International Journal of Computational Linguistics and Applications, vol. 5, no. 2, 2014, pp. 115–133
Received 17/08/2014, accepted 3/11/2014, final 16/11/2014.
ISSN 0976-0962, http://ijcla.bahripublications.com

An Algorithmic Approach for Learning Concept Identification and Relevant Resource Retrieval in Focused Subject Domains

RAJESH PIRYANI, JAGADESHA H., AND VIVEK KUMAR SINGH

South Asian University, India

ABSTRACT

Modern digital world has enormous amount of data on the Web easily accessible anywhere and anytime. This ease of access also creates new paradigms of education and learning. The modern-day learners have access to lot many and in fact one of the best learning materials created in any part of the world. However, despite abundant availability of material, we still lack appropriate systems that can automatically identify learning needs of a user and present them with the most relevant (and best-quality) material to pursue. This paper presents our algorithmic design towards this goal. We propose a text processing-based system that works in three phases: (a) identifying learning needs of a learner; (b) retrieving relevant materials and ranking them; and (c) presenting material to learner and monitoring the learning process. We use know-how of text processing, information retrieval, recommender systems and educational psychology and presents useful and relevant learning material (including slides, videos, articles etc.) to a learner in a focused subject domain. Our initial experiments have produced promising results. We are working towards a Web-scale deployment of the system.

1 INTRODUCTION

With newer form of digital storage devices, large screen readers and fast Internet access, we now have a large volume of anytime anywhere

accessible content. The ease of creation and the resulting rich material is paving the way for new paradigms of education and learning. However, the large amount of online/digital content makes it difficult to identify the most relevant one on a given topic. Imagine, a user, while reading some article/book chapter on 'Introduction to Machine Learning', is automatically presented with related quality resources (such as slides, videos).

This process will not only augment the learning material pursued by the user but will also substantially improve the learning experience/outcome. This automated process of learning resource identification, however, involves complex set of steps. First, we need to know the learning needs of a user, often without an explicit statement by the user. Secondly, good quality and most relevant learning material, in different forms, need to be identified (extracted from the web) and ranked in the order of their relevance and quality. Lastly, selected learning material should be presented to the user and the learning process should be monitored for implicit feedback from the user.

In this paper, we describe our algorithmic design and experimental work towards this theme. We propose to design an adaptable learning resource recommender system, which can effectively enhance the learning outcome by augmenting the learning environment of the user, with additional set of knowledge resources for the given learning concept being pursued by the user. The system assumes that there is a user with a specific learning need. However, the user need not specify it and the system should learn the same through user context and modeling. Thus, when a user is reading a particular piece of a text, the system should automatically extract the learning concepts described in the text, rank them in order of importance and use them as input for additional resource identification.

The additional resource identification process is similar to web search, where relevant articles/slides/videos located anywhere on the web need to be recalled and presented to the user. It is also equally important to measure whether the learning material so recommended is useful and relevant for the user or not. This requires a user interface with capability to monitor and log user learning behaviour (such as user clicks, on screen time etc.). The monitoring provides necessary feedback to the system and allows to adapt to the user learning behaviour and preferences. Thus, the system has three identifiable phases/parts: Concept Identification, Relevant Resource Locator and Adaptable User

Interface. We have used know how from Text Analytics, Computational Linguistics, Information Retrieval, Educational Psychology in designing the system.

The rest of the paper is organized as follows. Section 2 explains the system design and architecture and defines the relevant entities. Section 3 describes the process of parsing the document content, extraction of concepts from different sections, ranking the concepts in the order of their importance. Section 4 explains the learning resource identification and relevance ranking process. Section 5 talks about user modeling and adaptation useful for the system. We present a toy model of the system with the small dataset and experimental results obtained in the focused subject domain in Section 6. The paper concludes with a short discussion and further work to be done for a web-scale deployment of the system. This idea has appeared in a preliminary form in (Singh et al. 2013a) and the part of the work in a different context in (Relan et al. 2013) and (Khurana et al. 2013).

2 System Definition and Architecture

As the first step, an entire system can be depicted by one context diagram, the same is shown in the Figure 1 This figure gives an overview of architecture of the complete system. The system, however, can be more formally described mathematically as follows.

