Software Requirements Specification

for

.txtKnot

Text Clustering based concept hierarchy to generalize from different text sources

Version 1.0 approved

Prepared by PI-47

Jayasinghe D.S. – 060199T
Ketteepearachchi D.C. – 060235D
Abeywickrama G.P.S.P – 060012R
Hettiarachchi S. – 060168A

Department of Computer Science & Engineering

University of Moratuwa

31/07/2009

Table of Contents

1. Introduction 1
1.1 Purpose 1
1.2 Intended Audience 1
1.3 Project Scope 1
1.4 References 1
2. Overall Description 2
2.1 Product Perspective 2
2.2 Product Features 2
2.3 User Classes and Characteristics 2
2.4 Operating Environment 2
2.5 Design and Implementation Constraints 2
2.6 User Documentation 2
2.7 Assumptions and Dependencies 3
3. System Features 3
3.1 Specifying the source for clustering text and display source information 3
3.2 Creates a Meaningful Hierarchy of Text Documents 4
3.2.1 Description & Priority 4
3.2.2 Stimulus/Response Sequence 4
3.2.3 Functional Requirements 4
3.3 Visually Represent the Hierarchy in a Tree Structure 5
3.3.1 Description & Priority 5
3.3.2 Stimulus/Response Sequence 5
3.3.3 Functional Requirements 5
3.4 Category Search 5
3.4.1 Description & Priority 5
3.4.2 Stimulus/Response Sequence 5
3.4.3 Functional Requirements 6
3.5 Training the Clustering Process 6
3.5.1 Description & Priority 6
3.5.2 Stimulus/Response Sequence 6
3.5.3 Functional Requirements 6
4. External Interface Requirements 6
4.1 User Interfaces 6
4.1.1 Input type selection interface. 6
4.1.2 Interface with a virtual tree hierarchy with search ability. 6
4.2 Software Interfaces 7
4.2.1 Interface to the operating system. 7
4.2.2 Interface to the database system. 7
4.2.3 Interface to other useful software. 7
4.3 Communications Interfaces 7
5. Other Nonfunctional Requirements 7
5.1 Performance Requirements 7
5.2 Safety Requirements 8
5.3 Security Requirements 8
5.4 Software Quality Attributes 8
6. Other Requirements 8

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

".txtKnot" is a complete, standalone software system which is capable of transforming a large set of unsorted text documents in to a meaningful hierarchy of text documents. This transformation has the potential to increase the efficiency of data analyzing and decision making exponentially. As almost all the organizations in the modern era are heavily dependent on decision making processes based on electronically stored data, this system could greatly aid in the increase of efficiency and effectiveness of those organizations.

1.2 Intended Audience

- Supervisors and Coordinators
- Designers and Developers
- Contributors
- Software testers, validation and quality assurance persons
- Software evaluation teams
- Support staff (including technical writers)
- Help and Assistance teams
- Maintenance persons
- Clients(s) who adopt this software
- Users of this software

1.3 Project Scope

This system is intended to provide following outcomes,

- Provide an efficient tool for data analyzing
- Make decision making process more informed
- Maximize the utilization of breakthroughs made in text clustering related research
- Provide an easy to use interface to the user

1.4 References

1A. Klose, A. N¨urnberger, R. Krus and G. Hartmann, M. Richards, Interactive Text Retrieval Based on Document Similarities, Phys. Chem. Earth, Vol. 25, No. 8, pp. 649–654, 2000
2 Kohonen, T., Hynninen, J., Kangas, J., and Laaksonen, J. (1996a). SOM_PAK: The selforganizing map program package. Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo.
3 Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. Biological Cybernetics, 43:59-69.
4 Alahakoon, D., Halgamuge, S., and Srinivasan, B., A structure adapting feature map for

optimal cluster representation, in Proc. Int. Conf. On Neural Information Processing, pp. 809–812, Kitakyushu, Japan, 1998.

2. Overall Description

2.1 Product Perspective

".txtKnot" is a standalone product that could be used to convert a set of seemingly unrelated text documents in to a meaningful hierarchy of text documents. It uses the method of text clustering to deliver the aforementioned primary functionality. Even though the concept of text clustering is not entirely new, there are algorithms such as Growing Self Organizing Maps of which, the true potential is not yet fully utilized. This system provides a powerful and effective tool for data mining and data analyzing by utilizing the latest innovations in the field of text clustering.

