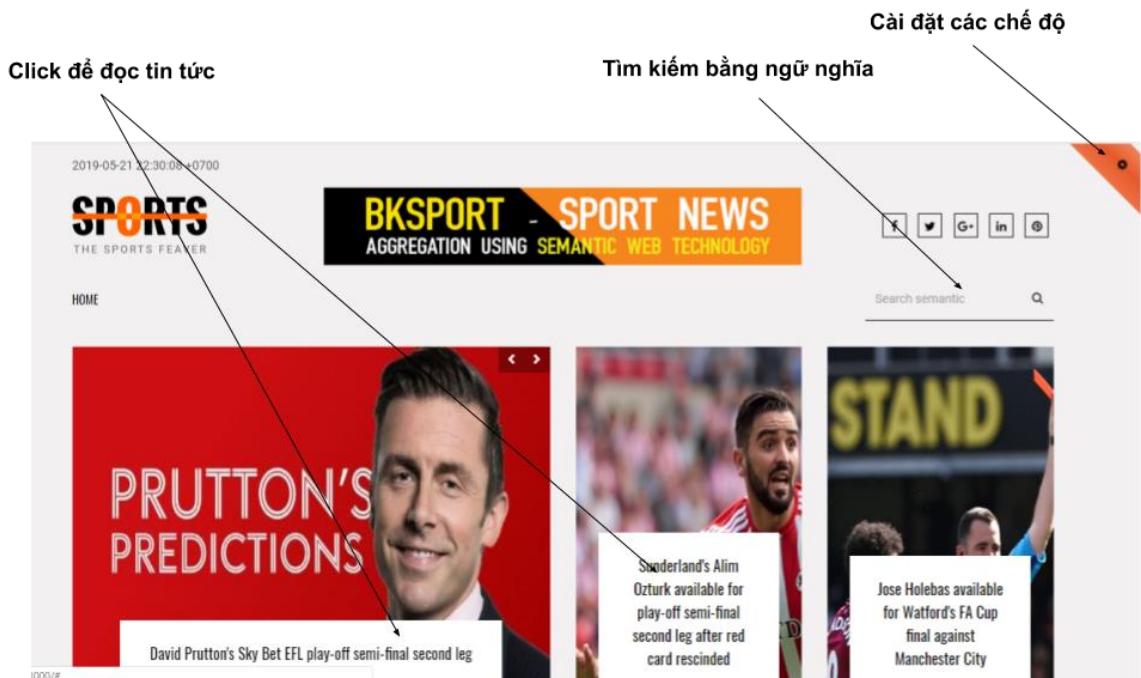
- Search_Controller có chức năng kiểm soát các luồng logic về việc người dùng hỏi đáp với hệ thống
- Articles_Controller có chức năng kiểm soát các luồng logic về hiển thị, đọc tin tức với các bài báo.

Gói Model gồm 2 Model chính là:

- Article.rb chứa thông tin về bài viết gồm: nội dung, tiêu đề, dữ liệu metadata.
- Recommender.rb chứa thông tin các bài báo liên quan, gợi ý về các bài báo tương tự.

3. Thiết kế chức năng chương trình.

Trên nhiệm vụ đã nêu ở phần 3.4, theo thiết kế module và định hướng tiếp cận, em đã thiết kế chương trình như sau:



Ở giao diện trang chủ, người dùng có thể ngay lập tức click vào các bài báo để xem tức, đọc tin một cách bình thường như các trang tổng hợp thông tin khác.

Ở góc phía bên phải trang chủ, ta có tùy chọn cài đặt và chức năng tìm kiếm thông tin.

Xem các bài viết trước đó hoặc tiếp theo

departure of Steve McLaren.

Hit play on the videos above to see all of Merson's predictions in full as well as some great offers from Sky Bet for the action this weekend.

Previous

Rafa Benitez cancels Newcastle players' day off hours after taking over as manager

Next

Rafa Benitez to Newcastle: Which men form his coaching team?

Recommend V1

Man City need nine wins for Premier League title, says Manuel Pellegrini

7 months ago by Jhonsone

Manuel Pellegrini feels Manchester City must win nine of their last 10 games to reclaim the Premier League title.

Spain going to made class football

9 Comments

Spain going to made class football

20 Comments

Spain going to made

Chức năng gợi ý tin tức

Cuối mỗi tin tức, hệ thống gửi các bài báo gợi ý, hoặc duyệt bài trước hoặc sau liền kề.

Setting

Search engine

Keywords Semantic

SPARQL on search

Show SPARQL(only semantic mode)

Article option

Show metadata Show meta HTML

Recommend mode

Semantic 1 Semantic 2

Save Close

Khi click vào tùy chọn cài đặt, người dùng có thể tùy chọn các yêu cầu như tìm kiếm theo keyword hoặc tìm kiếm bằng ngữ nghĩa, hiển thị câu truy vấn SPARQL, xem tin tức thông thường hoặc tin tức đã được trực quan ngữ nghĩa, chế độ gợi ý 1 hoặc 2.

Ở chức năng tìm kiếm ngữ nghĩa, người dùng khi truy vấn câu hỏi tự nhiên, hệ thống sẽ trả về kết quả là các đối tượng hoặc bài báo, hoặc cả hai (tùy câu hỏi) mà từ đây hiển thị cho người dùng.

WHAT ARE YOU LOOKING FOR?

which team defeat Chelsea ?

Search

Kết quả các đối tượng cần tìm 5

Manchester United
Sunderland
Jordan Pickford
Everton
Real Madrid

Show more

Các bài viết liên quan: 11

Man City, Man Utd or West Ham? Charlie Nicholas predicts who will secure the final Champions League spot

We challenged Soccer Saturday pundit Charlie Nicholas to predict how the battle for the fourth Champions League

Manchester United

Kết quả chi tiết của phần này sẽ được tiếp tục trình bày ở các chương tiếp theo.

Chương 4 Các giải pháp và đóng gópnổi bật

Trong phần trước, em đã lần lượt trình bày về việc thiết kế để tăng dữ liệu ngữ nghĩa được nhận diện ở Module Semantic Annotation, tiếp đến là thiết kế thêm lớp để tăng khả năng sinh câu hỏi ở Module Semantic Search. Và cuối cùng là thiết kế Module Web hiển thị để trực quan hóa dữ liệu ngữ nghĩa.

Trong phần này, em xin được đi sâu vào các đóng góp nổi bật của mình ở các phần trên, đồng thời làm rõ các thuật toán sử dụng được áp dụng vào từng phần.

4.1 Làm giàu và cập nhật cơ sở tri thức

Cơ sở tri thức trong BKSport – BKSport Ontology là tập dữ liệu ngữ nghĩa về các nhân vật, tổ chức, sự kiện... và như đã phân tích ở các chương trước: Với **Module Semantic Annotation** và **Module Semantic Search** việc nhận diện thực thể và nhận diện quan hệ trong văn bản cần sự giúp đỡ từ **BKSport Ontology**. Vậy nên cơ sở tri thức càng nhiều và đầy đủ thì càng nâng cao hiệu quả các module khác.

Đặc thù trong hệ thống BKSport là cơ sở tri thức liên quan đến thể thao. Và lĩnh vực thể thao có đặc điểm là luôn cập nhật, thay đổi theo thời gian. Ví dụ như một cầu thủ được chuyển nhượng hoặc thi đấu cho nhiều câu lạc bộ, một câu lạc bộ thay đổi Huấn luyện viên...

Vậy nhằm nâng cao độ chính xác và tăng cường vùng nhận diện cho BKSport Ontology, em đề xuất việc làm giàu cơ sở tri thức.

Cũng như đã phân tích, việc làm giàu cơ sở tri thức gồm 2 phần:

- Bổ sung dữ liệu về các thực thể
- Bổ sung dữ liệu về các quan hệ giữa các thực thể

4.1.1 Bổ sung dữ liệu về các thực thể

Bổ sung thêm ngữ nghĩa cho **BKSport Ontology** sẽ thêm dữ liệu cho các lớp thực thể sau:

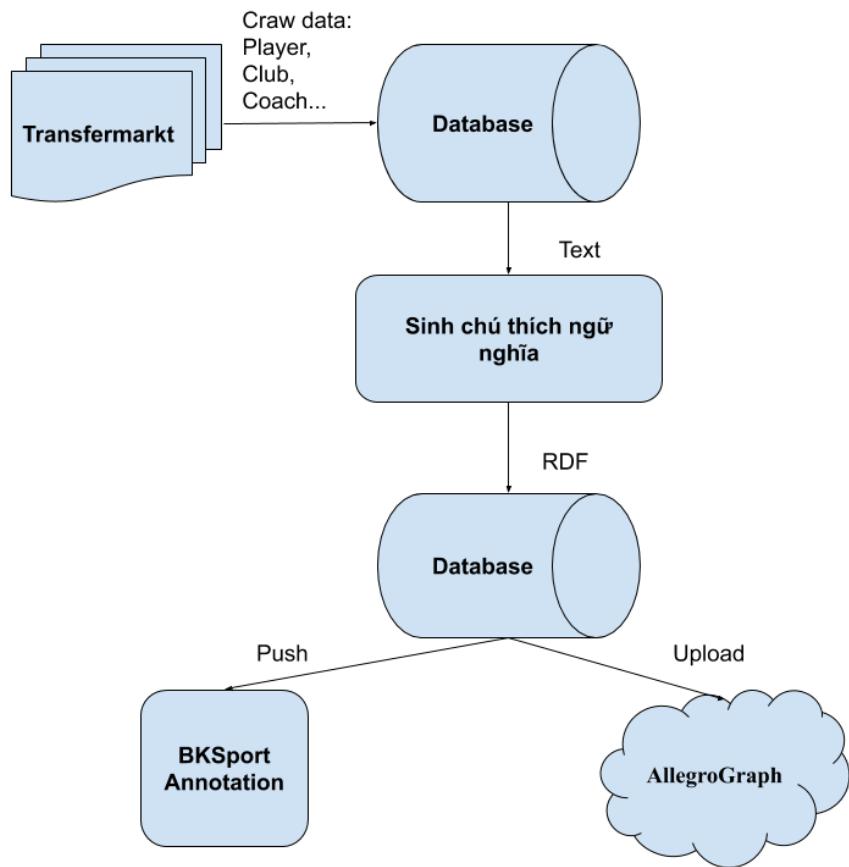
- Câu lạc bộ bóng đá.
- Cầu thủ bóng đá.
- Huấn luyện viên bóng đá.
- Sân vận động bóng đá.
- Giải đấu bóng đá.

Để tăng lượng cơ sở tri thức, cần lấy từ nguồn cung cấp đáng tin cậy, trong trường hợp này em đã sử dụng từ nguồn **Transfermarkt** – Là trang web chứa các thông tin về cầu thủ, đội bóng, thông tin chuyển nhượng làm nguồn cung cấp dữ liệu.

SQUAD OF MANCHESTER CITY				
 ⓘ The club's landing page - find all relevant information like the actual squad, relevant news, recent rumours and ... ✉️ 🌐 📱				
Select season		17/18	Show	
#	Player(s)	born/age	Nat.	Market value
31	 Ederson Keeper	 Aug 17, 1993 (24)		45,00 Mill. € ↑
1	 Claudio Bravo Keeper	Apr 13, 1983 (35)	 	5,00 Mill. € ↓
5	 John Stones Centre-Back	May 28, 1994 (23)		50,00 Mill. € ↑
14	 Aymeric Laporte Centre-Back	 May 27, 1994 (23)	 	50,00 Mill. € ↑
30	 Nicolás Otamendi Centre-Back	Feb 12, 1988 (30)		35,00 Mill. € ↑
4	 Vincent Kompany C Centre-Back	Apr 10, 1986 (32)		10,00 Mill. € ↓
22	 Benjamin Mendy Left-Back	 Jul 17, 1994 (23)	 	40,00 Mill. € ↑
2	 Kyle Walker Right-Back	 May 28, 1990 (27)		50,00 Mill. € ↑
3	 Danilo Right-Back	 Jul 15, 1991 (26)		18,00 Mill. € ■
25	 Fernandinho Defensive Midfield	May 4, 1985 (33)		15,00 Mill. € ↓
8	 Ilkay Gündogan Central Midfield	Oct 24, 1990 (27)		30,00 Mill. € ■
18	 Fabian Delph Central Midfield	Nov 21, 1989 (28)	 	10,00 Mill. € ↑

Hình 9: Trang tin Transfermarkt

Quy trình làm giàu cơ sở tri thức như sau :



Hình 10: Sơ đồ làm giàu cơ sở tri thức

Quá trình làm giàu cơ sở tri thức được thực hiện diễn qua ba giai đoạn:

- Dữ liệu từ nguồn tin cậy được trích xuất và lấy về Database theo các trường.
- Các thông tin trích xuất là các thông tin liên quan đến cầu thủ, huấn luyện viên, câu lạc bộ... theo các trường chỉ định, từ đây ta tiến hành sinh ra RDF là các chú thích ngữ nghĩa theo mẫu có sẵn. Quy trình được tiến hành theo phương pháp ánh xạ từ các thuộc tính trên database với các thuộc tính ngữ nghĩa đã được thiết kế ở mục 3.2.2.

