INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

# SIDE INFORMATION GENERATION BASED DATA CLUSTERING USING DETECTION SYSTEM METHODOLOGY

**RAJESH A. ROY[1], PROF. P. S. KULKARNI[2]**

Department of Computer Science &Engineering, RCERT, Chandrapur.

**Accepted Date: 15/03/2016; Published Date: 01/05/2016**

**Abstract***:* Side-information is available along with the text documents. Such side-information may be of different kinds, such as document provenance information, the links in the document, user-access behavior from web logs.Other non-textual attributes which are embedded into the text document. Such attributes may contain a tremendous amount of information for clustering purposes. However, the relative importance of this side-information may be difficult to estimate, especially when some of the information is noisy. In such cases, it can be risky to incorporate side-information into the mining process, it can either improve the quality of the representation for the mining process, or can add noise to the process. Therefore, we need a principled way to perform the mining process, so as to maximize the advantages from using this side information.

**Keywords:** Side Information, Weblog Clustering.

**Corresponding Author: MR. RAJESH A. ROY**

**Access Online On:**

www.ijpret.com

**How to Cite This Article:**

Rajesh A. Roy, IJPRET, 2016; Volume 4 (9): 1260-1268

*PAPER-QR CODE*

## INTRODUCTION

Side-information is available along with the text documents. Such side-information may be of different kinds, such as document provenance information, the links in the document, user-access behavior from web logs.(7) Other non-textual attributes which are embedded into the text document. Such attributes may contain a tremendous amount of information for clustering purposes. However, the relative importance of this side-information may be difficult to estimate, especially when some of the information is noisy. In such cases, it can be risky to incorporate side-information into the mining process, It can either improve the quality of the representation for the mining process, or can add noise to the process. Therefore, we need a principled way to perform the mining process, so as to maximize the advantages from using this side information. (7) In order to achieve this goal design algorithm which will combine partitioning approach with probabilistic estimation process and with this there should be the aspect of security of side information and I am designing a IDS system which detect attack on the side information and apply some countermeasure To detect the intruder in process information system, which can detected on the basis of the behavior pattern by entropy method the counter prevention has been taken to protect the data from these intruder attacks.

### Problem Formulation

The problem of text clustering arises in the context of many application domains such as the web, social networks Other digital collections. The rapidly increasing amounts of text data in the context of these large online collections has led to an interest in creating scalable Effective mining algorithms. The problem of pure text clustering, in the absence of other kinds of attributes. In many application domains, a tremendous amount of side information is also associated along with the documents.

### Objective

The main objective of this study can be

There is a need to have text mining application side information is available with text data. This information may be different type such as document provenance information, links in the document, the user access behavior from web logs this information is used for clustering purpose

- Sometime this information may be noisy so designing an approach in order to magnify the coherence between text content and side information and. in case both not show coherent behavior for the clustering process the effect of the portion is marginalized.
- In order to provide security to side information which is used for clustering purposes, design Zombie that will detect attack and provide some countermeasure snort

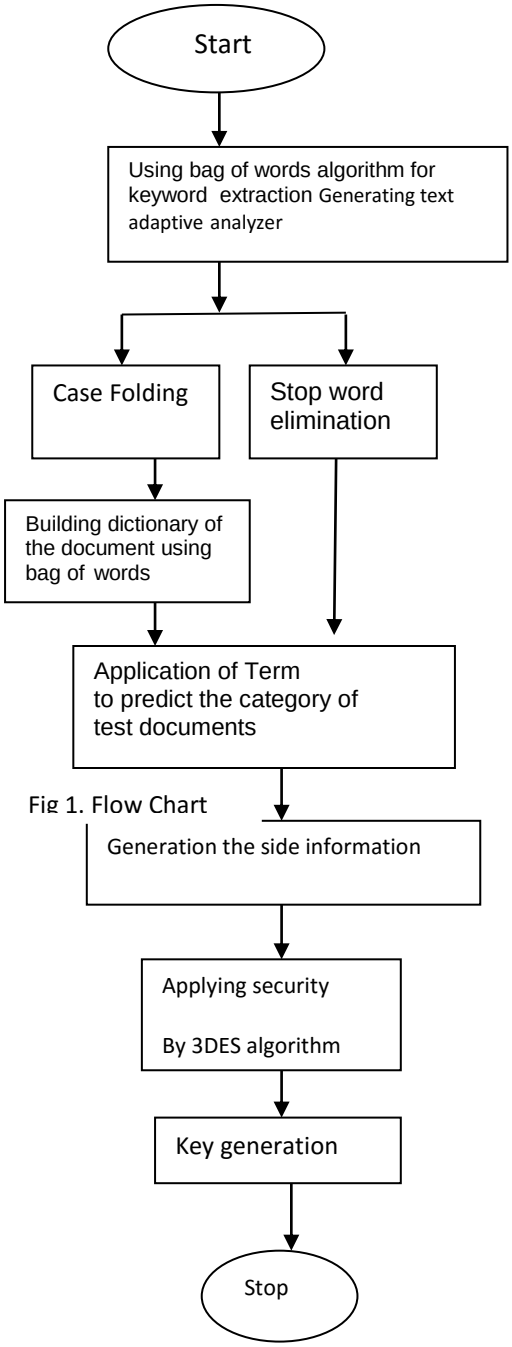


Fig 1. Flow Chart

**Research Methodology/Planning of Work:**

**Methodology**

Text Clustering with Side Information

Documents into k clusters which are denoted by C1 . . . Ck, based on both the text content and the auxiliary variables. I will use the auxiliary information in order to provide additional insights, which can improve the quality of clustering. In many cases, such auxiliary information may be noisy, and may not have useful information for the clustering process. Therefore, we will design our approach in order to magnify the coherence between the text content and the side-information, when this is detected. In cases, in which the text content and side-information do not show coherent behavior for the clustering process, the effects of those portions of the side-information are marginalized.