2.2 Product Features

- Specifying the source for clustering text and display source information
- Creates a Meaningful Hierarchy of Text Documents
- Visually Represent the Hierarchy in a Tree Structure
- Category Search
- Training the Clustering Process

2.3 User Classes and Characteristics

- System End Users
  All the users of the system falls in to this class as all the users have same access level.

2.4 Operating Environment

- System should be able to operate on an IBM compatible computer with a 2.4GHz Intel Core 2 Duo processor and 2 GB random access memory.
- System should be able to operate in Windows XP and later versions.
- System requires .NET framework version 3.5 to operate properly.

2.5 Design and Implementation Constraints

None

2.6 User Documentation

- A comprehensive user manual in .pdf format should be provided.
- An online help facility should be provided for further clarification.

2.7 Assumptions and Dependencies

Successful completion of the system depends on the availability of the text clustering algorithm implementations.

3. System Features

3.1 Specifying the source for clustering text and display source information

3.1.1 Description and Priority

User should be facilitated to specify the source where the software should collect data for clustering. The source can be located in the file system or in a database. When the user selects which type of source s/he is going to obtain data then detail specifications of the collection of documents or data should be asked. After user has specified the location of the source then source information should be displayed. For example when user specified a directory in the file system then all readable text document list should be displayed.

3.1.2 Stimulus/Response Sequences

1	Stimulus	User requests to set the input source as a collection of documents stored in the file system
	Response	System should ask the location of folder where it should collect documents for clustering.
2	Stimulus	User request input source location type as file system and then specified the location of the folder where all documents for clustering resides
	Response	All readable files in the selected directory should be list down with their information such as file name, file size, file type, etc.
3	Stimulus	User request to set the input source as a database
	Response	System should ask the type of the database, connection string, table where information for clustering is stored and the columns for reference and data. After user has successfully specified these details then all rows in the selected table should be displayed.

3.1.3 Functional Requirements

When selecting a folder for obtaining text documents system should ask what kind of file types that should be added to the list and whether all files in the sub folders should also be added. System should support following document types

- Text Documents
- Rich Text Documents
- Office Documents
- Page Downloaded Files (PDFs)

System should facilitate to change the list of selected file list. i.e. it should provides add more files to the list and remove files from the list.
When user specified a database to read the software should support all popular database types such as

- Microsoft SQL Server
- MySQL
- Oracle

3.2 Creates a Meaningful Hierarchy of Text Documents

3.2.1 Description & Priority

This feature takes the set of unsorted text documents that were given as input and creates a meaningful hierarchy of nodes that contains those documents, which represents the inter-connections of the text documents in the given document set. The priority for implementing this feature should be higher than implementing others.

3.2.2 Stimulus/Response Sequence

1	Stimulus	User has successfully added information on document collection for clustering and gives command to generate the hierarchy of documents.
	Response	Creates a meaningful hierarchy of nodes that contains contextually similar documents.
2	Stimulus	User gives the command to generate the hierarchy for an empty document collection
	Response	System should display an error message dialog box saying that no input data is given.

3.2.3 Functional Requirements

- A valid text document set should be specified.
- For the clustering purpose, we might use a part of the text document since it would consume a huge amount of computational power and memory if we use the whole document. Therefore we may take the introduction or the table of content of a text document and abstract or the introduction of a research paper to build up the input. Here we give the flexibility to the user by making this fact also a parameter of the system.
- Text clustering algorithm should be trained prior to generate the output.

3.3 Visually Represent the Hierarchy in a Tree Structure

3.3.1 Description & Priority

After the system has generated the meaningful hierarchy it should be displayed as a tree structure where each node represents a category. When user clicks a leaf node all documents which are contextually similar should be displayed and when user click that document it should be viewable. The priority for implementing this feature is also high.

3.3.2 Stimulus/Response Sequence

1	Stimulus	Create a document hierarchy
	Response	Display the hierarchy in a collapsed tree structure
2	Stimulus	User clicks a node in the tree view
	Response	Display child nodes of the selected node
3	Stimulus	User clicks a leaf node in the tree view
	Response	Display all documents belongs to the selected node.
4	Stimulus	User clicks a document that is shown in the list view which is viewed when a leaf node in the tree view is selected.
	Response	Opens the document using default viewing application Lists the most related set of documents contained in adjacent nodes in a separate pane as suggestions.

3.3.3 Functional Requirements

To display the hierarchy first a document hierarchy should be generated. If the process of creating document hierarchy fails then a suitable error message dialog box should be displayed.
Number of suggested documents shall not exceed 10.