Ví dụ: Cầu thủ Neymar, là một cầu thủ chạy cánh trái, hiện tại đang chơi cho Paris Saint Germain, đã từng chơi cho Barcelona và Santos FC, thì ngữ nghĩa được sinh ra là:

```
1. <owl:NamedIndividual rdf:about="http://bk.sport.owl#neymar">
2.   <rdfs:label xml:lang="en">neymar</rdfs:label>
3.   <protons:mainLabel>neymar</protons:mainLabel>
```

```

4.  <rdf:type rdf:resource="&bksport;Left-Winger"/>
5.  <protons:generatedBy rdf:resource="http://bk.sport.owl"/>
6.  <playFor rdf:resource="&bksport;paris-saint-germain"/>
7.  <hasPlayed rdf:resource="&bksport;paris-saint-germain"/>
8.  <hasPlayed rdf:resource="&bksport;fc-barcelona"/>
9.  <hasPlayed rdf:resource="&bksport;santos-fc"/>
10. <hasPlayed rdf:resource="&bksport;santos-fc-u20"/>
11. </owl:NamedIndividual>

```

- Lưu trữ dữ liệu dạng RDF, nạp vào Cơ sở tri thức (BKSport, KIM, AllegroGraph Server)

→ Kết quả chương trình đã gom thêm cho hệ thống :

- Gần 660 đội bóng, sân vận động.
- Gần 15000 thông tin các cầu thủ.
- Gần 600 thông tin các huấn luyện viên liên quan.

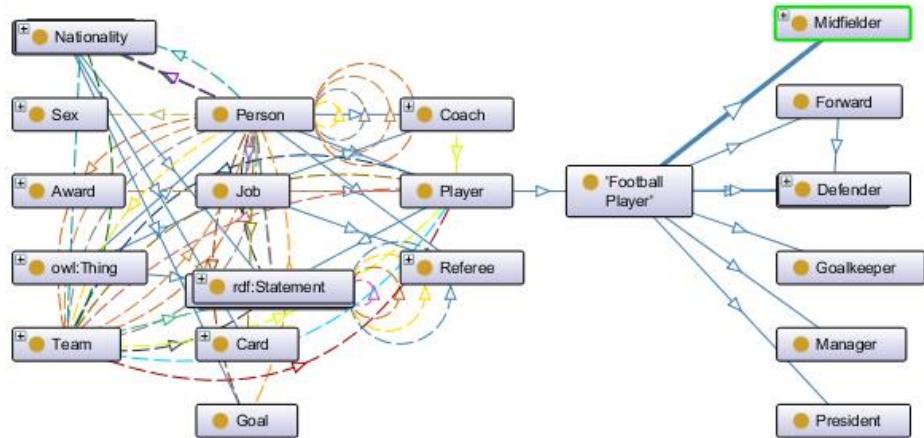
4.1.2 Bổ sung dữ liệu về các Ontology

Cũng như bổ sung dữ liệu về các thực thể, dữ liệu về khái niệm và các các quan hệ giữa các thực thể cũng đóng vai trò quan trọng trong nhận biết các quan hệ cũng như sinh chú thích ngữ nghĩa. Dưới đây là các quan hệ đã có hiện tại của hệ thống:

Để tăng khả năng phong phú cho nhận diện ngữ nghĩa, em đề xuất thêm một số Ontology bổ sung thêm các lớp ý nghĩa mới:

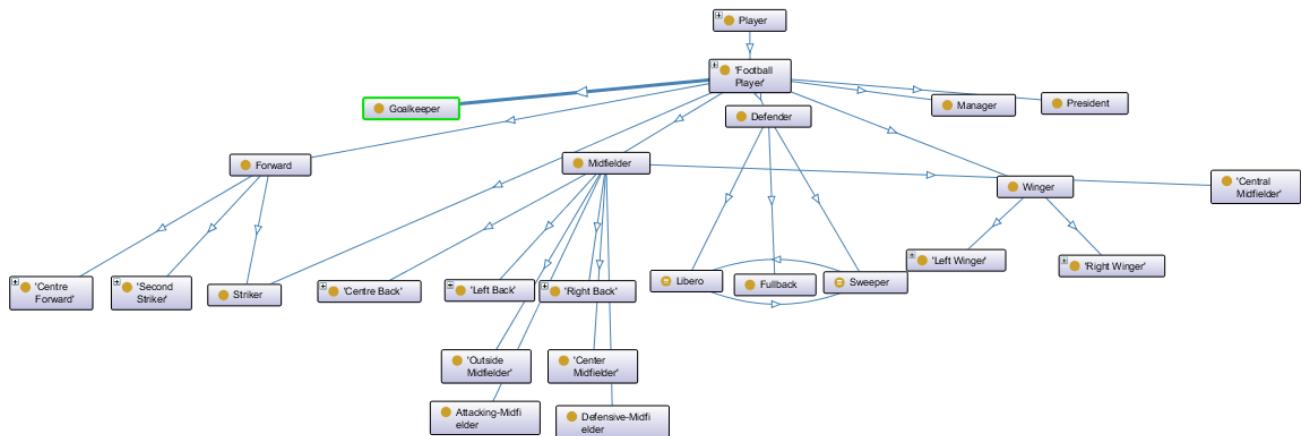
+ Về lớp Person:

Đây là các ontology mà hệ thống hiện tại đang có:



Em xin đề xuất thêm mới các thuộc tính như sau:

- Về ngữ nghĩa cầu thủ thể thao, đặc biệt là về **Football Player**, em đã bổ sung khái niệm sau:



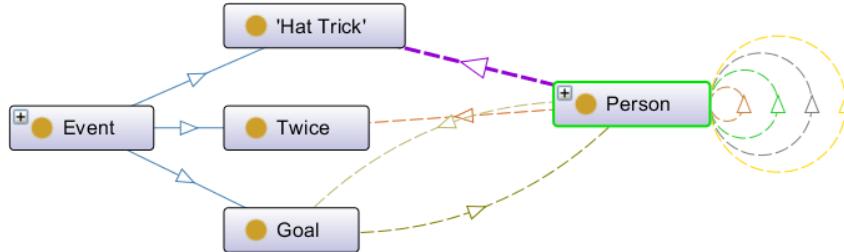
Các thuộc tính ngữ nghĩa Ontology bao gồm:

Tên Ontology	Thuộc tính ngữ nghĩa liên quan
<Winger>	<subClassOf><FootballPlayer>
<Left-Winger>	< subClassOf ><Winger>
<Right-Winger>	< subClassOf ><Winger>
<Central-Midfield>	< subClassOf ><Midfielder>
<Centre-Back>	< subClassOf >< Midfielder >

<Left-Back>	< subClassOf >< Midfielder >
<Right-Back>	< subClassOf >< Midfielder >
<DefensiveMidfielder>	< subClassOf >< Midfielder >
<AttackingMidfielder>	< subClassOf >< Midfielder >
<Striker>	< subClassOf >< Forward >
<Second-Striker>	< subClassOf >< Forward >

Bảng 8: Các thuộc tính ngữ nghĩa bổ sung vào Ontology lớp Player

- Về Lớp Event, em bổ sung các Ontology:



Em đã bổ sung về các sự kiện bóng đá của một cầu thủ như: Hattrick, Twice, Goal

Tên Ontology	Thuộc tính ngữ nghĩa liên quan
<Hattrick>	<subClassOf><Event> <Person><makeHattrick>
<Twice>	<subClassOf><Event> <Person><makeTwice>
<Goal>	< subClassOf ><Event> <Person><score> <scoreBy><Person>

Bảng 9: Các thuộc tính ngữ nghĩa bổ sung vào Ontology lớp Event -1

Bổ sung về các quan hệ bao gồm:

Tên Ontology	Các quan hệ liên quan
<Say>	<domain><Person> <subPropertyOf><Happen> <type><String> <hasResource><statement>
<Statement>	<hasSubject><Thing> <hasPredicate><Thing>

	<hasObject><Thing> <Content><String>
<makeHattrick>	<domain><Person> <range><HatTrick>
<makeTwice>	<domain><Person> <range><Twice>
<hasNumberOfGoal>	<domain><Person> <range><integer>
<match_result>	<type><Result> <has Abstract><string> <contain><Team>

Bảng 10: Các thuộc tính ngữ nghĩa bổ sung vào Ontology lớp Event -2

Sau khi bổ sung thêm các cơ sở tri thức trên, hệ thống đã mở rộng được vùng nhận biết trong chủ thích ngữ nghĩa. Các thực thể và quan hệ đã được nhận diện chính xác độ phủ sóng lớn.

4.2 Bổ sung nhận biết các quan hệ mới trong Module Semantic Annotation

Cùng với việc làm giàu cơ sở tri thức, việc nhận diện các quan hệ mới giúp tăng khả năng nhận diện chủ thích ngữ nghĩa trong Module Semantic Annotation. Trong phần này, em sẽ trình bày về việc nhận biết và sinh các quan hệ mới trong miền thông tin bóng đá.

4.2.1 Nhận biết quan hệ Trích dẫn gián tiếp

1. Mô hình hóa quan hệ

Trước hết, ta có ví dụ minh họa sau:

Pochettino said that "Messi will help Barcelona win against Chelsea this season".

Trước hết, hệ thống có thể bắt được quan hệ là:

<Pochettino><say><"Messi will help Barcelona win against Chelsea this season">

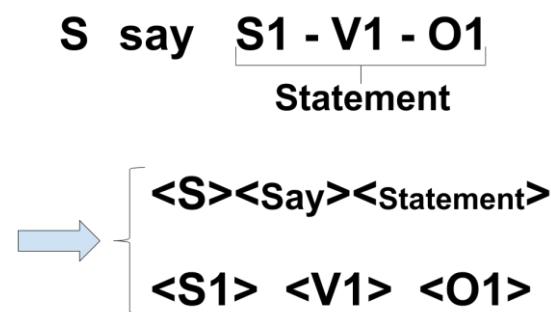
Ta thấy rằng, ngữ nghĩa vẫn còn trong câu là:

<Barcelona><win><Chelsea>

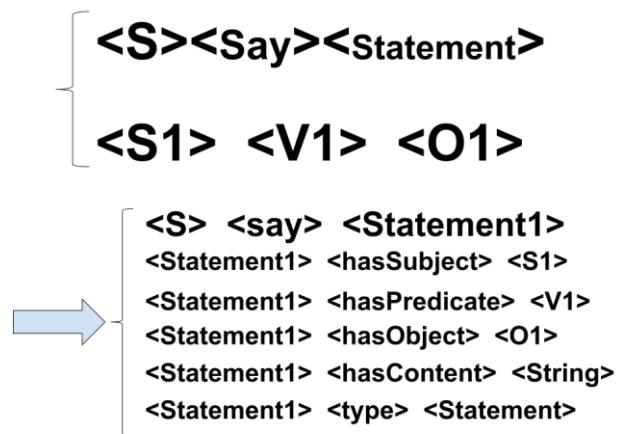
Đĩ nhiên, hệ thống có thể bắt được ngữ nghĩa này một cách riêng rẽ, thế nhưng nếu vậy sẽ ảnh hưởng đến liên kết giữa quan hệ liên kết trong câu. Bởi ta thấy rằng khi quan hệ trong câu khi một người nói là quan hệ chủ quan, nó không thể hiện tính thông tin chính xác trong câu. Tuy nhiên lượng thông tin này cũng cần thiết để trích xuất và chú thích ngữ nghĩa, bởi nó thể hiện quan điểm, ý kiến của một cá nhân nào đó.

