JST: Engineering and Technology for Sustainable Development
Volume 31, Issue 5, November 2021, 047-053

Context-Based Movie Searching Using Semantic Web

Tìm kiếm và tra cứu phim theo nội dung sử dụng web ngữ nghĩa

Trinh Thanh Trung*, Pham Huy Hoang
Hanoi University of Science and Technology, Hanoi, Viet Nam
**Email: trungtt@soict.hust.edu.vn*

Abstract

When searching for a movie, users often remember only the incidents happened in the movie instead of the actors or directors of that movie. However, these searches are not supported in our current movie information systems as data query is usually based on keywords. This research proposes a solution to search and query movies based on the content of the movie or the incidents happened in the movie. In our research, we have analysed and designed some movie representation models suitable for context-based movie searching. We also propose a quadruple-based representation model to resolve the disadvantages in current semantic web's triple-based model. Our system is capable of processing user requests precisely and has proven to have advantages over current movie information systems.

Keywords: Semantic web, ontology, movie

Tóm tắt

Trong thực tế, khi tìm kiếm một bộ phim đã xem, người dùng thường chỉ nhớ đến các tình tiết xảy ra trong một bộ phim chứ thường không nhớ đến các diễn viên, đạo diễn của bộ phim đó. Tuy nhiên, việc tìm kiếm như vậy là bất khả thi đối với các hệ thống hiện tại, chủ yếu truy vấn thông tin dựa trên các từ khóa. Nghiên cứu này đề xuất giải pháp tìm kiếm và tra cứu phim theo nội dung của bộ phim sử dụng web ngữ nghĩa. Cụ thể, người dùng có thể đưa ra các câu hỏi dưới dạng các nội dung, tình tiết xảy ra trong một bộ phim để nhận về kết quả là các bộ phim phù hợp với yêu cầu. Trong nghiên cứu này, chúng tôi đã đề xuất một số các mô hình biểu diễn yêu cầu thích hợp với việc tìm kiếm các bộ phim theo nội dung. Chúng tôi cũng đề xuất mô hình biểu diễn bộ bốn giúp khắc phục các nhược điểm trong mô hình biểu diễn bộ ba hiện tại của web ngữ nghĩa. Hệ thống chúng tôi đã xây dựng có khả năng xử lý các yêu cầu của người dùng một cách chính xác, tỏ ra nhiều ưu điểm so với các hệ thống thông tin phim ảnh hiện tại.

Từ khóa: Web ngữ nghĩa, ontology, tìm phim

1. Introduction

The Internet contains an immense amount of information. Therefore, finding the right information is not always an easy task. Major searching systems, such as Google, Bing or Yahoo have been creating a great deal of improvement in searching algorithms, data scope, etc. in order to improve the task. However, most of these systems still search for information based on keywords. For example, when a user searches for "Actors played in both Titanic and Inception", these systems will look for documents with the keywords like "actor", "play", "Titanic", "Inception". There are no semantic constraints between these keywords, which will become a problem if users want to ask a question in a particular data domain or in a specific context.

Particularly, in movie information searching, searching for a movie based on keywords that appeared in the movie's name or details is not sufficient. The reason is when thinking about a movie, we often remember the characters and the incidents that happened in the movie instead of the keywords. For instance, when thinking about the movie "Star Trek", we mostly remember the spaceship and a character with pointy ears, but we could not remember how the movie is reviewed in magazines. Another example is the movie series "The secret world of Alex Mack". In that movie, there is a girl with the ability to turn into liquid. With our current search engines, there is no easy way to search for this movie's name only with that information.

As a result, we have conducted research focusing on the context-based movie searching problem. More specifically, this helps users find a movie using the characters that appeared in the movie, the incidents that happened in the movie, etc., and also other basic movie information. Therefore, the requests could be much closer to those in real life, for example, "Find a movie which has a detective character"; which "a detective" must make a major contribution to the plotline, not just simply appears somewhere in the description of the movie. Another example of the

ISSN 2734-9381
https://doi.org/10.51316/jst.154.etsd.2021.31.5.7
Received: November 25, 2016; accepted: October 26, 2021

requests is "Find a movie which has an incident that a professor murders someone", and the system must also be capable of interpreting the request in the same way as "kill", "stab to death" or "execute", which is inefficient in keyword searching.

