A new approach for marking for mined text for perfect navigation in data hierarchical design

[1]Adireddiprasannakumarprasanna.Emailid: adireddi@gmail.com
Visakha institute of Engineering

Title justification:

Putting markers at the searched starting word in a new approach in the document to navigate for the other preceding terms in the hierarchical design patterns.

Abstract:

For any given input (normally search string) data the data will be the prefix data for any search criteria. So the document will be marked first using marking technique. Here the markers will be logged for further search criteria for pre or post term (Ex: Search string is java struts 1.0, so the terms are “java” “struts” so search string could be “java struts” or “struts java” “ java struts 1.0”). For this given string with respect to the first term markers will be placed first using marking technique. Once the markers are placed the document is navigated to get the fully qualified sentences with respect to pre/post terms.

With respect to pre and post terms the sentences with start term as search sentence terms with possible orders.

(Index Terms: Text mining, navigation, marking, Information extraction)

Introduction:

The problem of text mining, i.e. discovering useful knowledge from unstructured or semi-structured text, is attracting increasing attention. This paper suggests a new framework for text mining based on the integration of Information Extraction (IE) and Knowledge Discovery from Databases (KDD), a.k.a. data mining. KDD and IE are both topics of significant recent interest. KDD considers the application of statistical and machine-learning methods to discover novel relationships in large relational databases. IE concerns locating specific pieces of data in natural-language documents, thereby extracting structured information from free text. However, there has been little if any research exploring the interaction between these two important areas. In this paper, we explore the mutual benefit that the integration of IE and KDD for text mining can provide.

Traditional data mining assumes that the information to be “mined” is already in the form of a relational database. Unfortunately, for many applications, electronic information is only available in the form of free natural-language documents rather than structured databases. Since IE addresses the problem of transforming a corpus of textual documents into a more structured database, the database

constructed by an IE module can be provided to the KDD module for further mining of knowledge as illustrated in Figure 1. Information extraction can play an obvious role in text mining as illustrated. Although constructing an IE system is a difficult task, there has been significant recent progress in Using machine learning methods to help automate the construction of IE systems [5, 7, 9, 23]. By manually annotating a small number of documents with the information to be extracted, a Reasonably accurate IE system can be induced from this labelled corpus and then applied to a large corpus of text to construct a database. However, the accuracy of current IE systems is limited and therefore an automatically extracted database will inevitably contain significant numbers of errors. An important question is whether the knowledge discovered from this "noisy" database is significantly less reliable than knowledge discovered from a cleaner database. This paper presents experiments showing that rules discovered from an automatically extracted database are close in accuracy to that discovered from a manually constructed database. A less obvious interaction is the benefit that KDD can in turn provide to IE. The predictive relationships between different slot fillers discovered by KDD can provide additional clues about what information should be extracted from a document. For example, suppose we discovered that computer-science jobs requiring "MySQL" skills are "database" jobs in many cases. If the IE system manages to locate "MySQL" in the language slot but failed to extract "database" in the area slot, we may want to assume there was an extraction error. Since typically the recall (percentage of correct slot fillers extracted) of an IE system is significantly lower than its precision (percentage of extracted slot fillers which are correct) [13], such predictive relationships might be productively used to improve recall by suggesting additional information to extract. This paper reports experiments in the computer-related job-posting domain demonstrating that predictive rules acquired by applying KDD to an extracted database can be used to improve the recall of information extraction.

Modules:

Document
Pre-processing (cleaning/stemming).
Search terms evolution.
Marking
Pre/post terms with respect to main term positions.

Modules explanation:

- **Document**
- **Pre-processing (cleaning/stemming):** The selected document(s) will be cleaned and stemmed with stop words removal and extra/special character terms.
- **Search terms evolution:** The input sentence will be tokenized with respect to seed term and other items will be prioritized.
- **Marking:** The document will be marked with respect to seed term and vector will be

generated for the marked items index positions.

- **Pre/post terms with respect to main term positions :**
  At the marked position and with respect to pre post items sentences with starting word in the preceding words of search strings will be framed in vector to get the best patterns.

Literature survey

The increasing volume of data in modern business and science calls for more complex and sophisticated tools. Although advances in data mining technology have made extensive data collection much easier, it is still always evolving and there is a constant need for new techniques and tools that can help us transform this data into useful information and knowledge.

Various steps in data mining involved:

- Preprocessing
- Clustering
- Stemming
- Cleaning
- Categorization
- Organizing

Introduction:

Data will be organized in the proper format after pre-processing, cleaning, stemming and stop words removal. Once the cleaned document is ready with respect to input term markers are placed in the document. Once the markers are placed the positions will be in once particular structure. The structure contains offsets all markers in that particular single document.

With respect to second term the positions will given weightage. Once the weightage is put up for the sentences the effective output from mining will be revealed. Here the output could be swapped positions also.

