Extracting Keyphrases from books using Language Modeling Approaches

Rohini U*
AOL India R&D
Bangalore, India
rohini.uppuluri@corp.aol.com

Vamshi Ambati
Language Technologies Institute (LTI)
Carnegie Mellon University
Pittsburgh, USA
vamshi@cs.cmu.edu

Abstract

There has been a tremendous growth recently in the available digitial content. In a digital library consisting of several hundreds of thousands of books, it is clearly infeasible for a user to examine complete book to determine whether or not the document would be useful. Instead, short meta data containing descriptions of books like titles, summary, keyphrases etc., could be beneficial in helping a user in getting a quick summary about the key points in the book. We aim to perform automatic keyphrase extraction from books which can be used to get a quick overview of the key points contained. We focus on language independent approaches which can be easily be applied to other languages (other than english).

1 Introduction

Keyphrases are often chosen manually, usually by the author of a document representing some of the key information conveyed in the document. With the huge increase in the information, automatic identification of keyphrases received a lot of interest recently. Keyphrase extraction is important and useful to document summarization, information retrieval, lexicon construction, machine translation and digital libraries.

*This work was done when the first author was pursuing Masters through Research at IIIT Hyderabad, India

Digital libraries are increasingly growing in content in the recent decades and will continue to grow. With such vast growth in digital content, automatic ways of identification of keyphrases would be of great use. For example, given a digital library consisting of several hundreds of thousands of books, it is clearly infeasible for a user to examine complete book to determine whether or not the document would be useful. Instead, short meta data containing descriptions of books like titles, summary, keyphrases etc could be beneficial in helping a user to identify books of interest.

Automatic keyphrase extraction has received a lot of attention in the literture. The approaches proposed mainly are - rule based and heuristic based approaches and Statistical based approaches. Some hybrid approaches also have been proposed. Rule based approaches as the name suggests use predefined rules for automatic extraction of key phrases. On the other hand, statistical methods are used by the statistical approaches. Recently, keyphrase extraction has been studied as a learning problem and approaches have been proposed using machine learning techniques.

While the previous approaches to keyphrase extraction seem interesting and promising, there are a few problems directly adapting them to our problem. Firstly, it is difficult to specify rules for automatic extraction of keyphrases mainly because books are vast in content and it will be a tedious job. Also, we aim to build language independent automatic method of extracting keyphrases so it can be easily be used

for various languages. Hence, we resort to statistical methods. Though it is interesting to explore the use of machine learning methods, we suffer from the problem of lack of training data.

Takashi et. al [4] has proposed a language modeling approach for automatic extraction. They consider two important factors for extracting keyphrase. The first is *Phraseness* which describes degree to which a given word sequence is considered to be a phrase i.e., it measures the collocation or cohesion between consecutive words. Second is *Informativeness* which refers to how well a phrase captures or illustrates the key ideas in a set of documents. They make use of the relationship between foreground and background corpora to formalize the notion of informativeness. Here background corpus is not any training data but a general corpus which can be any general English corpus. They have applied this approach for extracting keyphrases from a news corpus data set called 20 newsgroups data set. In this paper, we report our findings applying this approach to extracting keyphrases from books. We choose this approach because it is language independent, does not require any training data and there is no necessity of tuning parameters.

A book is a collection of a huge content of text usually divided into chapters. Each chapter disscusses about a specific incident or topic depending on the book. Some times, the chapters are loosely connected and discrete though in a common line of thread. Our focus is on providing keyphrases for each chapter which can serve as a quick summarization technique for people who would be intereting to get a high level overview of what the chapter is about. Keyphrases are extracted for each chapter independent of another chapter.

The rest of the paper is organized as follows. In Section 2, we discuss some previous approaches. In Section 3, we discuss extraction of keyphrases from books in which we first give overview of the algorithm for keyphrase extraction followed by details of detailed steps of extraction from books. We applied it to some books from gutenberg project which we discuss in Section 4. In Section 6, we conclude also describing our future work.

Paper Description	*A paper on a web logging and visualization system*
Extracted Keywords	webquilt proxy logger, client side logging, proxy log user, proxy web designers, server page forwards
Paper Description	*Search Engine evaluation with clickthrough data analysis*
Extracted Keywords	navigational type queries, query type identification, annotated query set, search engine performance, type evaluation rr

Table 1: Keyphrase Extraction from Technical Documents

2 Previous Approaches to Keyphrase Extraction

Keyphrases may be associated with documents by two main methods: keyphrase assignment, and keyphrase extraction. Keyphrase assignment algorithms examine the text of a document and assign it keyphrases from a controlled vocabulary [2] among others. Creation and maintenance of the vocabulary requires time and expertise which may not always be available. Keyphrase extraction is done treating the problem as a supervised learning problem. Training data is used with phrases prior classified into two classes [5, 6, 3] among others. However, collecting such training data can be a problem especially for books.

