Open access Journal International Journal of Emerging Trends in Science and Technology

DOI: http://dx.doi.org/10.18535/ijetst/v2i9.16

Extracting Learning Concepts from E-books

Author
Vivek Kumar Singh
Department of Computer Science,
Banaras Hindu University, Varanasi-221005, India

ABSTRACT
This paper presents the algorithmic formulation to automatically extract the learning concepts from eBook and generate the RDF (Resource Description Framework) data which can be utilized for numerous purposes. The framework also extracted some metadata about the eBook such as author, price and reviews with the help of web crawling. The automated process of concept extraction and generation of RDF data is helpful for tasks like Information Extraction, Concept-based search and Machine Reading.
Keywords: *Information Extraction, Machine Reading, RDF Schemas*

INTRODUCTION

The newer types of digital storage devices with large screen readers and faster access to Internet have given a boom to the use of eBooks as a popular and environment friendly alternative to traditional printed books. This has also encouraged the creation and distribution of content freely over the World Wide Web (WWW). The impact is such that more and more publishers are now convinced to release their popular books as eBooks. As a matter of fact, it has been observed that eBooks are disseminated and used in digital electronic form over an eBook reader (like Kindle) or a personal digital assistant. It is in this regard that in this paper an algorithm has been formulated that extracts concepts described in an eBook so that semantic navigation tags can be associated to the eBook content. The concepts and the associated relation with metadata, described in eBook chapters/ sections have been extracted. The extracted information is transformed into a semantically navigable and machine readable RDF data. This eBook collection can be navigated and appropriate eBook and chapters located in them for a specific concept of interest to the reader. RDF structure is created programmatically with all the information extracted. The stored information in turn allows in identifying most appropriate eBooks (and also relevant chapters/ sections) for a particular reader willing to learn some specific concept. This structured information can be converted into machine readable form, which can be used for a variety of useful applications.

The layout of the paper is as follows. Section 2 describes the parsing of text and process of concept extraction. The concepts thus extracted are filtered for identifying concepts only in the CS domain. Section 3 describes the concept matching mechanism. Section 4 and Section 5 describes the dataset used and the experimental results respectively. This includes learning concepts extracted, RDF schema populated with data generated, and other information collected from external sources and made a part of RDF. The paper concludes in section 6 with a short summary and usefulness of the work reported here.

CONCEPT EXTRACTION

Since the target is basically eBook, the types of information extracted from these are also limited which includes entities like noun phrases or combination of adjective, noun and preposition. It is rare to get concept which contains other parts of

speech such as verbs, adverbs, or conjunctions. Here, three patterns (P1, P2, and P3) are considered for determining terminological noun phrases. The first two of these are from (Justeson and Katz, 1995) and the third one was proposed in (Agrawal et. al., 2012). One can express the three patterns using regular expressions as:

$$P1 = C * N$$
$$P2 = (C * NP)?(C * N)$$
$$P3 = A * N +$$

where N refers to a noun, P a preposition, A an adjective, and C = A|N. The pattern P1 corresponds to a sequence of zero or more adjectives or nouns which ends with a noun and P2 is a relaxation of P1 that also permits two such patterns separated by a preposition. Examples of the pattern P1 include "probability density function", "fiscal policy", and "thermal energy". Examples of the latter include "radiation of energy" and "Kingdom of Asoka". P3 corresponds to a sequence of zero or more adjectives, followed by one or more nouns. This pattern is a restricted version of P1, where an adjective occurring between two nouns is not allowed. The motivation for this pattern comes from sentences such as "The experiment with Swadeshi gave Mahatma Gandhi important ideas about using cloth as a symbolic weapon against British rule". As a result of allowing arbitrary order of adjectives and nouns, "Mahatma Gandhi important ideas" is detected as a terminological noun phrase by pattern P1. But pattern P3 would result in the better phrases, "Mahatma Gandhi" and "important ideas". Candidate concepts always comprise of maximal pattern matches. So there is no scope to have "density function" as a candidate in the presence of "probability density function".

The main target is that it is better to have more specific concepts than general concepts. A similar type of strategy was used in(Lent et al., 1997). It was also found in the study reported in (Agrawal et. al., 2010) that the pattern P1 perform better than P2. The pattern P3 showed slightly better performance than P1 in this study.