The research was conducted based on the concepts of semantic web and ontologies. We have analysed and introduced some ontology design models to represent movie information and also proposed query processing methods from user requests based on the information. The language we used for user requests is Vietnamese for better optimisation; however, these ontology design models can also be used for other languages with minor modifications.

2. Background

The idea of semantic web was first brought by Tim Berners-Lee [1]. The principle of semantic web is to provide an improvement upon the conventional Web by organising, accessing, and processing data using semantics instead of simple data storage. This could create a major advantage in data mining over the current Web and also helps computers in understanding the information on the Web more easily. With semantic web, the data will have relationships with each other like "include", "describe", "refers to", "wrote", etc. which does not exist in current Web.

One of the main concepts of semantic web is *ontology*, which is defined as "a formal, explicit specification of a shared conceptualisation." [2] In semantic web, ontology can be simply understood as words and their constraints. Several knowledge representation languages can be used to present ontologies, such as Web Ontology Languages (OWL) or Resource Description Framework Schema (RDFS). Ontology can consist of the following components:

- Individual: instances or objects
- Classes: the type of objects or instances.
- Attributes: the information about an object to describe the specification of the object.
- Relations: describe the relationship between objects. For example, some of the popular relations are "is a kind of" and "is a part of". By extension, a relation can also be the relationship between classes, the relationship between an individual and a class, the relationship between an individual and a collection, or the relationship between collections. [3]

To represent the relations, World Wide Web Consortium (W3C) provides a specification named *Resource Description Framework*, abbreviated as RDF. [4] With RDF, resource on the Internet can be described as meta-data using XML syntax. A RDF model consists of 3 components: Resources - the object to be described; Properties - specifications or relations to describe the resource; and Statements - the description.

For processing the data, a query language called SPARQL, which is the abbreviation for Protocol and RDF Query Language is used [5]. Similar to the query language SQL for queries in the database, SPARQL can be used to query the data described in an RDF graph based on the principle of comparing graph models.

3. Related Work

While there has been a great deal of research and applications for movie searching, there is not much research in the field of movie semantic has been done. In general, most of the applications are limited to searching via keywords. Several applications also add some improvement to keywords searching, such as filtering or natural language processing, for instance. However, we have not found any application which is able to search for a movie for the incidents that happened in the movie. Research on movie semantics in general, and searching movie by incidents in particular, is also very narrow.

In terms of applications for movie searching, there are a vast number of applications. Most of them are web applications, for example, Internet Movie Database – often abbreviated as IMDb (imdb.com), AllMovie (www.allmovie.com), or Rotten Tomatoes (www.rottentomatoes.com). Most official websites of major cinemas in every country also support basic movie listing and searching. However, the similarity of these applications is that all of them implement the searching feature by looking for the keywords. For instance, IMDb is a renowned web-based system which stores and queries a large volume of data of movies and all the people involved with the movie industry. In advanced search functionality, the system allows users to search for movies by a variety of criteria, including title, genre, release date, issue company, language, and even plot. However, there is no such way to search for a movie by the incidents that happen in the movie. While a keyword can be used while searching by plot, that keyword must exist in the plot summary and must be present in the exact order, which can cause difficulty to users. A minority of applications for movie searching adds extra features other than keyword lookup to implement the searching feature. A notable example of this is the solution of Jinni (www.jinni.com), which implements certain semantic algorithms to the feature. In addition to keyword searching similar to other systems, Jinni also allows users to search for a movie using semantic tags created based on the principles and rule sets by movie experts, combined with user-contributed information and natural language processing algorithms. The tags are categorised by criteria such as genre, plot, category, people, historical period, location, awards, etc. As a result, users are able to search with requests

similar to questions in real life, for example: “Find a movie based on a true story, occurred in the 70s in the last century”, or “Find a movie about serial killers and happened in Japan”. As can be seen from the examples, this implementation still lacks the capability of searching for a movie by the incidents from the movie.