On the other hand, some works have been done measuring the degree of collocation fo words. [1] uses the relative frequency ratio between two corpora to extract domain-specific keyphrases. [7] compare two metrics, MI and Residual IDF (RIDF), and observed that MI is suitable for finding collocation and RIDF is suitable for finding informative phrases. [4] proposed a language modeling approach to keyphrase extraction without any training data. They have experimented on news data. In this paper, we apply the approach for keyphrase extraction from books.

3 Extracting Keyphrases from books

In this section, we describe the approach employed in extracting keyphrases from books. A book is a collection of a huge content of text usually divided into chapters. Each chapter disscusses about a specific incident or topic depending on the book. Some times, the chapters are loosely connected and discrete though in a common line of thread. Our focus is on providing keyphrases for each chapter which can serve as a quick summarization technique for people who would be intereting to get a high level overview of what the chapter is about. First, the approach is described and then the process of extraction from books is described.

3.1 Approach Employed

The approach is based on language modeling techniques. A language model assigns a probability value to every sequence of words $W = w_1 w_2 ... w_n$. The probability of P(**W**) can be decomposed as

$$P(W = \prod_{i=1}^{n} P(w_i | w_1 w_2 ... w_{i-1})$$

Assuming w_i depends on the previous N words, N-gram language models are commonly used. The simplest case is unigram language model where each word is independent of each other. The probability computation for unigram language model is as follows:

$$P(W = \prod_{i=1}^{n} P(w_i)$$

The probability computation for trigram language model where each word depends on the previous two words is as follows:

$$P(W = \prod_{i=1}^{n} P(w_i | w_{i-2} w_{i-1})$$

Each keyword is associated with two important notions. *Phraseness* and *Informativeness*. Phraseness describes degree to which a given word sequence is considered to be a phrase i.e., it measures the collocation or cohesion between consecutive words. Informativeness which refers to how well a phrase captures or illustrates the key ideas in a set of documents.

Suppose there are *Foreground Corpus* which is current corpus under consideration and *Background Corpus* which is a general corpus computed from general english. The foreground and background corpus is used to compute the phraseness and informativeness values. Phraseness is computed from foreground corpus measuring how well a given phrase are collocated and the relationship between foreground and background corpora is made use of to formalize the notion of informativeness.

KL-divergence (also called relative entropy) is a metric to compare two language models. It is a measure of the inefficiency of assuming that the distribution is q when the true distribution is p. The KL-divergence between two probability mass functions $p(x)$ and $q(x)$ is defined as

$$D(p||q) = \sum_{x} p(x) log(p(x)/q(x)) \quad (1)$$

Takashi et. al [4] have defined *Pointwise KL divergence* to be term inside Equation 1. Pointwise KL divergence $\delta_w(p||q)$ between two different units in probability mass functions $p(x)$ and $q(x)$ is defined as

$$\delta_w(p||q) = p(w) log(p(w)/q(w)) \quad (2)$$

let LM^1_{bg} denote the unigram language model computed from background corpus, LM^1_{fg} denote the unigram language model computed from foreground corpus, LM^N_{bg} denote the N-gram language model computed from background corpus, LM^N_{fg} denote the N-gram language model computed from foreground corpus.

We can now quantify phraseness and informativeness as follows:

Phraseness of **W** is how much we lose information by assuming independence of each word by applying the unigram model, instead of the N-gram model. It is computed as

$$\delta_W(LM^N_{fg} || LM^1_{fg})$$

Informativeness of **W** is how much we lose information by assuming the phrase is drawn from the background model instead of the foreground model.

$$\delta_W(LM_{bg}^N||LM_{bg}^N)$$

or

$$\delta_W(LM_{bg}^1||LM_{bg}^1)$$

Combined: The following is considered to be a mixture of phraseness and informativeness.

$$\delta_W(LM_{fg}^N||LM_{bg}^1)$$

3.2 Applying to Books

In this section, we describe in detail, the process of applying the method to extraction of keyphrases from books. The steps are illustrated with an example in Table 2.

3.2.1 Step1 : Cleaning and Initialization

The text in each individual chapter is the *foreground corpus* under consideration. We use string concatenation of all the books in our collection as our *background corpus*.

As a first step, we clean both the background corpus and foreground corpus separately. We remove all punctuation marks and also remove stop words. The current approach focusses on english, hence, a stop word list is readily available from earlier efforts of various research in the community. For other languages too, it is possible to get a stop word list could be automatically generated from frequency analysis of words in a corpus of sufficient size. Highly frequent words can be removed.

Additionaly, we also convert all the words to lowercase in order to be able to capture the frequencies of the words appropriately.