Let

$$U = \{A_1, A_2, \dots, A_n\}, \tag{1}$$

where U is a finite set of attributes$A_1, A_2, \dots, A_n$, which represents user psycho-graphic profile such as on screen time, resources clicked and etc.

$$C = \{c_1, c_2, \dots, c_m\}, \tag{2}$$

where C is a finite set of learning concepts $c_1, c_2, \dots, c_m$ And

$$R = \begin{pmatrix} c_1 \rightarrow r_{11} \dots r_{1j_1} \\ c_2 \rightarrow r_{21} \dots r_{2j_2} \\ \dots \quad \dots \quad \dots \\ c_i \rightarrow r_{i1} \dots r_{ij} \end{pmatrix}, \tag{3}$$

where R is a collection of resources and organized as a linked list where each r_{ij} represents the resource j for the learning concept i, which can be Article, Video, and Slides. Each r_{ij} is sorted according to the ranking

of the resource as explained in the Section 5. Now we introduce a resource match function f which is given as:

$$f: R \rightarrow U \tag{4}$$

which recommends the resources based on the user experience g, and is given by:

$$g = \sum_{i=1}^{m} \sum_{j=1}^{n} h_{c_i r_j} \tag{5}$$

where m is the number of concepts n is the number of resources $h_{c_i r_j}$ represents the j^{th} resource for i^{th} concept h is a resource refining function, which is defined as follows:

$$h_{c_i r_j} = \begin{cases} dfb(c_i r_j) + cf(c_i r_j) * ost(c_i r_j), & cf \text{ and } dfb > 0, \\ dfb(c_i r_j) & if\ cf = 0, \\ 0 & otherwise, \end{cases} \tag{6}$$

where $dfb(c_i r_j)$ is direct feedback from the user for the resource r_j of a particular learning concept c_i. $cf(c_i r_j)$ is click feedback(either 0 or 1) for the resource r_j of a particular learning concept c_i, and $ost(c_i r_j)$ is the on screen time spent on the resource r_j of a particular learning concept c_i.

All these are obtained from user browsing behavior. Our goal is to maximize the function g by refining the recommendations with the most relevant resource for the learning concepts to enhance the understandability.

3 CONCEPT EXTRACTION

The first phase of our system extracts learning concepts from a document being read by the user. This requires a number of tasks as shown in Figure 2 ranging from POS tagging to concept filtering. First of all we parse the textual contents of a document and then use knowledge of linguistics to identify patterns that can represent concepts, there are various methods to do this as described in (Joorabchi and Mahdi 2013). The concepts so identified are subjected to a filtering process for identifying Computer Science (CS) domain concepts. The CS domain

concepts present in a section are then ranked in order of importance for use by the resource retrieval phase. For concept extraction, we had to first do multitude of text extractions from the document, currently we are considering eBook as a document that included extracting Table of Contents, Chapter and Section texts. This was followed by POS tagging and terminological noun phrase identification.

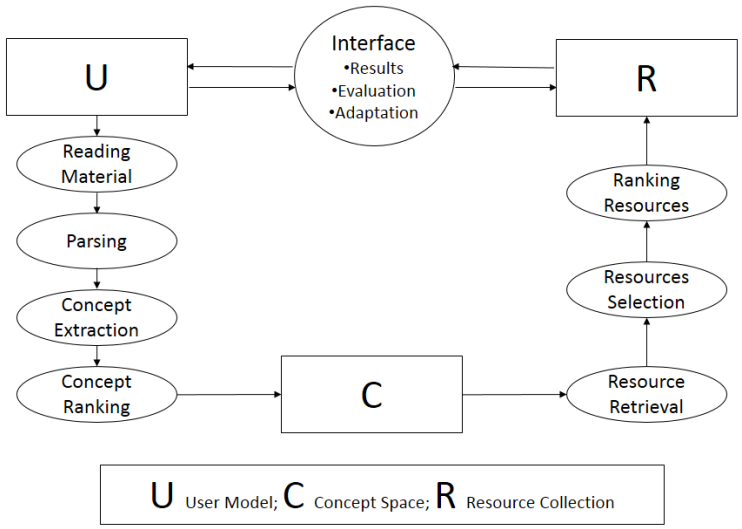

Fig. 1. Architectural Block Diagram of the System

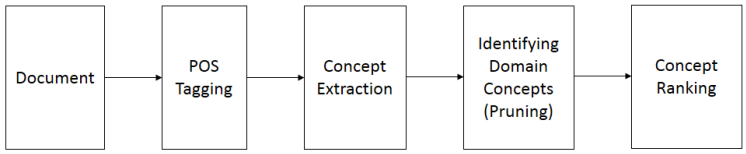

Fig. 2. Concept Extraction Block Diagram

3.1 *Learning Concept Extraction*

We extracted concepts using the terminological noun phrase identification, a set of three kinds of patterns known to represent important noun-phrase based concepts, based on the idea proposed in (Agrawal et al. 2011; Justeson and Katz 1995):