Applications of semantic web in movie searching in Vietnamese language are even sparser. Almost all of the movie information systems use the same keyword searching functionality. For example, HayHayTV which is a well-known movie listing in Vietnam only supports basic searching functionalities such as search by titles, directors, and actor names. Users can filter the search by categories, countries but these features are strictly limited.

In terms of studies related to this problem, there is not much research has been done. Suganyakala et al [6] introduced the concepts of semantic movie information searching. The paper provided the basic concepts of movie semantics as well as the idea of designing and processing movie ontologies. The paper does not focus on designing the ontologies in any specific cases. On the other hand, Sasikanth Avancha et al [7] proposed the ontologies for movie information from IMDb using RDFS. While the idea of bringing existing information from IMDb to be represented in semantic context might be useful in certain situations, the study does not provide any circumstances when this could be helpful. Besides, the ontology this research proposed does not involve the incidents happening in a movie, therefore the ontology cannot be used in our context.

4. Ontology Analysis

We proposed ontology based on a number of concepts and properties. The concepts provided are:

- Movie: categorized into categories like film, TV series, trailer, teaser, etc.
- Genre: genres of the movies such as drama, action, horror, etc. along with corresponding sub-genres.
- Country: the countries in which the movie is produced, categorised by continents.
- Character: the characters that appeared in the movie, categorised into a tree-like structure including human, animal, and plant nodes.
- Crew: the people participating in making of the movie, including actors, directors, etc. While this concept can be merged into Character, the separation will ensure clarity in semantics and original usages.

In terms of properties, the movie will consist of the following data properties:

`Name, Description, Duration, Poster, Release Year, IMDB Score, IMDB link, Resource link`

along with the object properties:

`Under category, Directed by, Played by, Include character, Has incident`

Similarly, the crew members (directors, actors...) will consist of the following data properties:

`Name, Birthday, Picture, Detailed info`

along with the object properties:

`Direct` (for directors), `Play` (for actors)

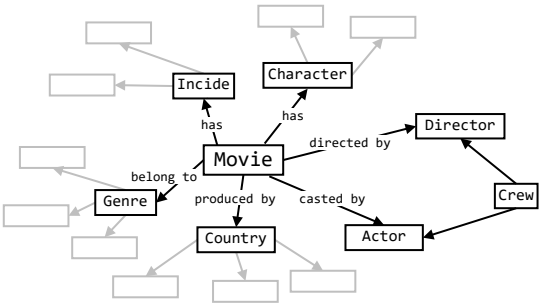

Fig. 1. Movie ontology representation

With this representation, the movie can be searched through the properties associated with it but is not quite flexible and does not guarantee that users will be able to search by the different expressions of synonyms. In fact, when we talk about a movie that has a doctor, it also means we are talking about the movie which has aesculapians, healers, physicians, etc., or even less relevant characters such as nurses or midwives. Therefore, the system must be able to recognise different expressions or synonyms.

The problem is, in this situation, doctor, healer, and physician, etc. are still different character types, but they still have a certain correlation with each other. As a result, we added an annotation property called Keyword for classes and instances to make it more flexible in receiving and understanding requests from users. This will be essential when the words input from users are different from ones specified in the Ontology. For example, when the user searches for “a character sacrifices himself” while there is only the incident “the character dies”, the system will not be able to understand and give appropriate results. The problem can be solved by adding relevant keywords. For example, “sacrifice”, “pass away” will be given the same meaning as “die”, or “murder”, “exter-minate” will be given the same meaning as “kill”.

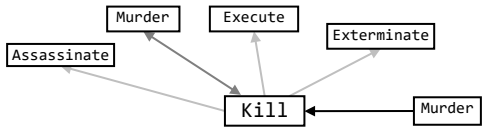

Fig. 2. Keyword property

Given the analysis, we continue to find a way to present the search request from users.

