Domain Ontology to find relevant articles

C Sneha
Information Technology
National Institute of Technology
Karnataka
Surathkal, India
snehac.181it112@nitk.edu.in

Bhagyashri Bhamare
Information Technology
National Institute of Technology
Karnataka
Surathkal, India
bhamare.bhagyashri1999@gmail.com

Gayathri Gutla
Information Technology
National Institute of Technology
Karnataka
Surathkal, India
gutlagayathri.181it216@nitk.edu.in

***Abstract*—Recommendation systems are getting more and more popular with the increasing internet usage. Many recommender systems are being developed, for movies, books, e-commerce, web page recommendations etc. Domain Ontology is of great help when it comes to recommendation systems. It added increased precision to the already existing recommender systems. They link related web pages, books etc based on the keywords. And based on the percentage of similarity they recommend the most relevant web pages, if it is a web page recommendation system, or books if it is a book recommender system.**

INTRODUCTION

The idea of this project is taken from an IEEE paper on web recommendation system using domain Ontology and web usage. Here we executed a part of the project, that is domain ontology. We try to code the system such that it takes in the input, tokenizes it into keywords and gives out the most related outputs.

For training the model, we created a dataset that is basically a collection of articles. The system gets trained on the dataset and creates a model that stores

A. PROBLEM STATEMENT

Get the most relevant articles to our input from the dataset.

B. **OBJECTIVES**

- To train the model to get the most relevant articles from the dataset.

LITERATURE REVIEW

A. Domain Ontology Construction

The technique to the preparation of domain ontology is a hierarchically structured set of concepts describing technique preparation domain of knowledge, which can be used to create a knowledge base. Technique preparation domain ontology contains concepts, relations between concepts, attributes, hierarchies, functions and possibly other axioms. It may also contain logic rules and cases

There are four basic logic relations between concepts, such as part-of, kind-of, attribute-of and instance-of. According to the above description, the formalization definition of technique preparation domain ontology

A.1. Collecting the terms

We follow some steps to collect the terms that are to be used in the construction of domain ontology. We first collect the web log file of the website from it's server over a period of one week. We then pre-process the web log file and produce a list of URLs that were accessed over the period. Then we extract all the titles from the webpages and finally we extract the terms, single tokens by removing stop words from the title. The same can be achieved from a dataset. Some terms are combined to make composite terms if they often occur at the same time with no other token in between them. Others are left as single tokens.

A.2. Define the concepts

Now all the terms are not independent of each other. So we categorize them into different concepts. We choose the concepts based on the terms and make the terms as different instances of the concepts.

A.3. Define taxonomic and non-taxonomic relationships

According to this paper we use a hybrid taxonomy, that is a combination of top down taxonomy and bottom up taxonomy. For example if we take the concept 'application' then the relation 'consistsOf' can be mapped to itself, because an application might have sub applications.

Non-taxonomic relations are also used in this model. Non-taxonomic relations can be thought of as the relations in a relational database, ie, M:N relations etc. For example the relation 'provides' can be thought of as a non-taxonomic relation of M:N order, because M manufacturers can provide N products.

The taxonomic and non-taxonomic models are combined to construct the domain ontology model.

The domain ontology(O man) constructed using taxonomic and non-taxonomic relationships can be defined as a tuple containing T man , D, A, B, ie, O man = <T man , D, A, B> where

O man = domain ontology

T man = set of domain terms in the given website

D = set of web-pages in the given website

A = set of association relations which are the taxonomic and non-taxonomic

B = set of axioms, e.g., an instantiation axiom assigning an instance to a class etc

This domain ontology is constructed at three levels:

1) General level, which holds the concepts that present the general domain terms of Web-pages and relationship definition sets;

2) Specific level, which holds the specific domain terms corresponding to the domain concepts, e.g. terms "Database" and "Office" are the instances of concept Application, and the relationships between terms;

3)Web-page level, which holds all the Web-pages within the given website, and the association relationships between web-pages and terms.

By matching keywords in terms and Web-page titles, the system can automatically map the web-pages with respect to the domain terms.

B.1 Framework of the approach

The framework of the approach is composed of three tiers such as application tier, ontology tier and data tier,. In this framework, the definition of every tier can be described as follows: Data tier is composed of five types' technique preparation information, such as product information, process information, resource information, task information and knowledge information. Ontology tier is the central part of the information retrieval approach. It is composed of several tools and models that provide information services, computing services and reasoning services to carry out the information retrieval requests.Application tier is an interface for users to access our system. The technique preparation information retrieval requests are input into the system and information retrieval results are shown for users through the application tier.