Vậy nên đây là vấn đề cần đặt ra là tạo ra mô hình miêu tả để trực quan hóa quan hệ trong câu trên.

Để mô hình quan hệ này em mô hình hóa dạng câu này như sau:



Nếu ta thực hiện việc mô hình hóa và trích xuất ngữ nghĩa như bình thường, ta khó lòng có thể mô tả được quan hệ lồng nhau như trên, bởi vậy em đề xuất biểu diễn ngữ nghĩa dưới dạng sau:



Bằng việc sử dụng mô hình biểu diễn như trên, giờ đây ta có thể lưu trữ, khai thác dữ liệu một cách dễ dàng hơn.

2. Thuật toán nhận dạng quan hệ

Để nhận dạng thuật toán này, cũng như trong hệ thống sử dụng, em sử dụng ngôn ngữ JAPE để thể hiện thuật toán này. Tuy nhiên, trong đồ án này, em chỉ xin được mô tả thuật toán bằng dạng giả ngữ như sau:

Thuật toán nhận dạng quan hệ Trích dẫn gián tiếp

Input:

- Mẫu quan hệ dưới dạng text
- Các thực thể nhận dạng được từ Ontology sau khi được ánh xạ

Output:

- Bộ ba thể hiện quan hệ

Thuật toán:

Với : P là mẫu dạng A say B

```
foreach(Annotation p in P) do{
    statement = p.get("B");
    // annotates statement
    annotationSet = BKSport.annotate(statement);
    for each(Annotation annotation in annotationSet){
        if(annotation.contains("semantic")){
            // Creates statement which is same annotation
            subject= annotation.get("subject");
            predicate= annotation.get("predicate");
            object= annotation.get("object");
            // Generate triples:
            <A><bksport:saidThat><statement>;
            <statement><rdf:subject> subject;
            <statement><rdf:predicate> predicate;
            <statement><rdf:object> object;
            <statement><rdf:hasContent> <String>;
        }
    }
}
```

3. Dạng quan hệ khi sau khi được trích xuất

Tùy quan hệ bộ 3 sau khi nhận dạng, ta sẽ tiến hành trích xuất ngữ nghĩa và lưu dưới dạng RDF. Với mỗi bộ 3, ta sẽ tiến hành tạo định dạng, xuất kiểu và đưa ra mẫu RDF.

Đây là ví dụ về một mẫu ngữ nghĩa khi đã trích xuất.

```
1. <rdf:Description rdf:about="http://bk.sport.owl#pochettino">
2.   <j.0:say rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#statement1"/>
3. </rdf:Description>
4. <rdf:Description rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#statement1">
5.   <rdf:subject rdf:resource="http://bk.sport.owl#Barcelona-fc"/>
6.   <rdf:predicate rdf:resource="http://bk.sport.owl#defeat"/>
7.   <rdf:object rdf:resource="http://bk.sport.owl#chelsea-fc"/>
8.   <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
9.   <rdf:sayStatement rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Messi will help
p Barcelona win against Chelsea this season </rdf:sayStatement>
```

4.2.2 Nhận biết quan hệ Kết quả trận đấu và sinh ngữ nghĩa đối lập

Cũng như quan hệ trên, trước khi đi vào xây dựng mô hình ngữ nghĩa, ta sẽ xét qua ví dụ sau: “Crystal Palace 2-1 Stoke in their final home game of the season to finally secure mathematical survival from Premier League relegation.”

Với quan hệ trên, ta thấy rằng bằng mắt thường ta nhận ra quan hệ sau:

< Crystal Palace FC> <Defeat> <Stoke City FC>

Với con người, ta nhận ra điều này khá rõ ràng, tuy nhiên với máy phải hiểu được quan hệ này dưới các bước:

- Nhận diện được Crystal Palace, Stoke là tên của đội bóng
- “2-1” là tỉ số trận đấu của một trận đấu → đây là một trận đấu giữa 2 đội bóng
- $2 > 1 \rightarrow$ Crystal Palace FC chiến thắng trước Stoke City FC

Trong phần này, em sẽ mô tả các bước mô hình, nhận diện và trích xuất dữ liệu và ngữ nghĩa về kết quả của một trận đấu. Đồng thời sinh ra quan hệ đối lập đồng nghĩa với quan hệ trên. Ví dụ:

< Crystal Palace FC> <Defeat> <Stoke City FC>

Thì ta cũng có thể sinh ra quan hệ:

<Stoke City FC><Lose> < Crystal Palace FC>

1. Mô hình hóa quan hệ

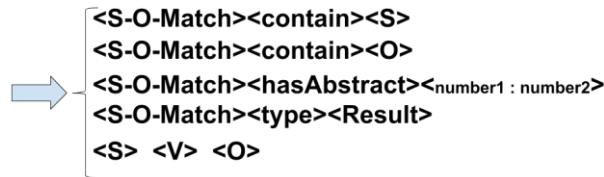
Trước hết, với mong muốn không chỉ lấy được kết quả trận đấu chỉ là thắng hay thua, em mong muốn lấy thêm được tỉ số trận đấu nhằm tạo độ phong phú cho ngữ nghĩa và thêm độ tin cậy và trực quan cho kết quả ngữ nghĩa được sinh ra.

Em xin được mô tả mô hình kết quả của một trận đấu có chứa tỉ số dưới các dạng thường gặp sau:

S V O [number1 : number2]

S [number1 : number2] O

S V O <token> [number1 : number2]



Vậy vấn đề ở đây là mô hình **< S [number1: number2] O >** vì đây là mô hình phi cấu trúc

Trong phạm vi đề tài này, em cũng tiến hành nhận biết thêm một số quan hệ phi cấu trúc (không phải là **<S - V - O>**) sẽ được trình bày ở mục kế tiếp.

2. Thuật toán nhận diện quan hệ

Dưới đây là thuật toán nhận diện quan hệ này viết bằng giả mã:

Ý tưởng: Với mẫu quan hệ bắt được, ta tiến hành so sánh cụm số, từ đó sinh ra kết quả trận đấu, tìm ngữ nghĩa đối lập với kết quả rồi trích xuất quan hệ.

Thuật toán nhận dạng quan hệ kết quả trận đấu và sinh ngữ nghĩa đối lập

Input:

- Mẫu quan hệ dưới dạng text
- Các thực thể nhận dạng được từ Ontology sau khi được ánh xạ

Output:

- Bộ ba thể hiện quan hệ

Thuật toán:

```
Với : P là mẫu dạng (S V O n1:n2) hoặc (S n1:n2 O)
foreach(Annotation p in P) do{
    subject= p.get("subject");
    predicate=p. get("predicate");
    object= p. get("object");
    n1= p.get("number1")
    n2= p.get("number2")
    if(predicate == null){
        //compare results
        if(n1>n2){ predicate1 = defeat}
        else if(n1 == n2){predicate1 = lost}
        else{predicate1 = lose}
    }
    // Generate triples:
    if(predicate != null){
        // find opposing relations
        predicate1 = findOpRelation(predicate);
        <subject><predicate><object>;
        <subject><predicate1><object>;
```

```
<S-O-match>< contain><subject>;
<S-O-match>< contain><object>;
<S-O-match>< hasAbstract><n1:n2>;
<S-O-match>< type><Result>;
} else {
    predicate2 = findOpRelation(predicate1);
    <subject><predicate1><object>;
    <subject><predicate2> <object>;
    <S-O-match>< contain><subject>;
    <S-O-match>< contain><object>;
    <S-O-match>< hasAbstract><n1:n2>;
    <S-O-match>< type><Result>;
}
}
```

3. Dạng quan hệ sau khi được trích xuất

Sau khi nhận diện được quan hệ bộ ba từ thuật toán, ta sẽ trích xuất để sinh ra dữ liệu ngữ nghĩa dưới dạng RDF. Đây là mẫu 1 dạng RDF sau khi được trích xuất:

```
1.  <rdf:Description rdf:about="http://bk.sport.owl#crystal-palace">
2.    <j.0:defeat rdf:resource="http://bk.sport.owl#stoke-city"/>
3.  </rdf:Description>
4.  <rdf:Description rdf:about="http://bk.sport.owl#stoke-city">
5.    <j.0:lose rdf:resource="http://bk.sport.owl#crystal-palace"/>
6.  </rdf:Description>
7. <rdf:Description rdf:about="http://bk.sport.owl#crystal-palace_stoke-
   city_match">
8.   <j.0:contain rdf:resource="http://bk.sport.owl#stoke-city"/>
9.   <j.0:contain rdf:resource="http://bk.sport.owl#crystal-palace"/>
10.  <rdf:type rdf:resource="http://bk.sport.owl#Result"/>
11.   <j.0:hasAbstract rdf:datatype="http://www.w3.org/2001/XMLSchema#string">2:1</j
        .0:hasAbstract>
12. </rdf:Description>
```

4.2.3 Nhận biết quan hệ là biến thể của dạng S – V – O

Trong hệ thống hiện tại, chủ yếu bắt các quan hệ có dạng S – V – O. Tuy nhiên do đầu vào là dạng ngôn ngữ tự nhiên, nên không phải quan hệ nào cũng có dạng chuẩn S – V – O

Tiêu biểu trong dạng này có dạng quan hệ mà ta đã bắt ở mục trước < S [number1: number2] O>, vậy nên việc nhận dạng thêm các mối quan hệ là biến thể của dạng S – V – O cũng đóng vai trò quan trọng trong việc nâng cao hiệu quả của việc tăng khả năng vùng nhận diện các ngữ nghĩa trong hệ thống.

1. Hệ thống và mô hình một số quan hệ phi cấu trúc

Trước khi bước vào hệ thống các quan hệ phi cấu trúc, ta biết rằng việc hệ thống các quan hệ này là không đơn giản và không thể thực hiện bằng máy tính. Bởi các cấu trúc này đều dưới dạng ngôn ngữ tự nhiên, mà ngôn ngữ tự nhiên thì bao gồm các lỗi nói ẩn dụ, gián tiếp... Vậy nên em chỉ xin tổng hợp một số quan hệ phi cấu trúc phổ biến trong lĩnh vực thể thao và hẹp hơn là trong lĩnh vực bóng đá.

- Dạng 1: S'(N/Adj/noun phrase) <over> O (trong đó N là một danh từ)

Ta sẽ chuyển về dạng bộ 3 ngữ nghĩa :

<S> <V> <O>

Ví dụ minh họa: "Chelsea's victory over Sunderland in champion league"

Ta sẽ có ngữ nghĩa:

<Chelsea FC> <defeat> <Sunderland FC>

- Dạng 2: (N/Adj/noun phrase) S O (Trong đó N là một danh từ)

Ví dụ minh họa:

“Barca recovery workout after the victory against Chelsea”

Ta có ngữ nghĩa sau:

<Barcelona FC> <defeat> <Chelsea FC>

Vậy mâu chốt ngoài việc nhận dạng các ngữ nghĩa, ta cần có bộ định nghĩa các quan hệ dựa trên các danh từ quan trọng trong mô hình này. Ở đây em xin được liệt kê một số ánh xạ từ Danh từ, cụm danh từ, danh động từ, tính từ sang động từ trong mô hình quan hệ.