5. Search Request Presentation

5.1. Triple-Based Representation Model

To provide a model for users' search requests, we have collected a majority of common requests from users. Some of the user requests, for instance, are "Find a movie called Titanic", "Find a movie directed by Steven Spielberg", or "Find a movie having a doctor character and has kidnapping content".

For such search requests, we could use Resource Description Framework (RDF) to represent the relationships. RDF is a World Wide Web Consortium (W3C) specification, which uses an entity composed of subject - predicate - object, called triple, to present a statement. [8]

- Subject: the location of the resource we want to describe
- Predicate: the relationship describes the property of the resource
- Object: the content of the property. This could be a value (e.g. string, number, date time), or another resource.

For instance, A search request example "Phim có tên là Titanic" in Vietnamese (translation: Movie called Titanic) could be represented as a triple:

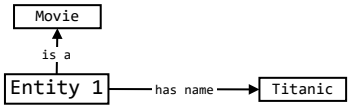

Fig. 3. Request "movie called Titanic"

In another example, "Phim được đạo diễn bởi Steven Spielberg" (Movie directed by Steven Spielberg), could be represented in triple-based model as below:

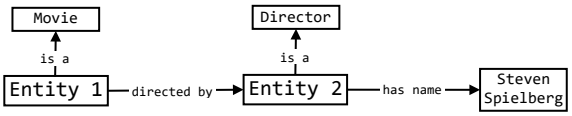

Fig. 4. Request *"Movie directed by Steven Spielberg"*

The model becomes more complicated with the third example, "Phim có nhân vật cảnh sát và tình tiết nhân vật giết người" (Movie having a police character and a murder incident):

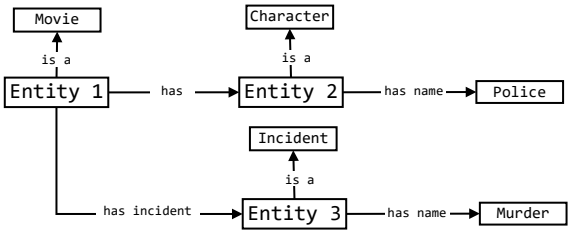

Fig. 5. Request *"Movie having a police character and a murder incident"*

As can be seen from the figures above, with basic search requests, mapping to corresponding ontology properties is fairly simple. However, with more complicated requests (consisting of multiple concurrent criteria and their associations, similar to natural language), the triple-based representation exposes noticeable limitations. This can be seen in the following example: "Phim có tình tiết cảnh sát giết người" (Movie having an incident that a police officer murders someone).

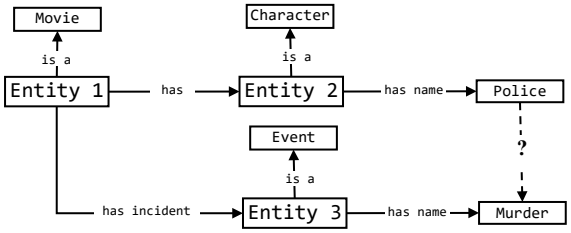

Fig. 6. Request *"Movie having a police officer murders someone"*

Using triple-based search request representation model, our current ontology could help to search for a movie that has a police officer character and also has a murder incident. Unfortunately, this representation may provide incorrect information since the combination of these two criteria does not guarantee that the user request is met. For example, this request might return a movie that has a police officer character and a murder incident, but the murder could be committed by another character, e.g. a professor or an actress. However, the result we want our system to return here is a movie that has a police officer character, and the murder incident must be caused by the same police officer.

Therefore, in our research, we proposed an improved model of the triple-based search request representation model. We called our model a quadruple-based representation model.