3.4 Category Search

3.4.1 Description & Priority

In this feature user should be facilitated searching categories that are generated in the tree view. A search text box is given to search a category whenever such a requirement comes. Furthermore top most categories in the tree view are displayed in a drop down list. User should able to search sub categories of the selected top most category via that drop down list.

3.4.2 Stimulus/Response Sequence

1	Stimulus	Select a category from the drop down list, type a key word and click the “Search” button.

	Response	All the articles containing the specified key word within the specified category should be listed.

3.4.3 Functional Requirements

A valid document hierarchy should exist.
If the number of results for a given key word in a specific category a message should be displayed to the user. Suggestions or alternative key words should be listed in a separate pane.

3.5 Training the Clustering Process

3.5.1 Description & Priority

This feature provides the users with the ability to train the clustering process with their own document set. Option of training incrementally or training from the beginning is given to the user.

3.5.2 Stimulus/Response Sequence

1	Stimulus	A set of text documents is provided as the training document set. Method of training (whether incremental or from the beginning) is specified.
	Response	Text clustering algorithm is trained according to the newly provided document set.

3.5.3 Functional Requirements

A valid set of text documents should be provided as the input for the training procedure. Minimum of 50 text documents should be provided as the input.

4. External Interface Requirements

4.1 User Interfaces

4.1.1 Input type selection interface.

In the context of user interfaces, there should be an interface for the user to input data to the system. There are many input types for this system such as text documents, web documents and text files stored in databases. Basically, the user should be prompted to select the method which he is going to input data to the system through this interface.

4.1.2 Interface with a virtual tree hierarchy with search ability.

Once the clustering process is done by the system for a given set of documents, it should provide a way to observe results for the user. That facility is provided through this interface. In further details, system should produce a visual representation of tree based hierarchy which represent document clusters. Here main categories of the given set of documents come in a drop down list. Then user can select a certain category which he is interested. User should be able to search a particular keyword also. For that purpose a text box and a button named "Search" also included in this interface. After the searching, user should be provided relevant set of the nodes in that virtual tree based hierarchy. Simply, if the user selects a particular node on this virtual tree it should point out the set of documents it represents and the user should be able to read a particular document he wants by simply clicking on it. These are the basic requirements of this user interface.

4.2 Software Interfaces

4.2.1 Interface to the operating system.

There should be an interface to the operating system on which this system is supposed to be run. This system is mainly expected to be run on Windows XP and Windows Vista. So there should be a software interface which enables the above requirement.

4.2.2 Interface to the database system

Since this system can take text files which are stored in a database system as inputs, there should be an interface to interact with the databases as well. This interface should support database systems like Ms SQL, My SQL, and Oracle.

4.2.3 Interface to other useful software.

Text documents come in many flavors like .doc, .docx, .pdf etc. So this system should have an interface to deal with Ms Word 2007, Ms Word 2003 and Open Office. In order to read documents in .pdf format system should have another interface to Acrobat Reader.

4.3 Communications Interfaces

This system should be able to communicate with the external world through internet. Web pages would be another type of text document which will be fed in to this system. An interface is required to communicate with web browsers like Internet Explorer, Mozilla – Firefox. Here the communication standard would definitely be HTTP.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system shall be able to process a minimum of 10,000 documents at a single clustering process in a computer system with 2.4GHz Intel Core 2 Duo processor and 2 GB random access memory.

- Training of the system shall take no longer than 1 minute to process a training document set of 500 documents.
- Clustering and generating the hierarchy for 10,000 documents shall be finished in no longer than 10 minutes
- The system shall take no longer than 5 seconds to produce results for any search operation.

5.2 Safety Requirements

No safety requirements have been identified.

5.3 Security Requirements

- The system shall be provided with ‘Fail Safe’ features to withstand any unexpected shut downs or power failures.
- No changes or corruptions shall be done to the original documents by the system during the process.
- When the system derives data from a database, it shall not interfere with other processes that might be running and the database shall be in its original condition after being used by “.txtKnot”.
- When communicating with the databases, and when information is stored in configuration files, the system shall use encrypted and secured Connection strings.

5.4 Software Quality Attributes

- The system shall produce the output hierarchy with a 99.9% accuracy level depending on its training. i.e. “.txtKnot” shall not categorize two completely non-relevant documents in a single category where as in the visual map; documents with similar content shall be placed nearby.

6. Other Requirements

None