Danh Từ	Động từ thuộc tính
The Victory/ Victory	<defeat>/<win>
left behind	<defeat>/<win>
destroyed	<defeat>/<win>
left dazed	<lose>
edge	<defeat>/<win>
step back	<lose>

2. Thuật toán nhận dạng ngữ nghĩa

Dưới đây là thuật toán được dùng để nhận dạng ngữ nghĩa của các mô hình trên

Thuật toán nhận dạng quan hệ là biến thể S - V - O

Input:

- Mẫu quan hệ dưới dạng text
- Các thực thể nhận dạng được từ Ontology sau khi được ánh xạ

Output:

- Bộ ba thể hiện quan hệ

Thuật toán:

Với : P là mẫu dạng : (N/Adj/noun phrase) S O hoặc S'(N/Adj/noun phrase) <over> O

foreach(Annotation annotation **in** P) **do**{

```

// get N/Adj/noun phrase
n = getN(N/Adj/noun phrase)
// Creates annotation
subject= annotation.get("subject");
object= annotation.get("object");
predicate= getPredicateFormList(n);
// Generate triples:
<subject> <predicate> <object>
}
```

4.3 Bổ sung nhận biết các dạng câu hỏi mới trong Module Semantic Search

Nhằm tăng khả năng đáp ứng đa dạng các loại câu hỏi đáp trong Module Semantic Search. Việc mở rộng các mẫu câu hỏi mới giúp hệ thống đa dạng với mức độ cao hơn. Trong phần này, em xin được trình bày các mở rộng về các mẫu câu hỏi bao gồm : Mẫu câu hỏi về Trích dẫn gián tiếp, Mẫu câu hỏi bao kết hợp hoặc chọn lựa, mẫu câu hỏi về một sự kiện.

4.3.1 Lớp câu hỏi về trích dẫn

Đây là một loại câu hỏi đặc biệt, cũng như trong khi phân tích về sinh ngữ nghĩa cho quan hệ trích dẫn gián tiếp, ta biết rằng một ngữ nghĩa khi một người nào đó là một ngữ nghĩa mang tính chủ quan, vậy nên nó không thể coi là một thông tin chính xác, tuy nhiên nó lại mang nhiều ý nghĩa cho việc tham khảo, cũng như nhận xét quan điểm của một người nào đó.

1. Mô hình hóa lớp câu hỏi

Trước hết, em xin được mô hình hóa dạng câu hỏi để việc phân tích hướng giải quyết được dễ dàng hơn. Tư tưởng của lớp câu hỏi này sẽ có 2 dạng câu hỏi chính:

- **Dạng 1: Dạng câu hỏi trích dẫn trực tiếp**

What did A(class) say/said?

Đây là dạng câu hỏi đơn thuần chỉ xem người, hay một Câu lạc bộ nói về một việc đó.

Ví dụ: “What did Lionel Messi say?”

Từ đây em xin được mô hình hóa câu hỏi đơn giản này như sau: (A là thực thể)

Dạng 1: Câu hỏi trích dẫn trực tiếp

```
SELECT DISTINCT ?graph ?x0
WHERE {
  GRAPH ?graph {
    <A> <say> ?x0.
  }
}
```

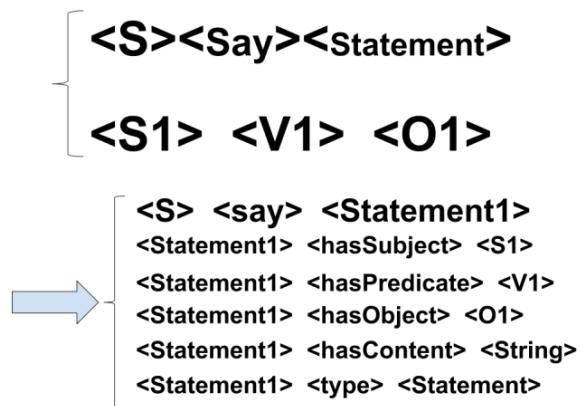
- **Dạng 2: Dạng câu hỏi trích dẫn gián tiếp**

What did A(class) say/said about B(class)?

Ví dụ: “What did Lionel Messi say about B(class)?

Đây là dạng câu hỏi khó hơn, khi như ta đã phân tích, A nói về một nhận định, và khó hơn nữa là nhận định này bao gồm một thực thể, mà thực thể này lại mang một quan hệ, hay thực hiện một quan hệ nào đó.

Trước khi tiếp tục mô hình câu hỏi lớp này, ta sẽ xem xét lại về ngữ nghĩa được tạo ra nhằm mục đích mô hình hóa dạng quan hệ trích dẫn gián tiếp đã được mô tả ở phần 5.2.1 như sau:



Như vậy ta đã có mô hình hóa về dữ liệu, từ đây em đề xuất mô hình hóa câu hỏi dưới dạng sau: (A – B là hai thực thể được nhắc đến trong câu)

Dạng 2: Câu hỏi trích dẫn gián tiếp

```

  SELECT DISTINCT ?graph ?x6
  WHERE {
    GRAPH ?graph {
      <A> <say> ?x0.
      {
        ?x0 <subject> <B>.
        ?x0 <hasContent> ?x6.
      }
    UNION {
      ?x0 <object> <B>.
      ?x0 <hasContent> ?x6.
    }
    ?x0 <type> <Statement>.
  }
  
```

2. Thuật toán xử lý

Để xử lý được lớp câu hỏi này, em chú ý quan tâm đến cây phân tích cú pháp. Ví dụ cho câu hỏi này sẽ có dạng như sau:

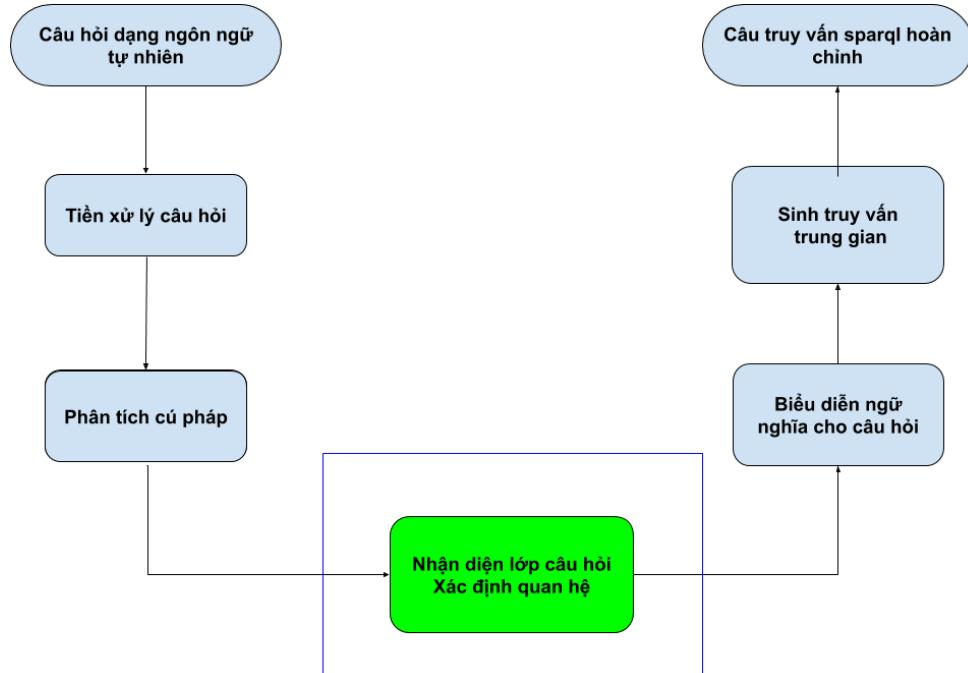
```
(ROOT
  (SBARQ
    (WHNP (WP What))
    (SQ (VBD did)
      (NP (NNP Lionel) (NNP Messi))
      (VP (VB say)
        (PP (IN about)
          (NP (NNP Chelsea)))))))
  (. ?)))
```

```
dobj(say-5, What-1)
aux(say-5, did-2)
nn(Messi-4, Lionel-3)
nsubj(say-5, Messi-4)
root(ROOT-0, say-5)
prep_about(say-5, Chelsea-8)
```

Như vậy ta có thể thấy, điểm khác biệt giữa dạng 1 và dạng 2 là cây cú pháp có xác định nhãn **prep_about**. Và nếu sau **prep_about** này là một thực thể được xác định với nhãn **NP** thì đây là câu hỏi thuộc dạng 2.

Vậy để nhận diện, xác định và sinh câu hỏi thuộc lớp câu hỏi này, em đề xuất thêm một bước trong module Semantic Search là “Nhận diện lớp câu hỏi – Xác định quan hệ”

Mô hình sau khi thêm như sau:



Hình 11: Mô hình hóa thuật toán xử lý lớp câu hỏi

Phần màu xanh da trời, được đánh dấu là phần em tiến hành thêm vào module, giúp nhận dạng tốt hơn các lớp hoặc dạng câu hỏi đặc biệt. Ở các phần sau, khi nhận diện các lớp câu hỏi như Kết hợp hoặc Chọn nura, hoặc câu hỏi về sự kiện, em cũng tiến hành so sánh theo nhãn và đưa ra kết quả ở bước này. Về thuật toán, ở phần sau em xin phép được trích dẫn từ đây.

3. Kết quả đạt được

Từ việc mô hình hóa và xử lý thuật toán, hệ thống đã xử lý được câu hỏi ở dạng này.

Ví dụ câu hỏi sinh ra ở dạng 1: “What did Lionel Messi say?”

```

1. PREFIX owl: <http://www.w3.org/2002/07/owl#>
2. PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3. PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4. PREFIX bksport: <http://bk.sport.owl#>
5. PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
6. PREFIX time: <http://www.w3.org/2006/time#>
7. SELECT DISTINCT ?graph ?x0
8. WHERE {
9.   GRAPH ?graph {
10.      <http://bk.sport.owl#lionel-messi> <http://bk.sport.owl#say> ?x0.
11.    }
12.  }
  
```

Ví dụ câu hỏi sinh ra ở dạng 2: “What did Lionel Messi say about Aston Villa FC?”

```
1. PREFIX owl: <http://www.w3.org/2002/07/owl#>
2. PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3. PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4. PREFIX bksport: <http://bk.sport.owl#>
5. PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
6. PREFIX time: <http://www.w3.org/2006/time#>
7. SELECT DISTINCT ?graph ?x6
8. WHERE {
9.   GRAPH ?graph {
10.     <http://bk.sport.owl#lionel-messi> <http://bk.sport.owl#say> ?x0.
11.   {
12.     ?x0 rdf:subject <http://bk.sport.owl#aston-villa>.
13.     ?x0 rdf:hasContent ?x6.
14.   }
15.   UNION {
16.     ?x0 rdf:object <http://bk.sport.owl#aston-villa>.
17.     ?x0 rdf:hasContent ?x6.
18.   }
19.   }
20.   ?x0 rdf:type rdf:Statement.
21. }
```

4.3.2 Lớp câu hỏi Kết hợp hoặc Chọn lựa

Lớp câu hỏi Kết hợp hoặc Chọn lựa là lớp câu hỏi khó, về mặt ngữ nghĩa nó sẽ suy luận khả năng để đưa ra kết quả. Ở đây em sẽ mô hình và đưa ra thuật toán để giải quyết 2 dạng của câu hỏi này là: Dạng câu hỏi xác định quan hệ và dạng câu hỏi chưa xác định quan hệ.

1. Mô hình hóa và xử lý lớp câu hỏi

Rõ ràng, để kết hợp hay lựa chọn thì trong câu hỏi phải bao gồm ít nhất là từ 2 thực thể và một quan hệ giữa các thực thể đó. Em sẽ chia ra làm các dạng

Dạng 1: Dạng câu hỏi đã xác định rõ quan hệ

Một ví dụ của dạng này là:

“Did Chelsea FC defeat Barcelona FC or Everton FC?” - Dạng lựa chọn

“Did Chelsea FC defeat Barcelona FC and Everton FC?” - Dạng kết hợp

Mô hình hóa dạng này:

A(class) – Predicate – B(class) or/and C (class)

Với A, B, C là các thực thể trong câu. Predicate là quan hệ đã được xác định sẵn, ví dụ như thắng, thua, hòa ...

Dạng 2: Dạng câu hỏi chưa xác định rõ quan hệ

Ví dụ của dạng này: “What happened between Everton FC and Chelsea FC?”

Ở đây, ta chỉ biết về thực thể trong câu hỏi, mà không biết rõ quan hệ gì giữa các thực thể là gì, câu hỏi phải thể hiện được quan hệ đang hỏi là gì? 2 đội này có quan hệ với nhau như thế nào.

Em mô hình hóa dạng này như sau:

What happen between/about A(class) and/or B(class)?