CONCEPT MATCHING

There are fourteen knowledge areas defined in AUGMENTED ACM CCF in computer science by ACM which includes standard set of concepts for each area. The concepts from ACM and IEEE taxonomy are combined to prepare the reference set. To decide a class for a given book, it is necessary to collect all concepts from the content table of the book and compare all concepts with the concepts of each category. The category having more matching is assigned to the book.

The concepts found from the patterns defined previously are not always meaningful. To filter the fruitful concepts, a measure of matching using following formula is calculated:

$$Similarity\ (A,B) = |(A \cap B)| / |(A \cup B)|$$

Where A and B stand for the two concepts. A∩B is the set of common words in both concepts and A∪B is the set of union of both concepts' words.

As one always cannot expect two concepts to match exactly in the same order of words, threshold of 0.6 is set for similarity to recognize as same concepts or in some cases relevant to the concept. Although this technique may bring some mismatch but at least one will not lose those same concepts which can be expressed using different number of words or in different orders with the same set of words. As an example, 'methods of numerical analysis' and 'numerical analysis methods' should be treated as same concept or topic. In this case above formula produces a similarity value of 0.75. Actually this technique gives the flexibility to identifying different close concept and solves a basic similarity problem in concept matching. In the other case, concept pair like 'algorithm' and 'clustering algorithm' will not be considered as same because the formula will give a value of 0.5 which is lesser than 0.6.

DATASET

The experiments have performed experimental evaluation on a moderate sized dataset collected. Data for about 30 eBooks in CS domain is collected from different sources. The text corresponding to various parts of a PDF eBook is

extracted using the iText API[1] 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).

RESULTS

The JAVA program designed to extract C produces a lot of other useful information from eBooks. An RDF (Resource Description Framework) schema to store the information produced for each eBook is designed. All this information is generated and written automatically (through program) in the RDF schema. The RDF schema contains $rdfs:R$ for the eBook metadata, C in a section and chapter, 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. 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:Descriptionrdf: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 important information include readability score, author(s), number of pages etc. The figure 1 shows the RDF Graph for a part of the eBook metadata.

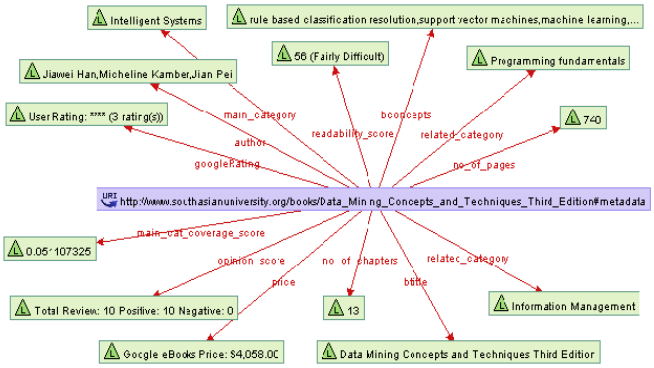

Figure 1. RDF Graph for Book Metadata

In the following paragraphs snapshot of some results produced at various stages of processing by the system is presented. The snapshot of results

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

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, all probable learning concepts (measured as terminological noun phrases) are extracted 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", total 1443 concepts are obtained 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, they are ranked 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, the entire dataset of eBooks is parsed to 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 schema for future retrieval.

CONCLUSION

In this paper, an algorithmic formulation for extraction of learning concepts from unstructured eBook text has been presented. The system designed performs three kinds of computational tasks. First, it extracts various parts of an eBook and identifies important learning concepts and relations between concepts and the metadata (it comprises of eBook title, author, number of chapters, number of pages) described in that part. This required sophisticated text parsing and mining. Secondly, the system automatically collects additional data about eBooks (such as reviews and ratings) for each eBook from the World Wide Web sources and performs computational tasks on them to draw useful inferences. The structured form RDF is generated with extracted information.

REFERENCES

1. Agrawal R., Gollapudi S., Kannan A. &Kenthapadi K. (2011a), “Data Mining for Improving Textbooks”, Newsletter ACM SIGKDD Explorations Newsletter archive, Volume 13 Issue 2, Pages 7-19 ACM New York, NY, USA.
2. Agrawal R., Gollapudi S., Kenthapadi K., Srivastava N. &Velu R. (2010), “Enriching textbooks through data mining”, In ACM DEV.
3. Justeson, J. S., & Katz, S. M. (1995). Technical terminology: some linguistic properties and an algorithm for identification in text. Nat. Lang. Eng., 1(01).
4. Lent B., Agarawal R. &Srikant R. (1997), “Discovering trends in text databases”, In KDD.