$$P1 = X^*N \quad (7)$$

$$P2 = (X^*NP)?(X^*N) \quad (8)$$

$$P3 = A^*N^+ \quad (9)$$

where, N refers to a noun, P a preposition, A an adjective, and $X = A\ or\ N$. The pattern $P1$ represents a sequence of zero or more adjectives or nouns which ends with a noun. The pattern $P2$ is a relaxation of $P1$ that allows two such patterns separated by a preposition. Examples of the pattern $P1$ may include "probability density function", "fiscal policy", and "thermal energy". Examples of the pattern $P2$ may include "radiation of energy" and "Kingdom of Ashoka". The pattern $P3$ corresponds to a sequence of zero or more adjectives, followed by one or more nouns. In $P3$, an adjective occurring between two nouns is not allowed that means it is a restricted version of $P1$.

It would be pertinent to mention here that symbol $*$ provides for zero or maximal pattern matches and $+$ provides for one or more pattern matches i.e., there is no chance to get "density function" as an extracted pattern if the actual concept mentioned is "probability density function".

Identifying terminological noun phrase patterns from the text require a number of text analytics steps. First of all we have to extract various parts (sections) of the eBook. Then we apply POS tagging on each section extracted. We used Stanford POS tagger[1] for this purpose. This paves the way for identifying terminological noun phrases. The terminological noun phrases so identified are noun phrase based concepts described in a section. A section may contain many such concepts. We have to do two things to proceed further. First, we need to distinguish CS domain concepts from other concepts. Secondly, we need to identify most important learning concepts for a section.

3.2 *Identifying CS Domain Concepts*

The terminological noun phrases extracted represent generic noun-phrase based concepts. Not all of them represent concepts belonging to CS domain. In order to identify relevant R to recommend, we need to know precisely what CS domain learning concepts are described in an eBook section. We have therefore tried to filter out the concepts not in the CS domain. For this, we have used a filtering list containing key

[1] http://nlp.stanford.edu/software/tagger.shtml

learning concepts in CS domain. We understand that this list could not be an exhaustive list of CS domain learning concepts. This may result in losing some CS domain learning concepts, however, the list is appropriate enough to identify key concepts in different subjects of study in CS domain. We have used ACM Computing Curricular Framework document[2] (ACM CCF) as our base CS domain learning concepts. We have augmented these concepts by incorporating in it terms from IEEE Computer Society Taxonomy[3] and ACM Computing Classification System[4]. The augmenting process involved merging the two later documents into the first one, while preserving the 14 categories it is divided into. The combined list is thus a set of 14 different sets of CS domain knowledge areas, each knowledge area containing key concepts (the important ones) worth learning in that area. We use this concepts as our filtering list.

Every concept identified through the terminological noun phrase identification process, is subject to this filtering. However, we cannot do an exact term matching. For example, two terms "algorithm complexity" and "complexity of algorithm" will not be a match, if we go for exact matching scheme. Therefore, we have used Jackard similarity measure, which allows two concept phrases to result in a match even when the word orders in the two are different, or there is an impartial match. The Jackard similarity equation is given in the equation below:

$$Similarity\ (C_R, C_T) = \frac{|C_R \cap C_T|}{|C_R \cup C_T|}, \tag{10}$$

where C_R is the concept in reference document and C_T is a concept in text.

Here, $C_R \cap C_T$ is the set of common words in both concepts, $C_R \cup C_T$ is the set of union of words in both concepts and S stands for the number of elements in the set S. We have to set a threshold value for deciding whether concept C_R and C_T constitute a match. We empirically found a threshold between 0.5 and 0.6, works best for identifying CS domain learning concepts. A simple example could help in understanding the suitability of this threshold. Consider, a concept $C_R =$ *"methods of numerical analysis"* is an identified terminological

[2] http://ai.stanford.edu/users/sahami/CS2013/ironman-draft/cs2013-ironman-v1.0.pdf

[3] http://www.computer.org/portal/web/publications/acmtaxonomy

[4] http://www.acm.org/about/class/2012

noun phrase and a concept $C_T = "numerical\ analysis\ methods"$ is a concept in the CS domain. In this case we get the similarity score = 0.75, greater than threshold and confirming that C_R is a valid CS domain learning concept. Thus, we use the reference list and similarity scores for deciding about every terminological noun phrase extracted from an eBook for being a valid CS domain learning concepts.