Trong đó, A, B là các thực thể xuất hiện trong câu. Vậy để xử lý các dạng câu hỏi này, ta sẽ xác định liệu tồn tại thực thể trong câu hỏi hay không. Nếu có và số thực thể lớn hơn 2 thì nó là dạng 1; Nếu không tồn tại quan hệ số thực thể lớn hơn hoặc bằng 2 thì nó thuộc dạng 2;

Việc sinh ra câu hỏi ở dạng 1 ta đơn thuần sẽ sinh ra các quan hệ bộ 3 và tùy vào là quan hệ kết hợp hay lựa chọn mà thực hiện sinh truy vấn.

Ở dạng 2 ta sẽ thực hiện việc này bằng một thuộc tính của ngữ nghĩa, đó là

<rdfs:subPropertyOf> . Đây là một thuộc tính thể hiện các quan hệ con thuộc một quan hệ lớn hơn mà ta đã định nghĩa trong ontology.

Ví dụ : <win> <rdfs:subPropertyOf> <happen>

<defeat> <rdfs:subPropertyOf> <happen>

<say><rdfs:subPropertyOf> <happen>

...

Vậy vấn đề về dạng đã được xử lý, giờ ta còn vấn đề về xác định một câu thuộc kiểu Kết hợp hay Lựa chọn. Để xử lý vấn đề này ta lại tiếp tục xem xét cây phân tích cú pháp. Ví dụ về câu hỏi “What happened between Everton FC and Chelsea FC?”

```

ROOT
(SBARQ
 (WHNP (WP What))
 (SQ
  (VP (VBD happened)
   (PP (IN between)
    (NP
     (NP (NNP Everton) (NNP FC))
     (CC and)
     (NP (NNP Chelsea) (NNP FC))))))
  (. ?)))

```

```

nsubj(happened-2, What-1)
root(ROOT-0, happened-2)
nn(FC-5, Everton-4)
prep_between(happened-2, FC-5)
nn(FC-8, Chelsea-7)
prep_between(happened-2, FC-8)
conj_and(FC-5, FC-8)

```

Từ đây ta nhận thấy rằng việc xác định câu hỏi thuộc dạng kết hợp hay lựa chọn phụ thuộc vào nhãn **CC** và nhãn **conj_and** hoặc **conj_or**.

Từ đây qua thuật toán xác định nhãn và quan hệ ở thuật toán đã nêu ở phần 5.3.1. Ta có thể xây dựng dạng câu hỏi như sau:

Dạng 1: Câu hỏi xác định quan hệ

```

SELECT DISTINCT ?graph ?x0
WHERE {
  GRAPH ?graph {
    <A><predicate><B>.
    <A><predicate><C>.
  }
  ?x0 rdfs:subPropertyOf <happen>.
}
<Đạng Kết Hợp>

SELECT DISTINCT ?graph ?x0
WHERE {
  GRAPH ?graph {
    { <A><predicate><B>.}
    UNION
    { <A><predicate><C>.}
  }
}
<Đạng Lựa Chọn>

```

Dạng 2: Câu hỏi chưa xác định quan hệ

```

SELECT DISTINCT ?graph ?x0
WHERE {
  GRAPH ?graph {
    <A> ?x0 <B>.
  }
  ?x0 rdfs:subPropertyOf <happen>.
}

```

2. Kết quả đạt được

Từ mô hình hóa trên, hệ thống đã sinh được câu hỏi dưới dạng này. Sau đây là ví dụ cho các dạng khi hệ thống được sinh ra:

Câu hỏi: “Did Chelsea FC defeat Barcelona FC or Everton FC ?”

```
1. PREFIX owl: <http://www.w3.org/2002/07/owl#>
2. PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3. PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4. PREFIX bksport: <http://bk.sport.owl#>
5. PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
6. PREFIX time: <http://www.w3.org/2006/time#>
7. ASK WHERE {
8.   GRAPH ?graph {
9.     {
10.       <http://bk.sport.owl#chelsea> <http://bk.sport.owl#defeat> <http://bk.sport.owl#barcelona>.
11.     }
12.     UNION {
13.       <http://bk.sport.owl#chelsea> <http://bk.sport.owl#defeat> <http://bk.sport.owl#everton>.
14.     }
15.   }
16. }
```

Câu hỏi: “Did Chelsea FC defeat Barcelona FC and Everton FC ?”

```
1. PREFIX owl: <http://www.w3.org/2002/07/owl#>
2. PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3. PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4. PREFIX bksport: <http://bk.sport.owl#>
5. PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
6. PREFIX time: <http://www.w3.org/2006/time#>
7. ASK WHERE {
8.   GRAPH ?graph {
9.     <http://bk.sport.owl#chelsea> <http://bk.sport.owl#defeat> <http://bk.sport.owl#barcelona>.
10.     <http://bk.sport.owl#chelsea> <http://bk.sport.owl#defeat> <http://bk.sport.owl#everton>.
11.   }
12. }
```

Câu hỏi: “What happened between Everton FC and Chelsea FC?”

```
1. PREFIX owl: <http://www.w3.org/2002/07/owl#>
2. PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3. PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4. PREFIX bksport: <http://bk.sport.owl#>
5. PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
6. PREFIX time: <http://www.w3.org/2006/time#>
7. SELECT DISTINCT ?graph ?x0
8. WHERE {
9.   GRAPH ?graph {
10.     <http://bk.sport.owl#everton> ?x0 <http://bk.sport.owl#chelsea>.
11.   }
12.   ?x0 rdfs:subPropertyOf <http://bk.sport.owl#happen>.
13. }
```

4.3.3 Lớp câu hỏi về một người với một sự kiện và câu hỏi so sánh hơn

Đây là dạng câu hỏi về một sự kiện diễn ra trong bóng đá. Đặc điểm của các sự kiện này là các sự kiện này là lớp con của lớp <EVENT>.

1. Mô hình hóa câu hỏi

Tư tưởng của lớp câu hỏi về sự kiện là trả lời 2 dạng câu hỏi:

Who has/make A (A là subClassOf <Event>)?

Who has/make A (A là subClassOf <Event> in B (class))?

Với A là đối tượng (thường là cầu thủ), B là một thực thể chứa cầu thủ, ví dụ đội bóng, giải đấu...)

Để giải quyết lớp câu hỏi này, ta tiến hành xác định lần lượt A có phải là một class thuộc lớp class Event hay không. Nếu trong câu tồn tại B là một lớp chứa A thì tiếp tục kiểm tra và đưa ra kiểu cho lớp này.

Với lớp câu hỏi về so sánh hơn, em tập trung trả lời dạng câu hỏi:

Who scored more/less than N goals? (N là một số)

Để xử lý lớp câu hỏi này, ta lại tiếp tục xem xét cây cấu trúc:

```
(ROOT
  (SBARQ
    (WHNP (WP Who))
    (SQ
      (VP (VBD scored)
        (NP
          (QP
            (XS (JJR more) (IN than))
            (CD 2))
            (NNS goals))))
        (. ?)))
```



```
nsbj(scored-2, Who-1)
root(ROOT-0, scored-2)
mwe(than-4, more-3)
quantmod(2-5, than-4)
num(goals-6, 2-5)
dobj(scored-2, goals-6)
```

Để nhận diện lớp câu hỏi này, ta tiến hành phân tách nhãn **JJR** và nhãn **mwe**. Từ đây với thuật toán gán nhãn và xác định quan hệ từ thuật toán đã mô tả ở mục 5.3.1, ta sẽ sinh ra câu hỏi tương ứng.

2. Kết quả đạt được

Từ bước mô hình và các thuật toán liên quan, hệ thống đã có thể sinh ra các câu hỏi thuộc lớp này.

Ví dụ về câu hỏi về sự kiện: “who make hat trick in Chelsea FC?”

```
1. PREFIX owl: <http://www.w3.org/2002/07/owl#>
2. PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3. PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4. PREFIX bksport: <http://bk.sport.owl#>
5. PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
6. PREFIX time: <http://www.w3.org/2006/time#>
7. SELECT DISTINCT ?graph ?x7
8. WHERE {
9.   GRAPH ?graph {
10.     ?x7 bksport:makeHatTrick ?x9.
11.   }
12.   ?x9 rdf:type bksport:HatTrick .
13.   ?x7 bksport:playFor <http://bk.sport.owl#chelsea>.
14.   ?x7 rdf:type <http://bk.sport.owl#Player>.
15. }
```

Ví dụ về câu hỏi more than: “Who scored more than 2 goals?”

```
1. PREFIX owl: <http://www.w3.org/2002/07/owl#>
2. PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
3. PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
4. PREFIX bksport: <http://bk.sport.owl#>
5. PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
6. PREFIX time: <http://www.w3.org/2006/time#>
7. SELECT DISTINCT ?graph ?x0
8. WHERE {
9.   GRAPH ?graph {
10.     ?x0 <http://bk.sport.owl#hasNumberOfGoals> ?x4.
11.   }
12.   ?x4 rdf:type xsd:integer.
13.   FILTER ((?x4) > 2).
14. }
```

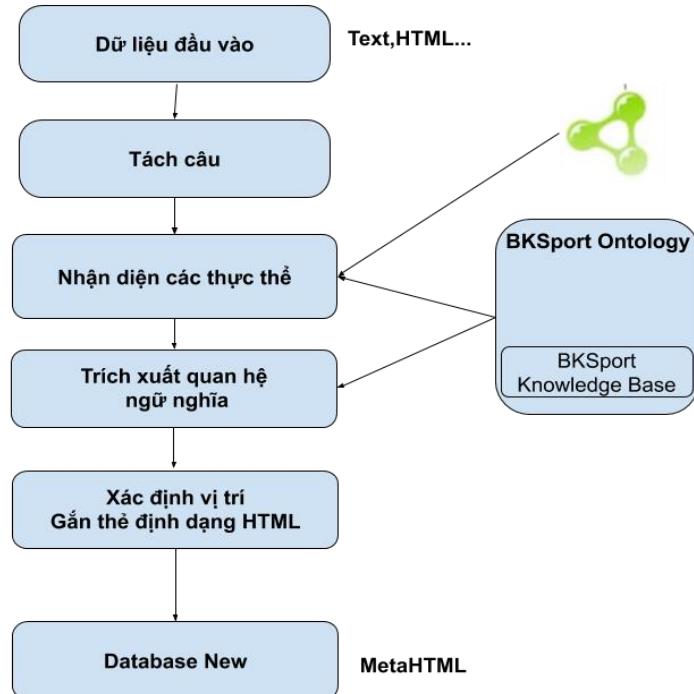
4.4 Trực quan hóa hiển thị dữ liệu ngữ nghĩa

Trước đây, hệ thống chỉ có khả năng hiển thị thông tin dưới dạng text với chức năng xem tin thông thường. Với mong muốn trực quan hóa thông tin cho người dùng có thể theo dõi, xem và tập trung vào các tin tức quan trọng, trong phạm vi đồ án này, em đã xây dựng tính năng hiển thị dữ liệu ngữ nghĩa với mục tiêu tập trung trực quan hóa các thông tin sau:

- Các cầu thủ xuất hiện trong bài báo
- Các huấn luyện viên xuất hiện trong bài báo

- Các đội bóng xuất hiện trong bài báo
- Các giải đấu trong bài báo
- Các quan hệ ngữ nghĩa trong bài báo.

1. Xây dựng thuật toán



Hình 12: Mô hình hóa thuật toán gắn nhãn cho dữ liệu ngữ nghĩa

Khó khăn lớn nhất khi thực hiện xây dựng tính năng này là khi nhận diện quan hệ, sinh chú thích ngữ nghĩa, ta có các vị trí của các thực thể, nhưng để hiển thị các thực thể này dưới dạng HTML, được gắn nhãn thì vị trí của các thực thể này đã thay đổi. Vậy nên, cần một thuật toán để tính toán vị trí offset và tổng ký tự trong thẻ từ đó đưa ra vị trí tuyệt đối cho từng thực thể.

