Identifying Keywords and Key Phrases

Ashwini Madane

Abstract*- Keywords and key phrases are widely used in large document collections. They describe the content of single documents and provide a kind of semantic metadata that is useful for a variety of purposes. Text mining is powerful tool to find useful and needed information from huge data set. For context based text mining, key phrases are used. Key phrases provide brief summary about the contents of documents. In document clustering, number of total cluster is not known in advance. In K-means, if prespecified number of clusters modified, the precision of each result is also modified. Therefore Kea, is algorithm for automatically extracting key phrases from text is used. In this kea algorithm, number of clusters is automatically determined by using extracted key phrases. Keameans clustering algorithm provide easy and efficient way to extract test document from large quantity of resources. Key phrase play important role in text indexing, summarization and categorization. Key phrases are selected manually. Assigning key phrases manually is tedious process that requires knowledge of subject. Therefore automatic extraction techniques are most useful.*

Keywords: Text mining, Key phrase extraction, key phrase.

I. INTRODUCTION

Text mining can be described as the process of identifying novel information from a collection of texts. By novel information we mean associations, hypothesis that are not explicitly present in the text source being analyzed. For example, suppose that a document establishes a relationship between topics A and B and another document establishes a relationship between topics B and C. These two documents jointly establish the possibility of a novel (because no document explicitly relates A and C) relationship between A and C.

II RELATED WORK

Automatic summarization involves reducing a text document or a larger corpus of multiple documents into a short set of words or paragraph that conveys the main meaning of the text. Two methods for automatic text summarization they are Extractive and Abstractive. Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary. In contrast, abstractive methods build an internal semantic representation and then use natural language generation techniques to create a summary that is closer to what a human might generate.

Manual selection of key phrases from a document by a human is not a random act. Key phrase extraction is a task related to the human cognition. Hence, automatic key phrase extraction is not a trivial task and it needs to automate due to its usability in managing information overload on the web.

III PROPOSED ARCHITECTURE

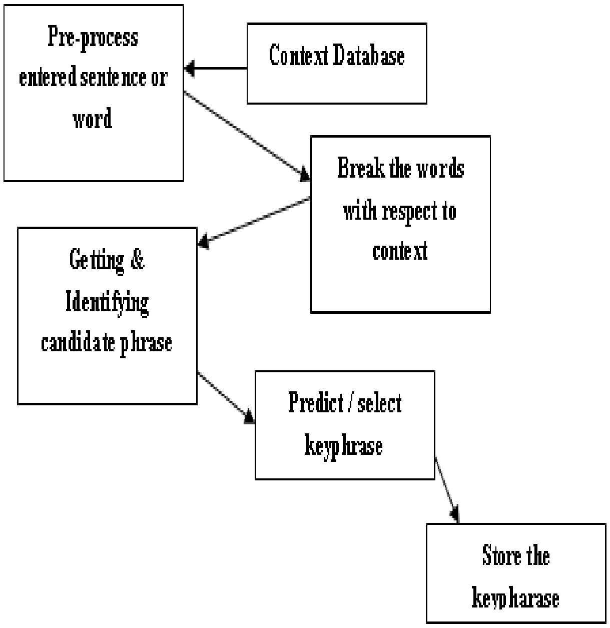
Pre-process entered sentence or word
Context Database
Break the words with respect to context
Getting & Identifying candidate phrase
Predict / select keyphrase
Store the keypharase

Figure1: Key phrase Extraction architecture.

A document is represented as a graph in which the nodes represent terms, and the edges represent the co-occurrence of terms. Whether a term is a keyword is determined by measuring its contribution to the tree. Key phrases are extracted from candidate phrases based on examination of their features. Key phrases provide a brief summary of a document's contents. As large document collections such as digital libraries become widespread, the value of such summary information increases.

IV KEYPHRASE EXTRACTION

This approach does not use a controlled vocabulary, but instead chooses key phrases from the text itself. It employs lexical and information retrieval techniques to extract phrases from the document text that are likely to characterize it [5]. Key phrase extraction consists of two steps: candidate phrase identification and key phrase selection. In this approach, the training data is used to tune the parameters of the extraction algorithm.

Manuscript received on July, 2012.

Ms Ashwini.c. Madane, BE .MTech (pursing), BharatiVidyappeth University, College of Engineering, Pune – 43.

STEP1: PREPROCESSING:

ASCII input files are filtered to regularize the text and determine initial phrase boundaries. The input stream is split into tokens (sequences of letters, digits and internal periods), and then several modifications are made:

- Punctuation marks, brackets, and numbers are replaced by phrase boundaries;
- Apostrophes are removed;
- Hyphenated words are split in two;
- Remaining non-token characters are deleted, as are any tokens that do not contain letters.