3.3 *Ranking Learning Concepts by Importance*

Our implementation tells us that a typical section in an eBook may have occurrences of several valid CS domain learning concepts. Since, we have to recommend resources R for eBook reader pursuing a particular learning concept c_i, we need to select only the most important learning concepts as the input for generating R. This means that if an eBook section results in 10 valid CS domain learning concepts, we can simply not generate R for all the 10 learning concepts, since it would make the R ineffective. We have to, therefore, restrict the learning concepts to be used as input for the process of generating R. This is equivalent to try identifying most important learning concepts in a section. An ideal position will be if we have a scheme to figure out learning concepts semantically, a section is about. But, in the absence of such a scheme to identify semantic tags about learning concepts described in a section, the only option is to use statistical evidence about the concept importance in a section. We have used statistical measures of term occurrence in the concerned section and the entire eBook to rank the learning concepts in order of importance. The rank score (section-rank) of a concept c_i belonging to a particular section S_j is computed as follows:

$$RankScore(c_i, S_j) = Freq(c_i, S_j) + \log\left(\frac{NOLC}{GRank(c_i)}\right) + \alpha \qquad (11)$$

where, $Freq()$ gives the number of occurrences of a particular c_i in a given section, $NOLC$ refers to the total number of CS domain learning concepts extracted from the eBook, $GRank$ is the rank of a c_i in the entire eBook (with highest occurring c_i getting the rank 1) and α is a significance score computed as a weighted sum of metadata, topical terms, wikipedia article, etc, as discussed in the section 3.4.

Thus, we have two ranks for each learning concept, a section-rank and a global-rank. The equation makes it clear that we compute section-rank of a c_i by combining its occurrence measures in the section and the

entire eBook. If the c_i concept refers to the highest ranking concept (rank 1), the $Freq(c_i, S_j)$ value is incremented substantially by addition of log normalized measure of its importance in the entire eBook. On the other hand, if the concept c_i refers to the concept with lowest global rank ($rank = no.of\ concepts$), its log normalized measure value becomes zero (since rank is equal to the number of concepts in eBook) and the section-rank of this concept is only a measure of its occurrence in the concerned section. In this manner, we are able to compute importance of a concept in a given section (measured as section-rank). This is in a sense equivalent to attempting to find the key section (most important) for a learning concept (Agrawal et al. 2010).

3.4 *Computing Significance Score*

When an user is pursuing an article to rank the identified concepts we use significance score which is an weighted sum of wikipedia article, metadata and topical terms i.e, if any concept extracted has an wikipedia article then increase the rank of concept, same way if it has an related concepts mentioned in metadata or topical terms of a document and then increase the rank of the concept so extracted. The mathematical form is as shown below:

$$\alpha = \frac{W + M + T}{3}, \tag{12}$$

where W, M and T are Wikipedia aticle, Metadata, Topical terms respectively and their values are either 0 or 1.

4 RESOURCE IDENTIFICATION AND RANKING

Our resource identification model contains two modules (a) Crawling and (b) Ranking of R. For crawling we have considered a defined set of websites. We use our concept extraction methods to identify the concepts within the link then we associate a tag to the link on the basis of reference library, metadata, co-occurrence and frequency of concepts. For ranking the resources we are invoking web APIs to collect the features of a link such as number of views, comments, likes and rating associated to the link, if no such features are available then we rank on the basis of metadata, wikipedia article and topical terms. The concepts and links are

stored in a database. Once the links are ranked we assign a weightage to the link now this value will vary based on user psycho-graphic profile.

In our existing system, after identifying important learning concepts presented in a section of eBook, we move to second phase of the system, which is to generate recommendations for relevant eResources for the learning concepts being pursued. While a section is being pursued by a reader, we have the key concepts in that section identified and ranked. The top learning concepts then form input for the recommendation generation process. The design of the second part is fairly simple. First of all, we explored about what useful eResources may be readily available. Thereafter, we wrote a JAVA code to invoke search APIs available for this purpose and integrate the results obtained. Our system returns a number of eResources, slides from Slideshare[5] web articles from Google Web Search[6], videos from YouTube[7], microblog posts in the area from Twitter[8], details of professionals working in the area from LinkedIn[9] and related documents from DocStoc[10].