3.2.2 Step2 : Extraction of Candidate Keyphrases

Once the cleaning of the text from foreground corpus is done, the method is to first extract a set of candidate keyphrases. These candidate keyphrases are then scored appropriately. Finally, the candidate keyphrases are pruned using the score and top few scores are taken and final keyphrases.

Extraction of candidate keyphrase extraction is an important step. In this step, first we extract all possible sequence of N words(phrases) and then eliminate all those which cannot be considered as candidates. Careful elimination of keyphrases has to be done in order not to eliminate important phrases.

We considered N to be 3 in this work. We generate all possible 3 word sequences from the foreground corpus. We then compute the frequency of occurence of each word sequence. From these, we eliminate those candidates whose frequency is greater than a threshold T. In order to be careful while eliminating the candidates, we use a simple elimination process and considered T to be 1 in our work. After the elimination process, we retain only those phrases in which none of the words are repeated. Let us consider that the N to be 3, then we only consider where each of the three words are different and eliminate those candidates where there is repitition of words. In this way, we get a set of candidate keyphrases.

3.2.3 Step3 : Scoring

We compute language models from *background corpus* and *foreground corpus*. Unigram, bigram and trigram language models are computed from the same. We used CMU SLM Toolkit [1].

let LM_{bg}^1 denote the unigram language model computed from background corpus, LM_{fg}^1 denote the unigram language model computed from foreground corpus, LM_{bg}^N denote the N-gram(where N can be 2 or 3) language model computed from background corpus, LM_{fg}^N denote the N-gram language model computed from foreground corpus.

Phraseness and informativeness notions are used to represent the importance of a keyphrase. A score is computed for each candidate keyphrase extracted in Step 3 as a simple sum of the phraseness measure and informativeness measure.

Phraseness φ_P of **W** is how much we lose information by assuming independence of each word by applying the unigram model, instead of the N-gram

[1] Available at http://www.speech.cs.cmu.edu/SLM_info.html

model. It is computed as

$$\varphi_P = \delta_W(LM_{fg}^N || LM_{fg}^1) \quad (3)$$

Informativeness φ_I of $\mathbf{W}$ is how much we lose information by assuming the phrase is drawn from the background model instead of the foreground model.

$$\varphi_I = \delta_W(LM_{bg}^1 || LM_{bg}^1) \quad (4)$$

The total score is a sum of both the scores.

$$\varphi_{total} = \varphi_P + \varphi_I \quad (5)$$

Another way to combined is to consider a mixture of phraseness and informativeness as follows which we have used in our work.

$$\delta_W(LM_{fg}^N || LM_{bg}^1)$$

3.2.4 Step4 : Pruning

Once scoring of candidate keyphrases is done, we now prune out certain keyphrases which are less probable to be important. The candidate keyphrases are sorted in descending order based on the score φ_{total} computed in Equation 5. We consider top K (We simply used K= 10 in our work) as the final set of keyphhrases. In this way, we extract keyphrases from books.

4 Discussion

We ran our approach on books available from Gutenberg Project [2]. Project Gutenberg is the first and largest single collection of free electronic books, or eBooks. Michael Hart, founder of Project Gutenberg, invented eBooks in 1971 and continues to inspire the creation of eBooks and related technologies today.

We extract kepphrases from each chapter of the book indepedently. We used the method described in Section 3.2. For each chapter, the text content is cleaned and language models are computed. Candidate keyphrases are extracted and then by scoring them appropriately and pruning keyphrases are extracted.

[2]Available at http://www.gutenberg.org

Table 4 shows some sample results. The results are from 2 books, the first one is a book called *Far from the Madding*. The second one is titled *Return of Sherlock Holmes, A collection of Holmes Adventures*. As the name suggests the book is a collection of some holmes adventures, each chapter describing one. The first chapter describes about an adventure of an empty house. The keyphrases extracted from this chapter are : { *man found lying, long street people,street holmes stopped, thin man coloured, remarkable man blame* }. As it can be seen, extracted phrases seem interesting and suggesting the topic suggested in the chapter. Some phrases like *man found lying, thin man coloured* give some hints about some incidents in the story. However, some phrases like *remarkable man blame* appear to be concatenation of words rather than a phrase though they convey some information from the story. This is probably because we have removed stop words which contained articles and some other joining words with which the phrase would have looked better.

Books contain a lot of text contained in it out of which not all words are important in understanding the key information. We call such words which are not important in understanding the key information as noise. Even though we removed stop words, the text is still not noise free. Word frequency analysis of the text of the book could help in pruning out the same.