STEP2 CANDIDATE IDENTIFICATION

Kea then considers all the subsequences in each line and determines which of these suitable candidate phrases are. Following rules are applied for the identification of phrases:

- Candidate phrases are limited to a certain maximum length (usually three words);
- Candidate phrases cannot be proper names (i.e. single words that only ever appear with an initial capital);
- Candidate phrases cannot begin or end with a stop word;

The stop word list contains 425 words in nine syntactic classes (conjunctions, articles, particles, prepositions, pronouns, anomalous verbs, adjectives, and adverbs). All contiguous sequences of words in each input line are tested using the three rules above, yielding a set of candidate phrases. Note that sub phrases are often candidates themselves. Thus, for example, a line that reads the programming by demonstration method will generate programming, demonstration, method, programming by demonstration, demonstration method, and programming by demonstration method as candidate phrases, because the and by are on the stop word list.

STEP 3: DETERMINING CANDIDATE PHRASES

The final step in determining candidate phrases is to case fold all words, and stems them using the iterated Lovins method. This involves using the classic Lovins stemmer to discard any suffix, and repeating the process on the stem that remains until there is no further change. So, for example, the phrase cut elimination becomes cut slim. Stemming and case-folding allow us to treat different variations on a phrase as the same thing. For example, proof net and proof nets are essentially the same, but without stemming they would have to be treated as different phrases. In addition, Kea uses the stemmed versions to compare.

STEP 4: FEATURE CALCULATION

Document frequency –

The number of documents containing the phrase in some large corpus. A phrase's document frequency indicates how common it is (and rarer phrases are more likely to be key phrases). Kea builds a document frequency file for this purpose using a corpus of about 100 documents. The document frequency file stores each phrase and a count of the number of documents in which it appears. With this file in hand, the TF × IDF for phrase P in document D is:

$$\text{TF-IDF} = \frac{FREQ\,(P,D)}{SIZE\ (D)}\ \text{X}\ (-\ \text{LOG}_2 \frac{df\,(P)}{N})$$

Where,
freq(P,D) is the number of times P occurs in D
size(D) is the number of words in D
df(P) is the number of documents containing P in the document
N is the size of the document.
First occurrence

The second feature, first occurrence, is calculated as the number of words that precede the phrase's first appearance, divided by the number of words in the document. The result is a number between 0 and 1 that represents how much of the document precedes the phrase's first appearance.

IV. CONCLUSION:

Keyphrases have long been recognized as good surrogates for documents and concepts and are used extensively in browsing and query reformulation in information retrieval systems to improve user satisfaction in retrieved results. In order to allow for multiple objectives over a corpus of documents.

V. REFERENCES:

[1] TEXT MINING USING KEYPHRASE EXTRACTION, Shobha S. Raskar, BharatiVidyapeeth University, College of Engineering BVUCOE ,Dhankawadi,Pune, D. M. ThakoreBharatiVidyapeeth University, College of Engineering BVUCOE, Dhankawadi ,Pune.

[2] National Library of Medicine, Unified Medical Language System, sixth experimental edition. Bethesda, MD, 1995.

[3] B., Krulwich, & C., Burkey, "The Infofinder Agent-Learning User Interests through Heuristic Phrase Extraction," IEEE Intelligent Systems & Their Applications, 12(5), 1997, pp. 22 - 27.

[4] P.D. Turney,"Learning Algorithms for Keyphrase Extraction," Information Retrieval, 2, 303-336, 2000.

[5] I. Witten, E. Frank, G. W. Paynter, "Doman-Specific Keyphrase Extraction," Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, 1999, pp. 668-673.

[6] A. Bookstein, S.T. Klein, T. Raita, "Clumping Properties of Content-Bearing Words," Journal of the American Society for Information Science, 49(2), 1998, pp. 102-114.

[7] J. Picard, "Finding content-bearing terms using term similarity," Proceedings of Ninth Conference of the European Chapter of the Association for Computational Linguistics, Bergen, Norway, 1999.

[8] C. Nevill-Manning, I. Witten and G. Paynter, "Browsing in Digital Libraries: a Phrase-Based Approach," Proceedings of 1997 International Conference on Digital Libraries, 1997, pp. 230-236.

[9] S. Sekine and R. Grishman, "A Corpus-based Probabilistic Grammar with only Two Non-terminals," Fourth International Workshop on Parsing Technologies, Prague, 1995.

[10] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, "A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multiobjective Optimization: NSGA-II," Proceedings of the Parallel Problem Solving from Nature VI (PPSN-VI), 2000, pp. 849-858.

[11] A. Messac and W. Chen, "The Engineering Design Descipline: Is its Confounding Lexicon Hindering its Evolution?" Proceedings of DETC' 98, DTM-5658, 1998.