The main objective of designing the recommender system for us was to identify and recommend additional set of eResources for eBook readers. While a reader is reading a particular section of an eBook, we want to provide him with additional learning resources as well as the set of professionals working in that area. While the first is aimed at improving the learning quality and pace; second is to provide an opportunity to the reader to connect to related professionals in the area. For learning concepts pursued by a reader, we generate a set of eResource recommendations. We have designed a web-based interface for this purpose. One important issue is to rank the recommendations based on their relevance to the learning concepts being pursued by the reader. The inherent ranking provided by the APIs invoked is one way to associate relevance to the learning concepts. These APIs use a sophisticated set of algorithms to retrieve only the most relevant results for a search query. We have, therefore, not attempted to rank the retrieved eResources afresh, except while recommending related eBooks

[5] http://www.slideshare.net/about,

[6] http://www.google.com

[7] http://www.youtube.com

[8] http://www.twitter.com

[9] http://www.linkedin.com

[10] http://www.docstoc.com/about/

(where we do rank the recommendations list). Our system design is thus a content-based recommendation system approach (Adomavicius and Tuzhilin 2005; Singh et al. 2011).

5 USER MODELING

User modeling attempts to facilitate the system to improve the quality of R. Our system initially provide the user with most relevant additional R to the user, then our system keeps track of time spent on reading a particular section of an eBook to predict the ability to understand that section. If user spends more time to apprehend the section then we refine the results R with more videos or slides to reduce the comprehension burden.

Our system interact with the user to know more about his/her interests and reconsider the result set R with more related resources of his/her interest. If user click many resource R_{ij} related to a particular c_i then our system revise the results R with more in-depth resources. For example consider user is reading about a concept "machine learning" and the resource set R includes results about "machine learning", "supervised learning" if user clicks several $j^{th}R$ of "supervised learning" then our system will revise the result R with more related resources of "supervised learning".

6 DATASET AND EXPERIMENTAL RESULTS

6.1 *Dataset*

We have performed our experimental evaluation on a moderate sized dataset collected on our own. We collected about 30 eBooks in CS domain from different sources. The text corresponding to various parts of a PDF eBook is extracted using the iText API[11] and programmatically reading the bookmarks. The different parts of an eBook are then parsed at a sentence level, starting with POS tagging and culminating in identification of C (denoted by terminological noun phrases).

[11] http://www.api.itextpdf.com

6.2 *Results*

The JAVA program designed to extract C, their ranks etc. produces a lot of other useful information from eBooks. We have designed an RDF (Resource Description Framework) schema to store the information produced for each eBook. All this information is generated and written automatically (through our program) in the RDF schema. The RDF schema contains rdfs:R for the eBook metadata, C in a section and chapter, concept relations and eBook reviews obtained by crawling the Web. The eBook metadata comprises of eBook title, author, number of chapters, number of pages, eBook price, eBook rating, its main and two related categories as determined from augmented ACM CCF, coverage score, readability score and consolidated sentiment score profile. For each chapter node in the RDF, the entry consists of section and chapter titles, top C with ranks, and relations extracted for the chapter. The populated RDF structure contains a lot of other information for eBooks. We have used only some of this information for our R generation. The other information can be used for a number of purposes like querying about relevant information for the eBook, designing a concept locator in the eBook or designing a semantic annotation environment. A sample example of RDF representation of eBook metadata is as follows:

```
<rdf:RDF
xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
xmlns:book="http://www.textanalytics.in/ebooks/
  Data_Mining_Concepts_and_Techniques_Third_Edition#">
<rdf:Description
rdf:about="http://www.textanalytics.in/ebooks/
Data_Mining_Concepts_and_Techniques_Third_Edition#metadata">
<book:btitle>Data Mining Concepts and Techniques Third
Edition</book:btitle>
<book:author>JiaweiHan,MichelineKamber,Jian Pei
</book:author>
<book:no_of_chapters>13</book:no_of_chapters>
<book:no_of_pages>740</book:no_of_pages>
<book:bconcepts>rule based classification, resolution,
support vector machines,machine learning,...
</book:bconcepts>
<book:main_category>Intelligent Systems</book:main_category>
<book:main_cat_coverage_score>0.051107325
</book:main_cat_coverage_score>
<book:related_category>Programming fundamentals
</book:related_category>
<book:related_category>Information Management
</book:related_category>
```

```
<book:googleRating>User Rating: **** (3 rating(s))
</book:googleRating>
<book:readability_score>56 (Fairly Difficult)
</book:readability_score>
</rdf:Description>
```

In this representation, the category and related category refers to the two closest of the 14 classes defined in ACM CCF. Similarly, other information include readability score, author(s), number of pages etc. The figure 3 shows the RDF Graph for a part of the eBook metadata.