5.2. Quadruple-Based Representation Model

As presented in the section above, the current triple-based representation model is able to represent simple user requests. For simple statements like "Phim có tình tiết nhân vật giết người" (Movie having a murder incident), the request could be described by 3 components:

<phim><co tinh tiet><nhan vat giet nguoi> (movie - has incident - murder)

However, a more complicated statement like "Movie has an incident that a police officer kills a man" must be described by more than 3 components:

<movie><has incident><police><kill><man>

Of course, the last three components could be combined into a single component, e.g. <police kill man>, however, this is fairly inflexible and could lead to multiple creations of similar statements. For instance:

<nurse kill people>
<boy kill zombie>
<wolf kill goat>

To improve upon the current representation model, we add another component called "context" to describe the context of our statements. This component can also be used for a sub-statement. As a result, we can represent a complicated statement as multiple simple requests, which are then represented as a triple. In our example above, the statement could be represented as below figure:

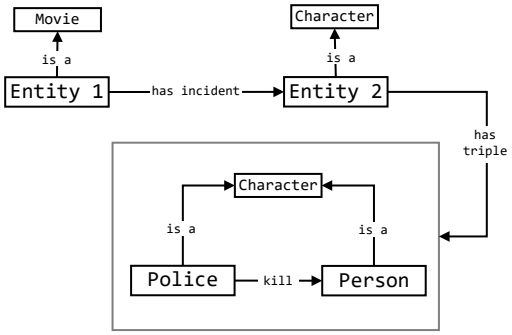

Fig. 7. Request *"Movie having a police character murder someone"*

6. Experiments

6.1. System architecture

Our proposed system consists of three modules: Query processor, data connection, and user interface. The architecture of the system is modeled as presented in Fig. 8.

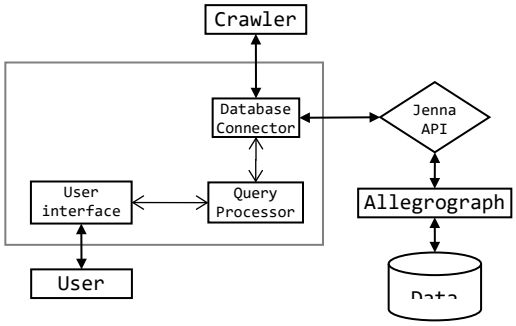

Fig. 8. System architecture

We have built a web application for the user interface. For search by tag functionality, our system receives user requests normalised with the help of Javascript based on the input keywords.

After the request from the user has been standardised, we process the request following these steps respectively:

1) Based on pre-defined SPARQL templates, we split the request into smaller criteria to build query components
2) Combine SPARQL query components into a complete SPARQL query, which is then brought to Allegrograph via Jena API. (see Fig. 9)
3) Allegrograph receive the data and convert these records to objects
4) Return the result to the user. (see Fig. 10)

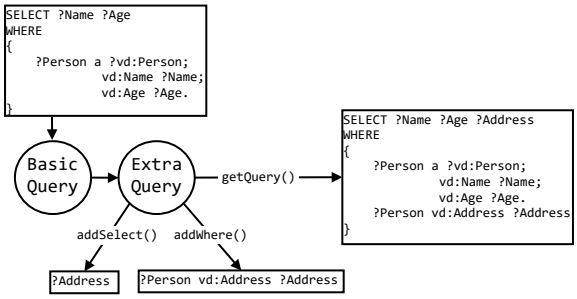

Fig. 9. Query Builder model

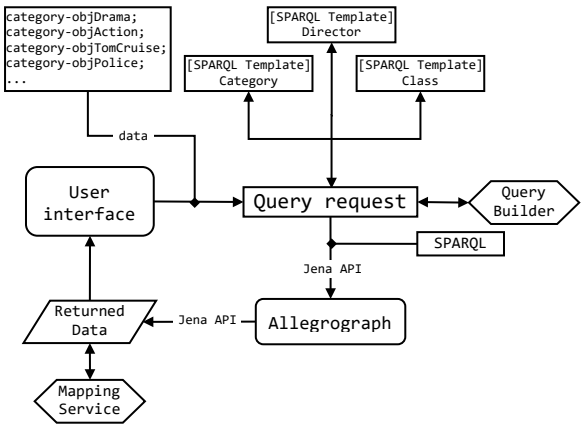

Fig. 10. User request processing

6.2. Implementation

Since user requests are not in natural language but are provided as tags instead, our system also includes a module to provide suggestions based on the keywords entered by users. In this module, the Keyword property of each class and entity allows the system to provide equivalent suggestions even when the keyword entered by the user does not match the

data specified from ontology. The returned result will be entities inside the criteria supported by our system, including Movie name, Director, Actor, Category, Character, Incident, Country. Users can add these criteria to the search request by clicking on the suggestions.