Ex: Input: "java struts framework"

Output:

- "java struts framework"
- "struts framework"
- "struts java framework"

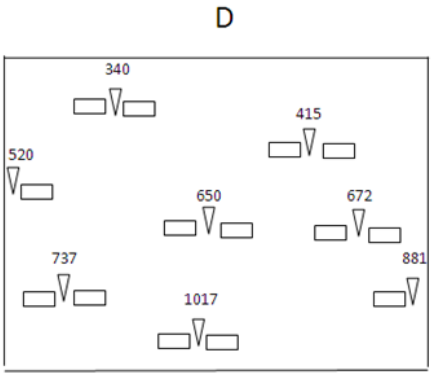
Disadvantages of existing system:

- Look for the data in query results, so big data can be revealed for processing.
- The hierarchies on fixed concepts.
- Navigation for mining in hierarchical organized data.

Advantages of proposed system:

- Data documents are general so frequency of outputs is more.
- Marking with the seed term is less complex for further processing.
- Pre/post terms with respect to main term positions is very handy to retrieve the best pattern output.

Algorithm:



1	340
2	415
3	520
4	650
5	672
6	737
7	881
8	1017

Initialization

$\sum D$ ←document

t_1← term1

t_2←term2

$\sum PV$ ←position vector

$\sum D_M$ ←marked document

For each t in D

n=0

Loop start

N=n+1

If t==t_1

D_m←mark (D)

PV← n

End loop

End for

Pre post algorithm

Tm← vector

For each P in Dm

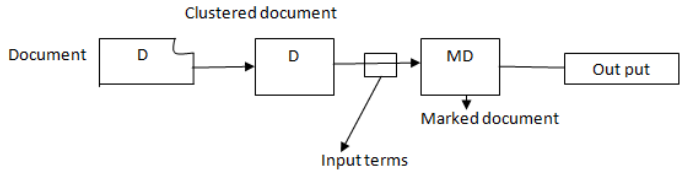
If Dm (p+1) ==t2 or Dm(P-1) == t2

Tm←Dm(p+1)

End if

End for

Architecture:



Metrics:

Software requirements:

Language: Jdk1.6

- **Tool IDE**(integrated development environment): Netbeans8/6
- **Technology UI**: Java Swings

Hardware requirements:

- **Processor**: Pentium-3
- **Speed**: 1.1 Ghz
- **Ram**: 256mb minimum

CONCLUSION:

For almost any granted suggestions (normally search string) files the info stands out as the prefix files for any search requirements. To ensure the report will likely be notable first employing marking approach. Here the guns will likely be logged for more search requirements for pre or maybe article term (Ex lover: Research sequence will be cappuccino struts 1. 0, and so the conditions

are usually “java” “struts” consequently search sequence could possibly be “java struts” or maybe “struts java” “ java struts 1. 0”). With this granted sequence with regards to the first term guns will likely be located first employing marking approach. In the event the guns they fit the report will be navigated to have the entirely skilled sentences with respect to pre/post conditions.

References:

[1] M. W. Berry, editor. Proceedings of the Third SIAM International Conference on Data Mining(SDM-2003) Workshop on Text Mining, San Francisco, CA, May 2003.

[2] S. Basu, R. J. Mooney, K. V. Pasupuleti, and J. Ghosh. Evaluating the novelty of text-mined rules using lexical knowledge. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001), pages 233–239, San Francisco, CA, 2001.

[3] F. Ciravegna and N. Kushmerick, editors. Papers from the 14th European Conference on Machine Learning(ECML-2003) and the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases(PKDD-2003) Workshop on Adaptive Text Extraction and Mining, Cavtat-Dubrovnik, Croatia, Sept. 2003.

[4] C. Cardie and R. J. Mooney. Machine learning and natural language (Introduction to special issue on natural language learning). Machine Learning, 34:5–9, 1999.

[5] R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval. ACM Press, New York, 1999.

[6] M. E. Califf, editor. Papers from the Sixteenth National Conference on Artificial Intelligence (AAAI-99) Workshop on Machine Learning for Information Extraction, Orlando, FL, 1999. AAAI Press.

[7] W. W. Cohen. Fast effective rule induction. In Proceedings of the Twelfth International Conference on Machine Learning (ICML-95), pages 115–123, San Francisco, CA, 1995.

[8] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proceedings of the 20th International Conference on Very Large Databases (VLDB-94), pages 487–499, Santiago, Chile, Sept. 1994.

[9] M. E. Califf and R. J. Mooney. Relational learning of pattern-match rules for information extraction. In Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99), pages 328–334, Orlando, FL, July 1999.

[10] C. Cardie. Empirical methods in information extraction. AI Magazine, 18(4):65–79, 1997.