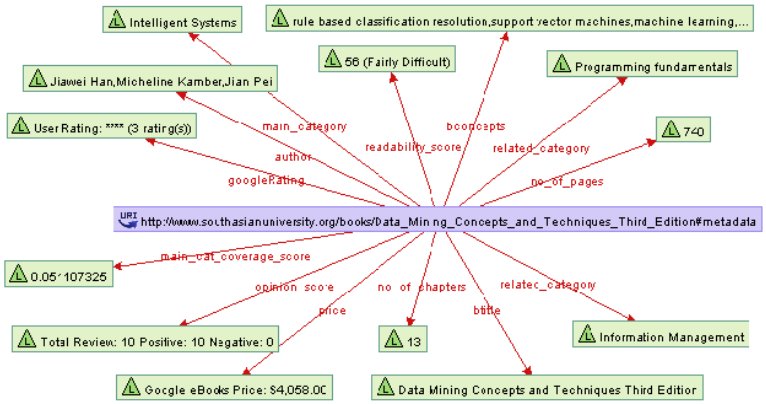

Fig. 3. RDF Graph for Book Metadata

The second key part of the information represented include information about C and their relations in the Chapter node of the RDF schema. A detailed discussion of the RDF schema and relation networks is available in (Uddin et al. 2013).

In the following paragraphs we present snapshot of some results produced at various stages of processing by our system. The snapshot of results shown correspond to a popular eBook on "Data Mining" that describes concepts and techniques of data mining and is a recommended eBook for graduate and research students. During phase 1 of system operation, we extract all probable learning concepts (measured as terminological noun phrases) from a section of the eBook. Then these concepts are filtered using the augmented ACM CCF reference document. For example, from the first chapter of the eBook having title "Introduction", we obtained 1443 concepts before filtering, out of

which 96 concepts refer explicitly to the CS domain. Some example CS domain concepts from beginning portion of this chapter are:

```
business intelligence, knowledge management, entity
relationship, models, information technology, database
management system
```

After obtaining the filtered list of CS domain C in a section of the eBook, we rank them in order of importance. This required that both local (concept occurrence frequencies in the section) and global knowledge (concept ranking for the entire eBook) are available. Thus, we parse the entire dataset of eBooks, identify C in them and rank them in order of importance (assuming whole eBook as unit), beforehand. The concept occurrence frequencies in the currently accessed section are computed at the time of their actual use by the eBook reader. As stated earlier, all the information extracted is also written in an RDF schemea for future retrieval.

The second phase involves generation of R relevant to the most significant C being pursued by the reader. Our R contain eResources of various kinds. The recommendation list R generated by us include videos from YouTube, slides form Slideshare, documents from DocStoc, Web articles from Google Web search, profile ids of professionals working in the area from LinkedIn, Articles or Multimedia from the repository and some others. We present below a sample results for a concept "Data mining" from the first chapter of the eBook used as an example demonstration. An example of recommended videos from YouTube for the concept are as follows:

```
Result for Concept: Data Mining
1.  Thumbnail:
    http://i.ytimg.com/vi/UzxYlbK2c7E/hqdefault.jpg
    URL: http://www.youtube.com/watch?v=UzxYlbK2c7E
2.  Thumbnail:
    http://i.ytimg.com/vi/EUzsy3W4I0g/hqdefault.jpg
    URL: http://www.youtube.com/watch?v=EUzsy3W4I0g
```

An example snapshot of recommended slides from SlideShare for the concept are as follows:

```
Result for Concept: Data Mining
1.  Title:The Secrets of Building Realtime Big Data Systems
    URL:http://www.slideshare.net/nathanmarz/the-secrets-of-
    building-realtime-big-data-systems
2.  Title:Big Data with Not Only SQL
```

```
    URL:http://www.slideshare.net/PhilippeJulio/big-data-
    architecture
```

A sample of recommended documents from DocStoc for the concept is as follows:

```
Result for Concept: Data Mining
1.  Title: Data Mining
    URL: http://www.docstoc.com/docs/10961467/Data-Mining
2.  Title: Data Mining Introduction
    URL: http://www.docstoc.com/docs/10719897/Data-Mining-
    Introduction
```

A snapshot of a part of recommended LinkedIn profiles for the concepts is as follows:

```
Result for Concept: Data Mining
1.  Name: Peter Norvig
    URL: http://www.linkedin.com/in/pnorvig?trk=skills
2.  Name: Daphne Koller
    URL: http://www.linkedin.com/pub/daphne-
    koller/20/3a8/405?trk=skills
```

It would be important to mention here that the results displayed are a very small part of the actual results obtained. More results can be seen at our text analytics portal[12]. Through a similar process of API invocation, we have also generated recommendations for top web links from Google Web Search and top profiles of persons writing on the topic on microblogging site Twitter. We have thus generated recommendations for a comprehensive set of eResources (in addition to identifying the most relevant eBook and its chapter) for a concept being pursued by a learner.