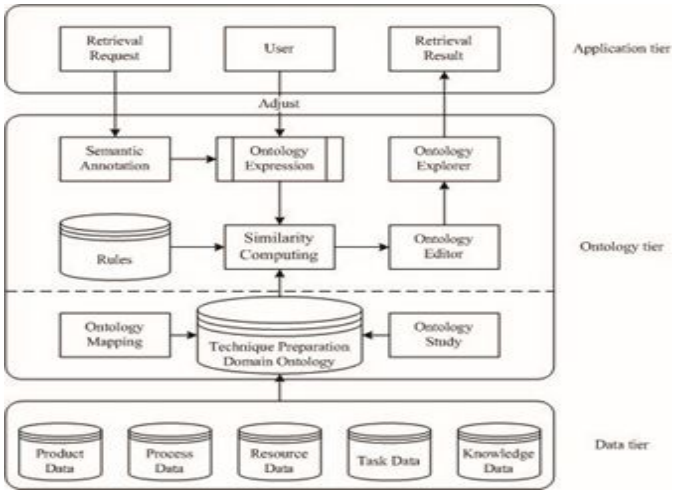

Fig1 .Framework of the approach used

METHODOLOGY

There are many modules involved in this project. It uses python libraries like numpy, sklearn, spacy, matplotlib.py_plot etc.

A. Pre-processing of data

The dataset used is a collection of articles. These are read word by word and stored in a list. This list is then pre-processed to remove any words that occur too frequently or rarely so that words can be grouped into topics more efficiently.

We define a class Vocab that takes the chosen tokenizer as a parameter. For tokenizing we use the 'en_core_web_sm' model. We store the obtained data into a vocab.mod file. The various keywords that occur in the dataset are stored in self.vocabulary_.

A function named to_bow is defined in the Vocab class. We pass a line of text as input, which is preprocessed to eliminate all the unused symbols like white spaces, periods, uni-code etc. Then the list of keywords obtained after preprocessing the input text are stored in a 'seq' list. The words in the seq list are compared with the words stored in self.vocabulary_ and if present there, they are output to the UI.

B. Visualization

We use libraries pyLDAvis for model visualization and genisim for unsupervised machine learning. We store all the keywords in a list and use 'gensim.corpora.Dictionary' to store them in a dictionary. We apply 'doc2bow' on the dictionary we have obtained to get a list of tuples that contain the token(keyword) id and the token count.

C. Topic extraction

All keywords are then stored in a dictionary by mapping them to integer values using gensim.corpora.Dictionary(). Then, doc2bow is used to store the frequency of occurrence of each of the keywords against their corresponding indices. After this is done, the function gensim.models.ldamodel.LdaModel() is used to group the words into topic.We use the ldamodel to get the topics in each article and store it in 'topics'. We use the 'ldamodel.state.get_lambda()' to normalise each row of the ldamodel to get a probability of 1. We use the wordcloud library to visualize our distribution of keywords.

D. Visualization as a graph

A heatmap is used to store the values which show how closely the topics are related to each other. If the heatmap value between two topics is greater than 0.3, then a link is drawn between the two.

E. Training the model

We use ‘gensim.models.doc2vec.Doc2Vec’ to train our model. Then given any input we first send it to sort_relavant function to get the relevant keywords from the input and then using the model give out the most relevant articles as output.

F. Output

The input from the user is processed to extract the keywords which are then sorted into topics. From these topics, the relevant articles are fetched from the dataset. These are then sorted according to the relationship weight in descending order. This is the final output.

Results and Analysis

As an example we have taken a sample dataset that contains a summary of novels of 10 novels from different sources.