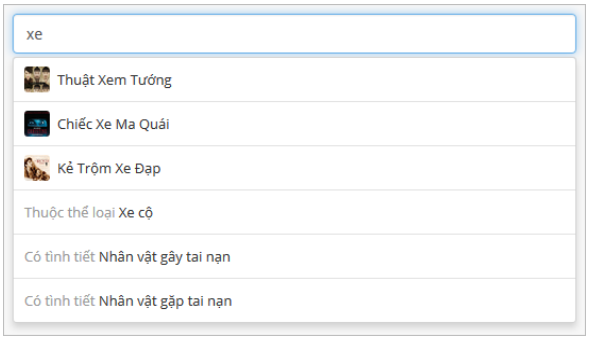

Fig. 11. Search suggestion

As can be seen from the figure above, the incident "Nhân vật gây tai nạn" (A character causes an accident) does not contain the word "xe" (vehicle); however "vehicle" is presented as a keyword of this incident. Therefore, the word "vehicle" is also shown as a suggestion for this search request. This could be really useful for the user, for example when he wants to search for a movie which has an incident that a character was hit by a car or a movie in which one of its characters has an accident when driving.

The system will then combine the requests from the user and return the results to the user. The result could be expanded to display related information such as Description, Category, Year, Character, Incident, etc. Fig. 12 shows the search result from the user request to find a movie with "a character causes an accident".

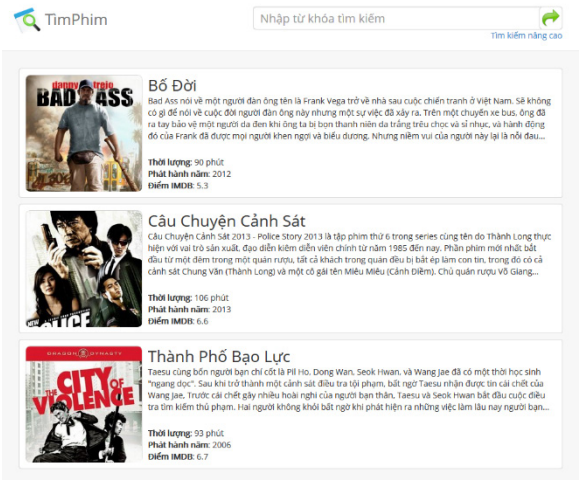

Fig. 12. Search result

Our implementation is called ***TimPhim***, which simply means "movie searching".

7. Evaluation and discussion

Evaluating the accuracy of the results returned by our system is a relatively hard process since there is not any system that can perform a similar task. Therefore, we choose to compare our system to other movie information systems with the following method: We prepare several user requests which can be given in real life and try to search within these systems. Specifically, these requests are:

1. *Movie named "Titanic"*
2. *Movie having the keyword "Police"*
3. *Movie in the "action" genre*
4. *Movie directed by "Ben Affleck"*
5. *Movie cast by "Tom Cruise"*
6. *Movie cast by "Tom Cruise" and "Jack Black"*
7. *Movie in the "drama" genre, released in UK*
8. *Movie has scenes that take place at "the sea"*
9. *Movie "based on a true story"*
10. *Movie has a "doctor" character*
11. *Movie has the incident "a character is kidnapped"*
12. *Movie in the action genre, cast by "Emma Watson"*
13. *Movie in the "science fiction" genre, has a "police" character and has the incident "a character has an accident"*
14. *Movie has the incident "a doctor murder people"*
15. *Movie has the incident "a women police officer is kidnapped"*

The movie information systems we use to compare with our system are IMDb, Jinni, and HayHayTV. The reason we choose IMDb is that IMDb is the most popular movie information system. We also choose Jinni since it is one of the few movie information systems that implement semantic algorithms. For HayHayTV, we choose this because of its support in the Vietnamese language. We do not include other movie information systems in our comparison because most of them have similar features to these three systems we have selected. The result of our comparison is represented in Table 1.