For a given important concept in a section, we also recommend related eBooks (ranked in order of their relevance). The recommended list of related eBooks are at present generated from our dataset collection itself. However, it is not a limitation and we can generate a list of related eBooks (related on the important C under consideration) from the Web. The list of related eBooks is ranked based on a computed sentiment score of their reviews obtained from Google book reviews and from Amazon. It was necessary to rank eBooks since the recommendation list of eBooks is not generated by an API having inherent ranking scheme, but by a concept-bases matching calculation. We want that the most popular

[12] http://www.textanalytics.in

eBooks (measured through wisdom-of-crowds) should be ranked at top and recommended. For this, we have collected user reviews of all the eBooks in the dataset by a selective crawling of Google Book review and Amazon sites. The textual reviews obtained for each eBook are then labeled as 'positive' or 'negative' through a sentiment analysis program designed by us (Singh et al. 2013b, 2013c). Thus for each candidate eBook, we compute sentiment labels and strengths of its reviews (between 10-50 reviews), normalize the strength score (by dividing with number of 'positive' or 'negative' reviews) and use it to rank the eBooks in order of their popularity. Figure 4 shows an example recommendation for the related eBooks recommended for concept “Data Mining”.

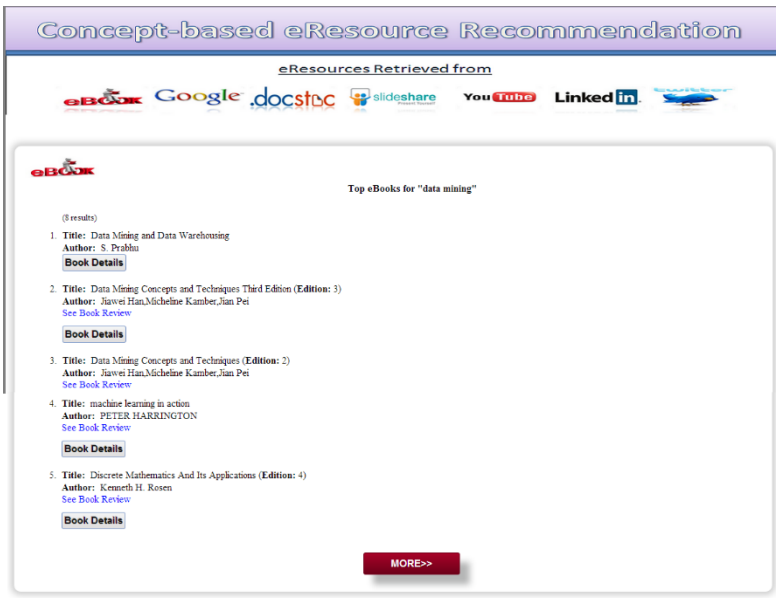

Fig. 4. Recommended eBooks for Concept: Data Mining

7 Conclusion and Future Work

We have presented our experimental work on design of concept-based eResource recommendation system. The system takes as input an eBook being currently read by a user and provides him with additional learning

resources for the learning concepts being pursued by him. The system uses a text analytics approach and works in two phases. In first phase, it identifies the main learning concepts that a user is trying to understand. In the second phase, it generates a set of eResource recommendations that are relevant to the learning concept and provide the user with additional learning material on the concept in concern. The recommender system design proposed and demonstrated by us, appears to be useful for learners.

Evaluation of recommendations is a key parameter of study for recommendation system design. Here, we have used Web APIs for collecting and recommending eResources. These APIs are inherently known to retrieve most relevant results for an information need. There is no such previous system or benchmark against which we can evaluate our system. While the first phase of the system is tested to work appropriately, the results of second phase need some more evaluations for relevance. Our preliminary observation shows that the retrieved and recommended eResources for a learning concept are the most relevant and authoritative ones. We are, however, working towards a wisdom-of-crowd kind of evaluation of the relevance of the recommendation results. Since it is largely a manual effort, it will take some more time to collect user feedbacks from the system hosted on an in-house web portal and being used by volunteers.