Trong việc trực quan hóa dữ liệu ngữ nghĩa, quan trọng nhất là gắn nhãn ứng với mỗi thực thể là các kiểu, đồng thời chú thích cho các thực thể đó nó thuộc dạng nào, có ngữ nghĩa ra sao. Để thực hiện việc này em đã tiến hành như sau:

- Input: File HTML của một bài báo
- Output: File HTML đã được đánh dấu mà em gọi là MetaHTML
- Bước 1: Tiến hành tách lần lượt các câu trong file văn bản HTML

- Bước 2: Sử dụng BKSport Ontology để nhận diện thực thể có trong văn bản. Lưu lại các thông tin quan trọng bao gồm:
 - Vị trí offset bắt đầu và kết thúc của các thực thể này
 - Kiểu của thực thể
 - Lớp mà thực thể này thuộc về
- Bước 3: Sử dụng BKSport Ontology để trích xuất quan hệ ngữ nghĩa có trong văn bản. Lưu lại các thông tin quan trọng bao gồm:
 - Vị trí offset bắt đầu và kết thúc
 - Quan hệ trích xuất được dưới dạng bộ 3
- Bước 4: Xác định vị trí và gán thẻ HTML

Đây là bước phức tạp và quan trọng nhất của thuật toán. Ta tiến hành các bước bao gồm:

- Xử lý nhiễu: Loại bỏ các thực thể có trong các đường link mà bài báo đề cập(vì các thực thể này không có giá trị trong bài báo)
- Tính toán lại vị trí của offset của thực thể và quan hệ
Nếu ta chỉ đơn thuần thêm các tag gắn vào để làm nổi bật thông tin ngữ nghĩa, thì vị trí của mỗi offset sẽ thay đổi sau mỗi lần thêm vào dẫn đến sai sót với các thực thể thêm ở sau. Vậy nên ta cần tính toán các loại tag, vị trí thay đổi, và trị trí mới sau khi thay đổi của mỗi thực thể.
- Gán thẻ cho mỗi loại thực thể và quan hệ:
Để nhằm nổi bật kết quả đạt được, em tiến hành gán tag như sau:
Thực thể: Với mỗi lớp thực thể, sẽ highlight với màu khác nhau, khi di chuột vào sẽ hiện lên lớp mà thực thể đó thuộc về
Với quan hệ: Với đoạn văn chứa quan hệ sẽ được gạch chân, khi di chuột vào sẽ hiện thị quan hệ dưới dạng bộ ba ngữ nghĩa
- Bước 5: Xuất dữ liệu ngữ nghĩa dưới dạng HTML đã được đánh tag (Meta HTML)

2. Kết quả đạt được

Sau thuật toán được thực hiện, khi có lựa chọn hiển thị trực quan ngữ nghĩa, hệ thống sẽ làm nổi bật các quan hệ và ngữ nghĩa trích xuất được trong bài báo.

Ví dụ về một mẫu thực thể và offset:

Tên	Text xuất hiện trong văn bản	Kiểu – Lớp	Offset	Offset
			Begin	End
Dwight Gayle	Dwight Gayle	Thực thể - Goalkeeper	144	156
crystal-palace defeat stoke-city	Crystal Palace beat Stoke 2-1	Quan hệ	3	32
Crystal Palace	Palace	Thực thể - Club Team	339	345

Sau khi chạy thuật toán, ta có đoạn html được gắn tag như sau:

Offset	HTML
3	<u style="text-decoration: underline; text-decoration-color: red;" data-toggle="tooltip" data-placement="top" title="crystal-palace defeat stoke-city">
89	</u>
179	<mark style="background-color: rgba(0, 151, 19, 0.25); color: black">
289	</mark>
379	<mark style="background-color: rgba(0, 255, 127, 0.25); color: black">
467	</mark>

Chương 5 Kết quả thực nghiệm và Đánh giá

Trong chương này, em sẽ tiến hành trình bày các nội dung bao gồm: Kết quả thực nghiệm khi so sánh giữa hệ thống trước và sau khi được phát triển, tối ưu.

Sau khi thực hiện cài đặt phương pháp nâng cao độ chính xác của module sinh chú thích ngữ nghĩa và module sinh câu hỏi em đã tiến hành so sánh với kết quả đã đạt được trước đó của hệ thống.

Tiếp đến em mô tả tính năng mới của module web sau khi được xây dựng.

5.1 Kết quả thực nghiệm về khả năng sinh chú thích ngữ nghĩa

1. Mẫu thực nghiệm

Để tiến hành thực hiện thực nghiệm về khả năng sinh chú thích ngữ nghĩa, em xây dựng bộ dataset bằng cách crawl dữ liệu từ các trang tin thể thao với nguồn là các trang tin bằng tiếng anh như: www.espnfcasia.com, <https://www.skysports.com>, <https://www.goal.com>...

Tiếp đó em sinh chú thích ngữ nghĩa bằng con người trên các tin tức này, lưu lại các mẫu tin và các chú thích để tiến hành thực nghiệm.

Để đánh giá sơ bộ kết quả cải tiến, em tiến hành thực nghiệm trên tập gồm 150 tin.

2. Kịch bản thực nghiệm

Tiến hành chạy mẫu thực nghiệm là bộ dataset ở module Semantic Annotation ở hệ thống mới và cũ để sinh ra chú thích ngữ nghĩa.

Tiến hành kiểm tra chú thích ngữ nghĩa bằng con người dựa trên các tiêu chí:

- Kiểm tra về số lượng các chú thích ngữ nghĩa được sinh ra ở hệ thống mới và cũ
- Kiểm tra về chất lượng các chú thích ngữ nghĩa:

➤ Tỷ lệ sinh đúng chú thích ngữ nghĩa: $\frac{\text{Số ngữ nghĩa đúng}}{\text{Tổng số ngữ nghĩa được sinh ra}} \%$

➤ Tỷ lệ ngữ nghĩa bao phủ: $\frac{\text{Số ngữ nghĩa đúng}}{\text{Tổng số ngữ nghĩa kỳ vọng}} \%$

3. Kết quả thử nghiệm và đánh giá

Sau khi chạy chương trình thực nghiệm, kết quả chương trình được biểu diễn ở bảng dưới đây

Thực nghiệm bằng con người	Hệ Thông Cũ			Hệ Thông Mới		
Tổng số chủ thích ngữ nghĩa	Tổng số chủ thích ngữ nghĩa	Số ngữ nghĩa sinh đúng	Tỷ lệ ngữ nghĩa bao phủ	Tổng số chủ thích ngữ nghĩa	Số ngữ nghĩa sinh đúng	Tỷ lệ ngữ nghĩa bao phủ
2587	2112	1571 (74,38%)	60,72%	2368	1906 (80,49%)	73,67%

Kết thực nghiệm cho thấy, tính riêng trong 100 bài báo, với tổng số ngữ nghĩa kỳ vọng là 2587 ngữ nghĩa, thì ở hệ thống cũ nhận diện được 2112 ngữ nghĩa, ở hệ thống mới số ngữ nghĩa nhận được là 2368 ngữ nghĩa, tăng 12,13% so với hệ thống cũ.

Tỷ lệ sinh ngữ nghĩa chính xác tính theo công thức 5.1.2 thì hệ thống cũ là 74,38% đã tăng lên 80,49%, tăng 6,11%.

Tỷ lệ ngữ nghĩa bao phủ ở hệ thống cũ là 60,72% đã tăng lên 73,67%, tăng 12,95%.

Kết quả so sánh với hệ thống cũ thì hệ thống mới đã tăng lên về cả số lượng ngữ nghĩa lẫn chất lượng ngữ nghĩa sinh đúng.

Kết quả đạt được như vậy không chỉ bởi đã mở rộng số quan hệ lên, mà do cơ sở tri thức đã được mở rộng làm tăng cả ba thông số là số chủ thích ngữ nghĩa, tỷ lệ ngữ nghĩa sinh đúng, tỷ lệ ngữ nghĩa bao phủ.

5.2 Kết quả thực nghiệm về khả năng sinh câu hỏi và khả năng trả lời câu hỏi

1. Mẫu thực nghiệm

Để thực nghiệm về khả năng sinh câu hỏi và khả năng trả lời câu hỏi, em tiến hành lựa chọn tập câu hỏi gồm 30 câu hỏi về lĩnh vực bóng đá. Trong bộ câu hỏi này, có các câu hỏi nằm trong bộ câu hỏi đánh giá về khả năng sinh câu hỏi trước đây của hệ thống được thực hiện bởi các thành viên trước đây.

Với mỗi câu hỏi em tiến hành đánh giá dựa trên 2 tiêu chí: Sinh đúng câu hỏi và hệ thống trả lời được câu hỏi hay chưa. Nếu hệ thống đã sinh đúng câu hỏi, em cho giá trị là 1, nếu chưa sinh được em cho giá trị là 0. Tương tự với khả năng trả lời được câu hỏi, nếu hệ thống trả lời được, em cho giá trị là 1, nếu chưa em cho giá trị là không. Từ đây, em tiến hành so sánh hiệu quả của hệ thống mới và cũ.

2. Kết quả đạt được

Kết quả thực nghiệm được trình bày ở bảng sau:

STT	Câu hỏi	Hệ thống cũ		Hệ thống mới	
		Sinh đúng câu hỏi	hệ thống trả lời được câu hỏi	Sinh đúng câu hỏi	hệ thống trả lời được câu hỏi
1	Who is Lionel Messi?	1	0	1	0
2	Did Ronaldo play for Real Madrid?	1	1	1	1
3	Which team Lionel Messi has played ?	0	0	1	1
4	Which team defeated Chelsea?	1	1	1	1
5	What did Ronaldo say about Barcelona?	1	0	1	1
6	did Barcelona defeat Chelsea FC?	1	1	1	1
7	News about Chelsea?	1	1	1	1
8	Which news contains Lionel Messi?	1	1	1	1
9	Which event relates to Lionel Messi?	1	1	1	1
10	What is the result of the match between Chelsea and Barcelona?	0	0	1	1
11	Which team defeated Real Madrid and Manchester City?	0	0	1	1

12	did Manchester United and Chelsea defeat Barcelona?	0	0	1	1
13	Who scored 3 goals?	0	0	1	1
14	Which player will leave Chelsea?	1	1	1	1
15	What happened between Chelsea and Barcelona?	0	0	1	1
16	Which football team plays its home games at Wembley?	0	0	0	0
17	did chelsea FC defeat Barcelona FC or Everton FC ?	1	1	1	1
18	Which player holds the record for the most goals in consecutive Premier League games?	0	0	0	0
19	Who scored more than 2 goals?	0	0	1	1
20	Who has hat trick in chelsea FC?	0	0	1	1
21	Is Chelsea bigger than barcelona ?	0	0	0	0
22	Who has managed Reading, Swansea and Liverpool?	1	1	1	1
23	Where was Lionel Messi born?	0	0	0	0
26	How many teams in the Premier League?	1	0	1	0
27	Which team in England oldest?	0	0	0	0
28	What did Lionel Messi say ?	1	1	1	1
29	give me all player of Barcelona?	1	1	1	1
30	Who make hat trick ?	0	0	1	1
31	Tổng	14	11	23	21

Bảng 11: Thực nghiệm kết quả sinh câu hỏi từ ngôn ngữ tự nhiên sang ngôn ngữ truy vấn SPARQL và khả năng trả lời câu hỏi của hệ thống

Từ kết quả trên cho ta thấy: số lượng câu hỏi sinh được từ 14 ở hệ thống cũ là tăng lên 23 tương đương 30% ở hệ thống mới. Số lượng câu hỏi trả lời được cũng tăng từ 11 lên 21 tương đương 33,3 % ở hệ thống mới.

Điều này đạt được do hệ thống sau khi được phát triển đã mở rộng thêm các lớp câu hỏi, đồng thời dữ liệu do module sinh ngữ nghĩa được mở rộng, góp phần tăng số lượng câu hỏi trả lời được.

5.3 Kết quả xây dựng trang web tổng hợp tin tức BKSport phiên bản 2.0 (Module Web giao diện người dùng)

Một trong những nhiệm vụ khi thực hiện đồ án của em là tiến hành xây dựng giao diện cho người dùng với các chức năng của hệ thống BKSport. Với mục đích nêu trên, em đã tiến hành xây dựng module Web giao diện người dùng với tên gọi Trang Web tổng hợp tin tức thể thao BKSport phiên bản 2.0.