We extracted keyphrases from technical documents using our approach. Table 1 shows the sample of results. The first document is about a web proxy logger and the second one discussed about automatic search engine evaluation with clickthrough data analysis. The keyphrases extracted from the first document very much convey some of the key points in the paper name {*client side logging,proxy log user,proxy web designers,server page forwards*}. Similarly for the second document, the keyphrases conveyed some of the key points in the paper. In Technical documents, the key points were emphasized in certain portions of the paper like abstract, introduction, proposed approach etc. Repitition of emphasis of key points in multiple parts of the paper has helped in capturing of the same appropriately by the approach. Also, since the content is compact it was less noise

prone.

5 Conclusions and Future Work

In this paper, we apply a language modeling technique to keyphrase extraction from books. We extracted all candidate keyphrases i.e., trigrams (consecutive three words) from books and examined if each word can be a keyphrase or not by scoring them appropriately and pruning them based on scores and other heuristics. We plan to further perform more robust evaluation and with more books. For extracting candidate keyphrases currently we extracted consecutive three words after stop word removal. It would be interesting to experiment with extracting consecutive three words skipping 1 or more words in between also as candidate keyphrases. This gives us more candidates to prune from giving us better chances of finding good keyphrases. Also we plan to extend this approach to extracting keyphrases from the whole book and using noise reduction techniques to filter noise.

References

[1] Fred J. Damerau. Generating and evaluating domain-oriented multi-word terms from texts. *Information Processing and Management*, 29(4):433–447, 1993.

[2] S.T Dumais, J Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the 7th international conference on information and knowledge management*, page 148155. ACM Press, 1998.

[3] Min Song, Il-Yeol Song, and Xiaohua Hu. Kpspotter: a flexible information gain-based keyphrase extraction system. In *WIDM '03: Proceedings of the 5th ACM international workshop on Web information and data management*, pages 50–53, New York, NY, USA, 2003. ACM Press.

[4] Takashi Tomokiyo and Mathew Hurst. A language modeling approach to keyphrase extraction. In *Proceedings of the ACL 2003 workshop on Multiword expressions*, pages 33–40, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[5] P.D. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, 2006.

[6] I.H. Witten, G.W. Paynter, E. Frank, C. Gutwin, and C.G Nevill-Manning. Kea: Practical automatic keyphrase extraction. In E. A. Fox and N. Rowe, editors, *Proceedings of digital libraries 99: The fourth ACM conference on digital libraries*, pages 254–255. ACM Press, 1999.

[7] Mikio Yamamoto and Kenneth W. Church. Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus. *Computational Linguistics*, 27(1):1–30, 2001.

Step	**Example**
Sentence	Topics are also used to construct user profiles via explicit specification of interests or automatic analysis of Web pages visited
1.	topics construct user profiles explicit specification interests automatic analysis web pages visited
2.	{analysis web pages, automatic analysis web, construct user profiles, explicit specification interests, interests automatic analysis, profiles explicit specification, specification interests automatic, topics construct user, user profiles explicit, web pages visited }
3.	profiles explicit specification : 0.0281 explicit specification interests : 0.0281 specification interests automatic : 0.0272 user profiles explicit : 0.0260 construct user profiles : 0.0260 interests automatic analysis : 0.0255 topics construct user : 0.0243 automatic analysis web : 0.0227 web pages visited : 0.0226 analysis web pages : 0.0217
4.	profiles explicit specification explicit specification interests specification interests automatic user profiles explicit construct user profiles interests automatic analysis topics construct user automatic analysis web web pages visited analysis web pages

Table 2: Keyphrase extraction from books:Steps with an example

Book Name	**Far from the Madding Crowd**
Chapter 1	*Description of Farmer Oak –An Incident*
Extracted Keywords	watch oak remedied, oak looked disputants, gabriel oak regarded, looked gabriel minute, oak smiled corner
Chapter 2	NIght–the Flock–an Interior- -Another Interior
Extracted Keyword	oak plantation pushed, till oak withdrew, stood sheep crook, small sheep farm, hill clear midnight
Chapter 3	*A Girl on Horseback –Conversation*
Extracted Keywords	hut gabriel fire, hand oak held, oak began wiping, oak plantation lingering, gabriel oak miss
Book Name	**THE RETURN OF SHERLOCK HOLMES, A Collection of Holmes Adventures**
Chapter 1	*The Adventure of The Empty House*
Extracted Keywords	man found lying, long street people, street holmes stopped, thin man coloured, remarkable man blame
Chapter 2	*The Adventure of The norwood builder*
Extracted Keywords	holmes mr mcfarlane, holmes watson kindness, holmes papers removed, lestrade time remain, mcfarlane lestrade unfortunate
Chapter 3	*The Adventure of the Dancing men*
Extracted Keywords	holmes inspector earnestly, inspector holmes remarked, mr cubitt plan, holmes eyes forbid, holmes wife frightening

Table 3: Keyphrase Extraction from Book chapters