Table 1. Comparison to other movie systems

Request#	**TimPhim**	HayHayTV	IMDB	Jinni
1	o	o	o	o
2	o	o	o	o
3	o	o	o	o
4	o	o	o	≠
5	o	o	o	≠
6	o	×	×	≠
7	o	×	×	≠
8	×	×	×	o
9	×	×	×	o
10	o	×	×	×
11	o	×	×	×
12	o	×	×	≠
13	o	×	×	×
14	≠	×	×	×
15	≠	×	×	×

In the table above, the symbol ○ means the system is able to process the request and also return the right results. The symbol ≠ means the request can be processed, but the system gives incorrect results. The symbol × means the system is unable to represent the given request.

As can be seen from the table, both IMDb and HayHayTV are unable to process requests 6-15 as these requests cannot be represented by keywords. Jinni has proved to be superior in terms of semantic search, but it still exposes many flaws. For the requests 4, 5, 6, and 12, Jinni only considers "Tom Cruise", "Ben Affect", "Jack Black" or "Emma Watson" as an object of "Person" class. Similarly, for the request 7, Jinni only considers "UK" as an object of "Place" class but not the data domain users want to search for. As a result, the data returned also includes "noise". For example, if users want to find movies released in UK, Jinni might also return movies filmed in UK; or if users want to find movies directed by "Ben Affleck", Jinni might also return movies cast by "Ben Affleck".

On the other hand, our system is still unable to handle requests as 14 and 15. This limitation explains the difficulty presented in section 5.1.

8. Conclusion

This research focuses on the implementation of movie searching based on semantic web in which searches can be made from context-based request such as the characters appeared in the movie or the incidents happened in the movie. In our research, we have analysed and designed ontologies to represent movie information so as such requests from users can be modelled and processed. We also proposed a quadruple-based representation model to handle specific requests which cannot be represented using the current triple-based model.

The system we have built has proven to have an advantage over current movie information systems in many circumstances. While certain limitations still persist in the system, our approach has proven its innovation and high capabilities in future movie information systems.

Acknowledgements

This work was supported by Hanoi University of Science and Technology under grant number T2015-222.

References

[1] Berners-Lee, Tim, James Hendler, and Ora Lassila, The semantic web, Scientific American, 284(5), pp. 34-43. 2001
https://doi.org/10.1038/scientificamerican0501-34

[2] Kincho H. Law, Ontology: basic definitions and a brief introduction, TN-2007-03. NEESit – Workshops 2007.

[3] N. Guarino, Formal ontology in information systems. Proceedings of FOIS'98:3-15, Trento, Italy, 6-8 June 1998. Amsterdam, IOS Press.

[4] S Decker, F van Harmelen, J Broekstra, M Erdmann, Dieter Fensel, Ian Horrocks, Michel Klein, Sergey Melnik, The semantic web - on the respective roles of XML and RDF, IEEE Internet Computing, 4(5), pp. 63-74, 2000.
https://doi.org/10.1109/4236.877487

[5] DuCharme, Bob, Learning SPARQL: querying and updating with SPARQL 1.1., O'Reilly Media, Inc., 2013.

[6] R. Suganyakala, R. R. Rajalaxmi, Movie related information retrieval using ontology based semantic search, In 2013 International Conference on Information Communication and Embedded Systems (ICICES), IEEE, 2013
https://doi.org/10.1109/ICICES.2013.6508320

[7] S. Avancha, S. Kallurkar, T. Kamdar, (n.d.). Design of ontology for the internet movie database (IMDb). Retrieved August 20, 2016
http://ebiquity.umbc.edu/ontologies/imdb0.9/imdb.pdf

[8] Aurona Gerber, Alta van der Merwe, Andries Barnard, Semantic web technologies, UNISA-TR-2006-02, 2006.