Fig2. Topics created

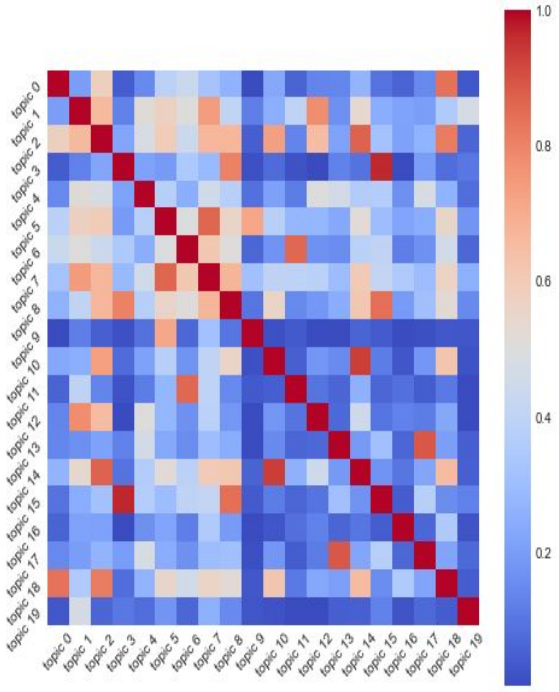

Fig2.Relation between the topics represented as a heatmap

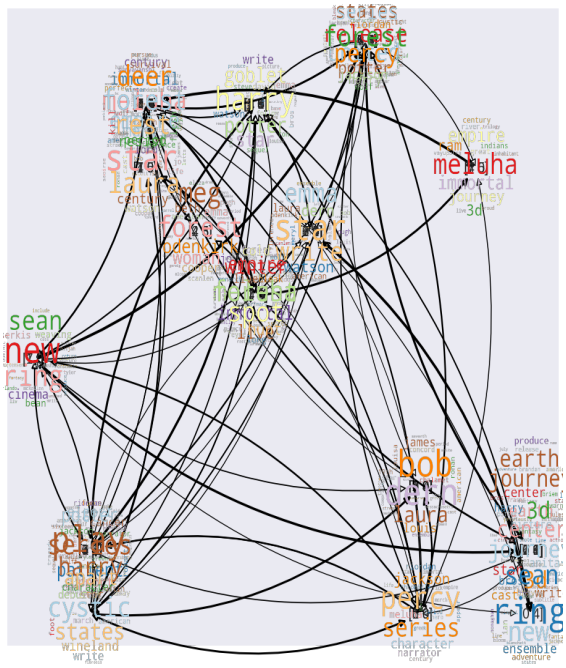

Fig4: Relation between the topics

Your question is: Harry Potter

['harry', 'potter']

TOP 0 RELEVANT (45, 0.9158288240432739): «Harry Potter and the Goblet of Fire is a 2005 fantasy film directed by Mike Newell and distributed by Warner Bros. Pictures, based on J. K. Rowling's 2000 novel of the same name. Produced by David Heyman and written by Steve Kloves, it is the sequel to Harry Potter and the Prisoner of Azkaban (2004) and the fourth instalment in the Harry Potter film series. The film stars Daniel Radcliffe as Harry Potter, with Rupert Grint as Ron Weasley, and Emma Watson as Hermione Granger. Its story follows Harry's fourth year at Hogwarts as he is chosen by the Goblet of Fire to compete in the Triwizard Tournament. »

TOP 1 RELEVANT (36, 0.9095256328582764): «Harry Potter and the Goblet of Fire is a 2005 fantasy film directed by Mike Newell and distributed by Warner Bros. Pictures, based on J. K. Rowling's 2000 novel of the same name. Produced by David Heyman and written by Steve Kloves, it is the sequel to Harry Potter and the Prisoner of Azkaban (2004) and the fourth instalment in the Harry Potter film series. The film stars Daniel Radcliffe as Harry Potter, with Rupert Grint as Ron Weasley, and Emma Watson as Hermione Granger. Its story follows Harry's fourth year at Hogwarts as he is chosen by the Goblet of Fire to compete in the Triwizard Tournament. »

TOP 2 RELEVANT (9, 0.8963267803192139): «Harry Potter and the Philosopher's Stone (released in the United States and India as Harry Potter and the Sorcerer's Stone) is a 2001 fantasy film directed by Chris Columbus and distributed by Warner Bros. Pictures, based on J. K. Rowling's 1997 novel of the same name. Produced by David Heyman and screenplay by Steve Kloves, it is the first instalment of the Harry Potter film series. The film stars Daniel Radcliffe as Harry Potter, with Rupert Grint as Ron Weasley, and Emma Watson as Hermione Granger. Its story follows Harry Potter's first year at Hogwarts School of Witchcraft and Wizardry as he discovers that he is a famous wizard and begins his education. »

Fig5:Output for given input from user

CONCLUSION

We have developed a model for the retrieval of articles which are most relevant to a given input using domain ontology. For the future work, this can be further developed to use in web searches to give results using domain ontology. It can also be integrated into other applications to get optimized search results.

References

[1] Web-Page Recommendation Based on Web Usage and Domain Knowledge https://ieeexplore.ieee.org/document/6514870

[2] B. Liu, B. Mobasher, and O. Nasraoui, “Web usage mining,” in Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data, B. Liu, Ed. Berlin, Germany: Springer-Verlag, 2011, pp. 527–603.

[3] B. Mobasher, “Data mining for web personalization,” in The Adaptive Web, vol. 4321, P. Brusilovsky, A. Kobsa, and W. Nejdl, Eds. Berlin, Germany: Springer-Verlag, 2007, pp. 90–135

[4] G. Stumme, A. Hotho, and B. Berendt, “Usage mining for and on the Semantic Web,” in Data Mining: Next Generation Challenges and Future Directions. Menlo Park, CA, USA: AAAI/MIT Press, 2004, pp. 461–480