
Data-Mining of Medline Abstracts Revisited

Instructors

Professor Orlando Karam

Professor Venu Dasigi

Patrick Bobbie

Students

Stanley Iriele

Abstract

- MEDLINE is a database containing citations and abstracts of most health-related publications in the USA. The amount of data is big (around 12 million documents), which poses interesting database problems. we are experimenting with using Z-scores for selecting words useful for clustering concepts. These concepts may be genes, proteins, drugs or any other biologically meaningful terms

Introduction

- SAX tool/Stemmer
 - Analysis
 - Observations
 - Additions
 - Modifications
 - Stemmer

Analysis-Extended Parsing

- Inconsistent Reporting of publish date
- The random reporting of corrections makes it hard to report them
- Reporting the authors with the other information
- Reporting Volume/Issue number and medline TA and other things.
- Program used too much heap memory

Observations/Analysis

- 1 character words were being stored
- Regular stemming algorithms have no effect on medical words
- Using diGrams for stemming is the best alternative so far

Additions

- Date Class
 - Creates a Date Object
- Word filter
 - Added to Document Class.
 - Stops frequently used English words.
- Future Additions: improved file manager
 - Improving the stop word file

Additions (continued)

- Author Class
 - Creates a Author Object
 - Handles author list problem
- Added files
 - ArticleINFO
 - Comments
 - Authors

Modifications

- Heap memory
 - Combined two hash maps
 - Combined abstract words
 - Modified regex pattern and added secondary filter.
- Extended parsing
 - revised date, published date, Page number, MedlineTA

Modifications (continued)

- Event handler class
 - Extended to include Volume/Issue
 - Parsed MedlineTA, publish date, Date revised
 - Handles corrections
- Document
 - Extended to handle changes with event Handler

Example of Digrams Similarity

- Word 1:Statistics
 - Digrams: st ta at ti is st ti ic cs
 - Unique digrams:at cs ic is st ta ti(7)
- Word 2:Statistical
 - Digrams: st ta at ti is st ti ic ca al
 - Unique digrams: al at ca ic is st ta ti(8)
- Shared unique digrams at ic is st ta ti (6)
 - Function $(2 * \text{shared digrams}) / (\text{unique in 1} + \text{unique in 2})$

Additions to handle stemming

- WordClass
 - HashSet unique digrams
 - Int uniqueDigrams
 - Function getDigrams()
- Standard word list file
 - Entered the alphabet and all single digit numbers

Modifications to handle stemming

- Wordlist class
 - Added a digram similarity function
- Save Method in wordList
 - Function to calculate similarity between words using digrams

Stemmer 1.0

- The document words are iterated through and the similarity matrix is comprised of the actual matrix
- Similarity ratio was set to: .6
- File size upon completion: 18 gigabytes
- Time : 8+ hours
- Size experimented on: 0.25%
- Result: unacceptable performance

Stemmer 1.1

- The document words are iterated through and the similarity matrix is comprised the word and its stem
- Similarity ratio was set to: .6
- File size upon completion: 50 MB
- Time : 3+ hours
- Size experimented on: 0.25%

- Unforgivable performance

Assessments/Conclusion

- The stemmer in any form demands far too much time and should be done in alternate language
- A potential file analyzer could be added to improve the word filter

References

- Text: "Professional XML Databases, Kevin Williams, Programmer to Programmer
- Text: "Java and XML", Bret McLaughlin , O'Reilly
- <http://java.sun.com/docs/books/tutorial/essential/regex/>