Trang web đã được xâ dựng và triển khai ở môi trường localhost và được deploy public trên internet với địa chỉ truy cập là: <http://103.27.237.85:3000/>

Trong mục này, em xin được thực nghiệm kết quả ở 2 chức năng ứng với 2 module Sinh chủ thích ngữ nghĩa và sinh câu hỏi truy vấn ngữ nghĩa và tiếp đó là chức năng trực quan hóa ngữ nghĩa cho người dùng.

5.3.1 Kết quả về khả năng tìm kiếm ngữ nghĩa.

Với chức năng tìm kiếm ngữ nghĩa, hệ thống đã trả lời được 3 dạng câu hỏi:

1. Dạng câu hỏi về Cơ sở thi thực

Tư tưởng của dạng câu hỏi này là các câu hỏi liên quan đến thông tin của một cá nhân hoặc tập thể nào đó. Đây là các thông tin không có trong nội dung của một bài báo hay trang tin nào mà là các thông tin về cá nhân hoặc tập thể nào đó trong cơ sở thi thực để trả lời cho người dùng.

Ví dụ câu hỏi được đưa ra là: “All player of Chelsea FC?”

Theo vào đó hệ thống sẽ tìm ra các thực thể có kiểu là Football Player, và hiện tại đang chơi cho Chelsea FC. Dữ liệu này đã được mô tả trong Cơ sở dữ liệu, hệ thống sẽ sinh câu hỏi và truy vấn thông tin đó lên server và hiển thị cho người dùng đầy đủ các thông tin bao gồm: Số lượng câu trả lời phù hợp và chi tiết các câu trả lời phù hợp.

Trong câu hỏi này ta có 53 cầu thủ đang chơi cho Chelsea, và chi tiết tên của các cầu thủ ở phía dưới được hệ thống hiển thị rất trực quan.

The screenshot shows the BKSPORT - SPORT NEWS website. At the top, the logo reads "BKSPORT - SPORT NEWS" with the subtitle "AGGREGATION USING SEMANTIC WEB TECHNOLOGY". Below the logo is a search bar with the placeholder "WHAT ARE YOU LOOKING FOR?". To the right of the search bar is a "Search" button. A line of text "all player of Chelsea FC?" is entered into the search bar. Below the search bar, a section titled "Kết quả các đối tượng cần tìm" (Search results for objects) displays a list of names: Ross Turnbull, Marin, Oriol Romeu, Raul Meireles, Jose Bosingwa, and Falcao. A bracket on the left side of the list is labeled "Danh sách các câu trả lời phù hợp" (List of suitable answers). Another bracket on the left side of the search bar is labeled "Số lượng câu trả lời phù hợp trả về" (Number of suitable answers returned).

Một ví dụ khác cho câu hỏi này: “Who is Lionel Messi?”

The screenshot shows the BKSPORT - SPORT NEWS website. At the top, the logo reads "BKSPORT - SPORT NEWS" with the subtitle "AGGREGATION USING SEMANTIC WEB TECHNOLOGY". Below the logo is a search bar with the placeholder "WHAT ARE YOU LOOKING FOR?". To the right of the search bar is a "Search" button. A line of text "Who is Lionel Messi?" is entered into the search bar. Below the search bar, a section titled "Các bài viết liên quan: 1" (Related articles: 1) displays a list of related articles. A bracket on the left side of the search bar is labeled "Câu hỏi dạng ngôn ngữ tự nhiên" (Natural language question). A bracket on the left side of the search bar is labeled "Kết quả chi tiết" (Detailed results).

Câu hỏi hỏi về Lionel Messi, hệ thống sẽ trả về chi tiết thông tin của cầu thủ Messi trong cơ sở ngữ nghĩa và hiển thị cho người dùng.

2. *Dạng câu hỏi đúng/sai*

Đặc điểm câu hỏi này không phải là danh sách câu trả lời về bài báo hay về cơ sở thi thực. Mà câu trả lời là một nhận định đúng hoặc sai. Hệ thống sẽ truy vấn vào các ngữ nghĩa được sinh ra trong các bài báo, nếu có thì đưa ra kết quả là True. Nếu không tồn tại ngữ nghĩa nào liên quan thì trả về kết quả là False.

Ví dụ câu hỏi: ‘Did Everton FC defeat Chelsea FC?’



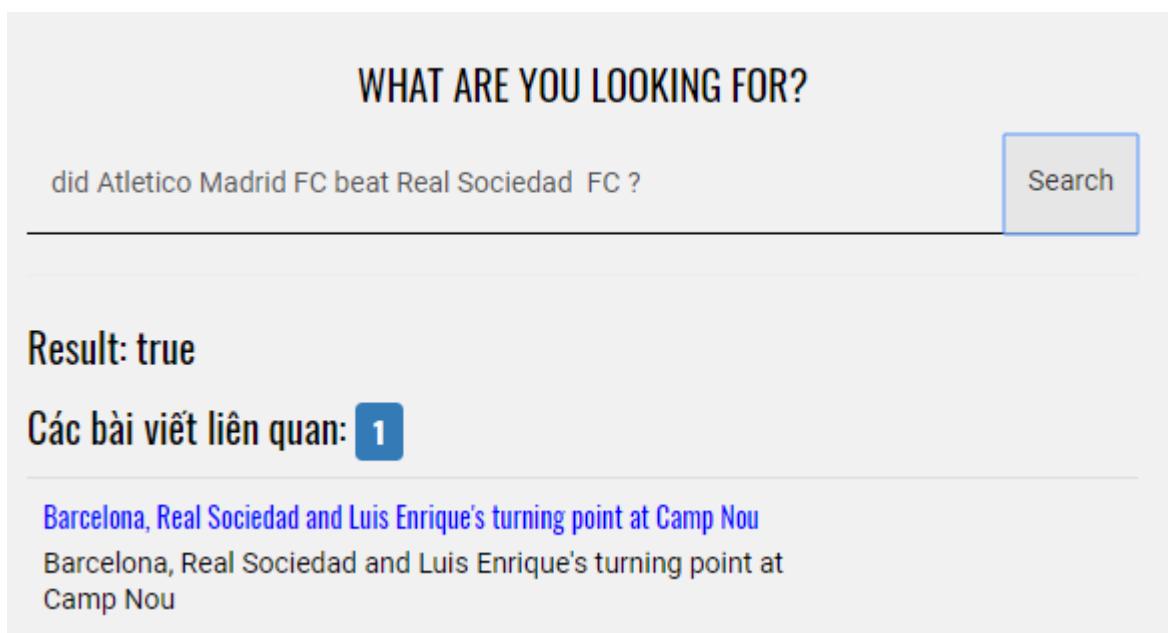
WHAT ARE YOU LOOKING FOR?

did Everton FC beat Chelsea FC? Search

Result: false

Đây là mẫu trả lời cho kết quả là False.

Một ví dụ khác: “Did Atletico Madrid FC beat Real Sociedad FC ?”



WHAT ARE YOU LOOKING FOR?

did Atletico Madrid FC beat Real Sociedad FC ? Search

Result: true

Các bài viết liên quan: 1

[Barcelona, Real Sociedad and Luis Enrique's turning point at Camp Nou](#)
Barcelona, Real Sociedad and Luis Enrique's turning point at Camp Nou

Ở đây, câu hỏi chỉ hỏi “beat”, hệ thống sẽ suy diễn ra mô tả của hệ thống là “defeat” rồi truy vấn cho hệ thống. Ở đây, với câu hỏi này, câu trả lời là True. Hệ thống không chỉ đưa ra câu

trả lời là True – Mà còn đưa ra bài viết có nội dung đó. Khi click vào bài viết – Với lựa chọn xem MetaHTML, ta có thể thấy rõ nội dung được nhắc đến là hoàn toàn chính xác như sau:

 starting lineup, with Messi on the right and Luis Suarez through the middle. That team followed the Real Sociedad loss with a performance in their next league fixture that couldn't have been more of atletico-madrid defeat real-sociedad a contrast, a huge 3-1 win over Atletico Madrid, from which came the now iconic image of Messi, Luis Suarez and Neymar, arm in arm, after each had scored. Barca beat Atleti two more times in the following two weeks, and went on to steamroll all in their path en route

Tuy nhiên điểm yếu của chức năng ở dạng câu hỏi này là phụ thuộc rất nhiều vào dữ liệu. Nếu dữ liệu không bao quát thì câu trả lời khó chính xác.

3. *Dạng câu hỏi đầy đủ*

Đây là dạng câu hỏi phổ biến hơn cả, câu hỏi sẽ đề cập đến một quan hệ hoặc thông tin về một thực thể mà người đó quan tâm. Hệ thống sẽ truy vấn ngữ nghĩa và trả về kết quả là bài báo chứa nội dung đó và kết quả người đó quan tâm.

Ví dụ câu hỏi: “Give me result of the match between aston villa and Bournemouth?”

WHAT ARE YOU LOOKING FOR?

Give me result of the match between aston villa and bournemouth

Search

Các bài viết liên quan: 2

[Leicester and Spurs' title battle continues while Villa could finally go](#) 0:2

[Leicester and Spurs' title battle continues while Villa could finally go](#)

[Leicester and Spurs' title battle continues while Villa could finally go](#) 1:1

[Leicester and Spurs' title battle continues while Villa could finally go](#)

Các nội dung kết quả tìm thấy:

0:2

1:1

Hệ thống sẽ truy vấn và đưa ra kết quả cho kết quả cần truy vấn và cả bài viết mà có kết quả phù hợp:

Kết quả phù hợp
với truy vấn đề ra

Prediction: Aston Villa 1-1 Bournemouth – Kevin Hughes



BOURNEMOUTH: The Cherries will want revenge for a 1-0 home defeat to [Aston Villa](#) in their first ever [Premier League](#) game last August. After two chastening defeats, at [Tottenham](#) and then most recently at home to [Manchester City](#), Eddie Howe's team desperately need to get back on track. Expect improved performances all round – and Villa's relegation confirmed.

Prediction: Aston Villa 0-2 Bournemouth – Steve Menary



LIVERPOOL: The Reds will be boosted by a strong performance and decent result against [Borussia Dortmund](#) in the [Europa League](#) and [Stoke](#) don't have too good a league record at [Anfield](#). But this one looks a stalemate and most likely a game with very little excitement, with the home side possibly distracted by their exploits in Europe. It will be

Một ví dụ khác, thể hiện tính ưu việt của hệ thống:

Ta có ví dụ sau: “What happened between Everton FC and Chelsea FC?”

WHAT ARE YOU LOOKING FOR?

What happened between Everton FC and Chelsea FC? Search

Kết quả các đối tượng cần tìm 1

defeat

Show more

Các bài viết liên quan: 1

Judge Everton at end of the season, says Roberto Martinez defeat

Everton boss Roberto Martinez has reiterated that the club's season should only be assessed once it has come to an end.

Đây là một câu hỏi đặc biệt, hệ thống phải suy diễn ngữ nghĩa, và hiểu defeat là một quan hệ, và người dùng muốn tìm quan hệ này. Hệ thống tìm được quan hệ và đưa ra bài viết có chứa nội dung đó cho người dùng. Suy diễn ngữ nghĩa cũng là điểm mạnh và khác biệt so với các hệ thống tìm kiếm khác.

5.3.2 Kết quả về tính năng trực quan hóa dữ liệu ngữ nghĩa.

Sau khi tiến hành gắn tag nhằm mục đích trực quan hóa dữ liệu ngữ nghĩa, hệ thống đã có khả năng làm cho người dùng xem xét dữ liệu ngữ nghĩa một cách dễ dàng.

Sự khác biệt được thể hiện qua hình ảnh như sau:

Everton have lost three of their last four home matches but Martinez believes his side will rise to the occasion on Saturday with new investor Farhad Moshiri watching from the stands.

"The squad has always reacted bravely and positively," said the Spaniard. "We have already been to the semi-finals of the League Cup, which I think gives us the inspiration to be able to perform well at this stage of the competition."

Everton have already recorded a home victory against a Chelsea side who struggled early on in the campaign under Jose Mourinho.

However, interim manager Guus Hiddink, who led Chelsea to an FA Cup final triumph over Everton in 2009, has helped revive the club's fortunes, leading them on an 11-game unbeaten run which ended when PSG dumped them out of the Champions League on Wednesday.

Hệ thống bình thường

Everton have lost three of their last four home matches but Martinez believes his side will rise to the occasion on Saturday with new investor Farhad Moshiri watching from the stands.