There are some possible improvements and extensions of the current work. One of them is to work on a large dataset and explore our system's applicability on open source eBooks from the Web not only for CS but other domains as well. Secondly, we are still working on an appropriate evaluation scheme for ascertaining the quality of recommendations generated. Though, wisdom-of-crowds seem the most natural way, other ways of evaluation may be explored. Thirdly, we wish to extend the system to a full-blown web-based learning resource recommendation system, which can automatically identify users' information needs. Fourth, behavioural and user-based modeling studies may be carried out to evaluate usefulness of the system and to deduce lessons for information need modeling of IR systems. And lastly, the linguistics-based formulations for concept identification, refinement still have possibility of improvement.

ACNKNOWLEDGEMENT This work is partly supported by a UGC-India Research Grant Number 41-642/2012(SR).

REFERENCES

1. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734–749, doi:10.1109/tkde.2005.99 (2005)
2. Agrawal, R., Gollapudi, S., Kenthapadi, K., Srivastava, N., Velu, R.: Enriching textbooks through data mining. In: Proceedings of the First ACM Symposium on Computing for Development, ACM DEV'10, doi:10.1145/1926180.1926204 (2010)
3. Agrawal, R., Gollapudi, S., Kannan, A., Kenthapadi, K.: Data mining for improving textbooks. ACM SIGKDD Explorations Newsletter, vol. 13, no. 2, p. 7, doi:10.1145/2207243.2207246 (2012)
4. Joorabchi, A., Mahdi, A.E.: Automatic keyphrase annotation of scientific documents using Wikipedia and genetic algorithms. Journal of Information Science, vol. 39, no. 3, pp. 410–426, doi:10.1177/0165551512472138 (2013)
5. Justeson, J. S., Katz, S. M.: Technical terminology: some linguistic properties and an algorithm for identification in text. Nat. Lang. Eng., vol. 1, no. 1, doi:10.1017/s1351324900000048 (1995)
6. Khurana, S., Relan, M., Singh, V.K.: A text analytics-based approach to compute coverage, readability and comprehensibility of eBooks. In: 2013 Sixth International Conference on Contemporary Computing (IC3), doi:10.1109/ic3.2013.6612200 (2013)
7. Relan, M., Khurana, S., Singh, V.K.. Qualitative Evaluation and Improvement Suggestions for eBooks using Text Analytics Algorithms. In: Proceedings of Second International Conference on Eco-friendly Computing and Communication Systems, Solan, India (2013)
8. Singh, V. K., Mukherjee, M., Mehta, G. K. Combining a Content Filtering Heuristic and Sentiment Analysis for Movie Recommendations. Computer Networks and Intelligent Computing, pp. 659–664, doi:10.1007/978-3-642-22786-8_83 (2011)
9. Singh, V. K., Piryani, R., Uddin, A., Pinto, D.: A Content-Based eResource Recommender System to Augment eBook-Based Learning. Multi-Disciplinary Trends in Artificial Intelligence, pp. 257–268, doi:10.1007/978-3-642-44949-9_24 (2013)
10. Singh, V. K., Piryani, R., Uddin, A., Waila, P.: Sentiment analysis of Movie reviews and Blog posts. In: Proc. of 2013 3rd IEEE International Advance Computing Conference (IACC), doi:10.1109/iadcc.2013.6514345 (2013)
11. Singh, V. K., Piryani, R., Uddin, A., Waila, P.: Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification. In: Proc. 2013 International Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), doi:10.1109/imac4s.2013.6526500 (2013)

12. Uddin, A., Piryani, R., Singh, V.K.: Information and Relation Extraction for Semantic Annotation of eBook Texts. In: Recent Advances in Intelligent Informatics, pp. 215–226, doi:10.1007/978-3-319-01778-5_22 (2014)

RAJESH PIRYANI
DEPARTMENT OF COMPUTER SCIENCE,
SOUTH ASIAN UNIVERSITY, NEW DELHI-110021, INDIA

JAGADESHA H.
DEPARTMENT OF COMPUTER SCIENCE,
SOUTH ASIAN UNIVERSITY, NEW DELHI-110021, INDIA

VIVEK KUMAR SINGH
DEPARTMENT OF COMPUTER SCIENCE,
SOUTH ASIAN UNIVERSITY, NEW DELHI-110021, INDIA