"The squad has always reacted bravely and positively," said the Spaniard. "We have already been to the semi-finals of the League Cup, which I think gives us the inspiration to be able to perform well at this stage of the competition."

Everton have already recorded a home victory against a Chelsea side who struggled early on in the campaign under Jose Mourinho.

However, interim manager Guus Hiddink, who led Chelsea to an FA Cup final triumph over Everton in 2009, has helped revive the club's fortunes, leading them on an 11-game unbeaten run which ended when PSG dumped them out of the Champions League on Wednesday.

Hệ thống dữ liệu đã được trực quan hóa

Ta thấy rõ sự phân biệt rõ ràng về khả năng làm nổi bật thông tin. Từ đây ta có thể thấy rõ được tính ưu việt của hệ thống khi làm nổi bật dữ liệu quan trọng. Với khả năng trực quan hóa dữ liệu ngữ nghĩa, các thực thể bao gồm các đội bóng, cầu thủ, huấn luyện viên... được nổi bật để ta dễ nhận diện; Các câu văn có chứa các quan hệ được gạch chân và khi di chuột vào có thể xem được các nội dung quan hệ đó.

Ví dụ về một tin tức khi chưa được trực quan hóa:

Dwight Gayle stars in Crystal Palace victory over Stoke

Crystal Palace beat Stoke 2-1 in their final home game of the season to finally secure mathematical survival from Premier League relegation. Dwight Gayle scored twice to help the Eagles come from behind, and while the result was just what everyone at the club wanted, it also left a few questions to be answered.

Namely, how does Palace boss Alan Pardew get the best out of Gayle? The impish forward was superb against the Potters – pushed all the way by James McArthur for the Man of the Match – and both goals were brilliantly taken. But all too many times this season, Eagles fans have seen flashes of what Gayle can offer without it ever being backed up with goals.

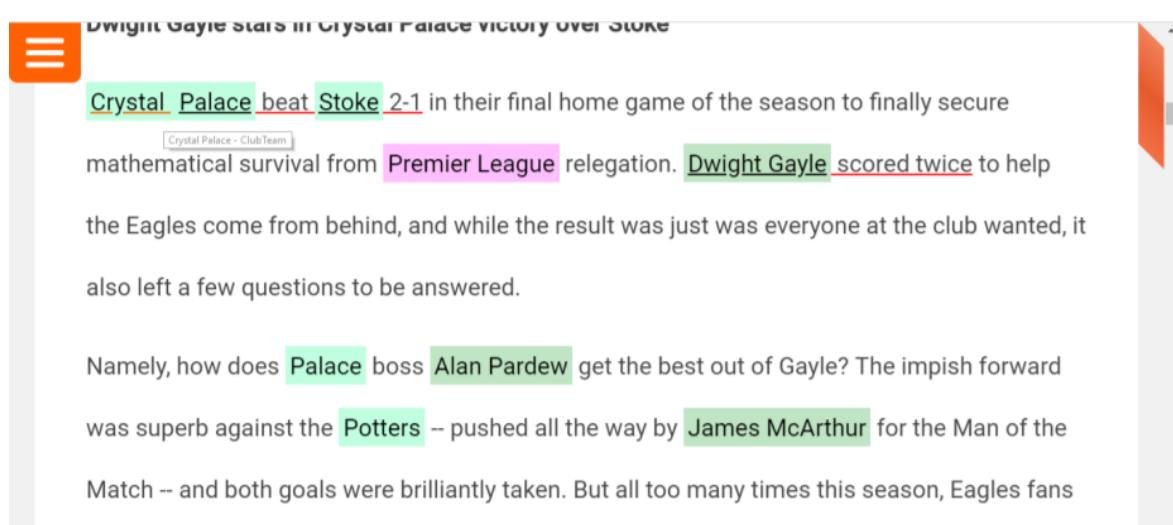
Some of that is down to Pardew restricting the striker's gametime and some of that is down to the system he chooses, but against Stoke Gayle was paired with Connor Wickham in a more rigid 4-4-2 and it worked very well indeed.

Which leads nicely onto question two; should Pardew consider switching up his formation more often? Rarely does he sway from his preferred 4-2-3-1, but the reverting back to four in midfield and two up front to accommodate Gayle was the right choice against Stoke.

It was something of a game of two halves; a dour and tight first half in which Mark Hughes' team somehow took the lead was followed by an open second half where Palace scored twice and Stoke could have also netted again. Palace's chances came from a combination of a more confident approach and a complete abandonment of any defensive discipline from the Potters.

Hình 13: Ví dụ về trang web xem tin thông thường

Ví dụ về dữ liệu đã được trực quan hóa:



Dwight Gayle stars in Crystal Palace victory over Stoke

Crystal Palace beat Stoke 2-1 in their final home game of the season to finally secure mathematical survival from Premier League relegation. Dwight Gayle scored twice to help the Eagles come from behind, and while the result was just what everyone at the club wanted, it also left a few questions to be answered.

Namely, how does Palace boss Alan Pardew get the best out of Gayle? The impish forward was superb against the Potters – pushed all the way by James McArthur for the Man of the Match – and both goals were brilliantly taken. But all too many times this season, Eagles fans

Hình 14: Ví dụ về trang web sau khi được trực quan hóa thực thể

Hình ảnh về trực quan hóa hiển thị các thực thể như hình 14 đã cho ta thấy rõ được các thực thể tồn tại trong câu, bao gồm các cầu thủ, huấn luyện viên được nêu bật trong dữ liệu khác. Để xem xét họ là ai, vai trò là gì, ta có thể di chuột vào từng chỗ được làm nổi bật sẽ có thông tin hiển thị chi tiết hơn.



Dwight Gayle stars in Crystal Palace victory over Stoke

Crystal Palace beat Stoke 2-1 in their final home game of the season to finally secure mathematical survival from Premier League relegation. Dwight Gayle scored twice to help the Eagles come from behind, and while the result was just what everyone at the club wanted, it also left a few questions to be answered.

Namely, how does Palace boss Alan Pardew get the best out of Gayle? The impish forward was superb against the Potters -- pushed all the way by James McArthur for the Man of the Match -- and both goals were brilliantly taken. But all too many times this season, Eagles fans have seen flashes of what Gayle can offer without it ever being backed up with goals.

Hình 15: Ví dụ về trực quan hóa quan hệ bộ ba ngữ nghĩa

Hình ảnh dữ liệu trực quan hóa về quan hệ bộ ba ta có thể thấy rõ được quan hệ này bắt đầu từ vị trí nào, được tác giả bài viết thể hiện qua ngôn từ gì, sẽ được gạch chân. Để xem xét đó là quan hệ nào, ta di chuột để xem quan hệ đó một cách ngắn gọn theo dạng quan hệ bộ ba. Từ đây người đọc dễ dàng nắm bắt, tổng hợp thông tin, hay nói cách khác, dữ liệu ngữ nghĩa đã được trực quan hóa.

Chương 6 Kết luận và hướng phát triển

Trong phạm vi đồ án tốt nghiệp, em đã đề xuất các phương pháp cải thiện và mở rộng khả năng nhận diện ngữ nghĩa của module sinh chủ thích ngữ nghĩa, module sinh câu hỏi từ ngôn ngữ tự nhiên, phát triển trực quan hóa dữ liệu ngữ nghĩa. Dưới đây là các công việc em đã thực hiện trong đồ án:

- Tìm hiểu về web ngữ nghĩa, các lý thuyết về các thành phần liên quan như ontology, ngôn ngữ RDF, ngôn ngữ JAPE, ngôn ngữ SPARQL,... và các công nghệ như KIM Platform, Allegrograph, Ruby on rails...
- Nghiên cứu, phân tích hệ thống tổng hợp thông tin BKSport.
- Tiến hành làm giàu kho cơ sở thi thức trong lĩnh vực bóng đá.
- Tiến hành tăng khả năng sinh chủ thích ngữ nghĩa trong module semantic annotation bằng cách tăng thêm các quan hệ nhận diện được bao gồm: quan hệ trích dẫn gián tiếp, quan hệ kết quả trận đấu và sinh ngữ nghĩa đối lập và quan hệ là biến thể S – V – O.
- Tiến hành tăng khả năng sinh ra câu hỏi mới bao gồm các lớp câu hỏi như: lớp câu hỏi về trích dẫn, lớp câu hỏi về kết hợp hoặc chọn lựa, lớp câu hỏi về một sự kiện và so sánh hơn.
- Tiến hành gắn thẻ, trực quan hóa dữ liệu ngữ nghĩa để hiển thị cho người dùng.
- Tiến hành thực nghiệm so sánh hiệu quả của hệ thống sau khi cải thiện với chính hệ thống trước đó.

Từ những cải tiến về sinh thêm ngữ nghĩa và sinh thêm các lớp câu hỏi cho hệ thống, nay hệ thống BKSport đã có thể nhận diện được thêm nhiều lớp ngữ nghĩa, tăng khả năng sinh thêm các lớp câu hỏi cho người dùng có thể truy vấn. Đồng thời với việc trực quan hóa dữ liệu ngữ nghĩa, nay người dùng có thể theo dõi tin tức một cách đơn giản và nhanh chóng hơn.

Mặc dù đã cải thiện hệ thống, giúp hệ thống có thể trích xuất ngữ nghĩa cũng như đáp ứng nhu cầu của người dùng nhiều hơn, tuy nhiên, hệ thống vẫn còn nhiều điểm cần khắc phục trong tương lai như: Kho cơ sở tri thức cần phải cập nhật thường xuyên, còn nhiều quan hệ mà hệ thống cần thêm vào để nhận diện, cũng như còn nhiều lớp câu hỏi phải bóc tách thủ công.

Trong những nghiên cứu sau, em sẽ tập trung giải quyết những vấn đề trên như: Xây dựng bộ cập nhật cơ sở tri thức tự động, dùng học máy để tăng khả năng tự nhận diện các quan hệ trong việc sinh ngữ nghĩa, cũng như tăng các lớp câu hỏi một cách tự nhiên. Tiến hành xây dựng hệ thống BKSport theo hướng dễ dàng cài đặt và thống nhất.

Tài liệu tham khảo

- [1] https://en.wikipedia.org/wiki/Semantic_Web lần cuối truy cập 13/05/2019
- [2] **Gruber, Thomas R.** (tháng 6 năm 1993). “A translation approach to portable ontology specifications” (PDF). *Knowledge Acquisition* 5 (2): 199–220. **Peterson L. L. and Davie B. S., Computer Networks: A Systems Approach**, 2nd ed., Morgan-Kaufmann, 1999.
- [3] **Arvidsson, F.; Flycht-Eriksson, A.** “Ontologies I” (PDF). Truy cập ngày 26 tháng 11 năm 2008.
- [4] <https://www.ontotext.com/knowledgehub/fundamentals/semantic-annotation/> truy cập lần cuối 13/05/2019
- [5] <https://www.w3.org/2001/sw/wiki/RDFS> lần cuối truy cập 13/05/2019
- [6] <https://www.w3.org/OWL/> lần cuối truy cập 13/05/2019
- [7] <https://www.w3.org/TR/rdf-sparql-query/> lần cuối truy cập 13/05/2019
- [8] **Borislav Popov, Atanas Kiryakov, Angel Kirilov, Dimitar Manov, Damyan Ognyanoff, Miroslav Goranov**, “KIM – Semantic Annotation Platform”, Proceeding of Second International Semantic Web Conference, ISWC 2003, LNCS 2870, pp. 835-890.
- [9] <https://en.wikipedia.org/wiki/AllegroGraph> lần cuối truy cập 13/05/2019
- [10] https://en.wikipedia.org/wiki/Ruby_on_Rails lần cuối truy nhập 13/05/2019
- [11] [https://en.wikipedia.org/wiki/Ruby_\(programming_language\)](https://en.wikipedia.org/wiki/Ruby_(programming_language)) lần cuối truy nhập 13/05/2019
- [12] <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller> lần cuối truy nhập 13/05/2019

- [13] M. Marcus, B. Santorini and M.A. Marcinkiewicz (1993). [Building a large annotated corpus of English: The Penn Treebank](#). In *Computational Linguistics*, volume 19, number 2, pp. 313–330.