I presented methods for mining text data with the use of side-information. Many forms of text databases contain a large amount of side-information or meta-information, which may be used in order to improve the clustering process. In order to design the clustering method, we combined an iterative partitioning technique with a probability estimation process which computes the importance of different kinds of side-information. This general approach is used in order to design both clustering and classification algorithms.

**Adaptive Text Filter Algorithm**

We make two key contributions in our research. First, our algorithm adapts standard filtering techniques to the needs of personalized web applications: lightweight, privacy protecting and responsive to attacker provided examples. The algorithm learns a profile of attacker information interests. Second, we adapt a rigorous evaluation method to web systems by using text filtering benchmarks to simulate attacker behavior. Traditionally, Web systems have been evaluated with attacker studies, with the disadvantages of slow data collection, little experimental control and decreased objectivity of conclusions. Relying on simulated attacker feed- back allows us to test many alternative design decisions before subjecting the system to a attacker study, which means we are less likely to waste subjects' time and are more likely to produce a well tuned system

DES is a block cipher encryption algorithm: it takes a fixed-length block of data and converts it into a fixed-length block of encrypted data of the same size by using a symmetric key. The key's length is 64 bits, but because 8 bits are used for parity, the effective key length is 56 bits.

Decryption uses a reverse process on the encrypted data block with the same symmetric key, resulting in the original clear-text block of data.

No easy method has been found to break DES; however, with a brute force approach, guessing the keying information used can be done by trying out $2^{55}$ possible key values. There are other possible methods of breaking DES encryption, but the brute force approach has proven to be the best option. For example, DES was broken in 1998 by a supercomputer in 56 hours and again broken in 1999 in 22 hours by a network of distributed computers. On top of this, it is possible to build specialized hardware appliances to break DES even more quickly . . . in less than 1 hour!

Because computers were becoming more and more powerful during the 1980s and 1990s, and because DES was proven crackable in a reasonable amount of time, NIST created 3DES in 1999. 3DES is basically an enhanced version of DES. 3DES uses three stages of DES and is more secure. DES is applied three times with three different 56-bit keys, resulting in an effective key length of 168 bits. Whereas no successful attack has ever been documented in cracking 3DES, this enhanced security of DES is sufficient for most current applications. No current amount of computing power exists to use a brute force approach to break 3DES.

Social Tagging: Much of the interaction between attackers and social networks occurs in the form of tagging, in which attackers attach short descriptions to different objects in the social network, such as images, text, video or other multimedia data. Social Media provides a wealth of social network data, which can be mined in order to discover useful business applications. Data mining techniques provide researchers and practitioners the tools needed to analyze large, complex, and frequently changing social media data.

We will use the auxiliary information in order to provide additional insights, which can improve the quality of clustering. In many cases, such auxiliary information may be noisy, and may not have useful information for the clustering process. Therefore, we will design our approach in order to magnify the coherence between the text content and the side-information, when this is detected. In cases, in which the text content and side-information do not show coherent behavior for the clustering process, the effects of those portions of the side-information are marginalized we presented methods for mining text data with the use of side-information. Many forms of text databases contain a large amount of side-information or meta-information, which may be used in order to improve the clustering process. In order to design the clustering method, we combined an iterative partitioning technique with a probability estimation process which computes the importance of different kinds of side-information. This general approach is

used in order to design both clustering and classification algorithms. We present results on real data sets illustrating the effectiveness of our approach. The results show that the use of side-information can greatly enhance the quality of text clustering and classification, while maintaining a high level of efficiency

**Features of Net**

Microsoft. NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There's no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).

2. A hierarchical set of class libraries.

The CLR is described as the "execution engine" of .NET. It provides the environment within which programs run. The most important features are Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the Targeting CLR can, depending on the language you're using, impose certain constraints on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn't get garbage collected but instead is looked after by unmanaged code.

- Platform being executed on.
- Memory management, notably including garbage collection.
- Checking and enforcing security restrictions on the running code.
- Loading and executing programs, with version control and other such features.
- The following features of the .NET framework are also worth description:

**Managed Code**

The code that targets .NET, and which contains certain extra

Information - “metadata” - to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

**Managed Data**

With Managed Code comes Managed Data. CLR provides memory allocation and

Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not.

**PROCESS**

- ☐ Text preprocessing
- ■ Semantic text analysis
- ☐ Features Generation
- ■ Bag of words
- ☐ Features Selection
- ■ Simple counting
- ☐ Text/Data Mining
- ■ Classification- Supervised learning
- ☐ Analyzing results.

**CONCLUSION**

We presented methods for mining text data with the use of side-information. Many forms of text databases contain a large amount of side-information or meta-information, which may be used in order to improve the clustering process. In order to design the clustering method, we combined an iterative partitioning technique with a probability estimation process which computes the importance of different kinds of side-information. This general approach is used

in order to design both clustering and classification algorithms. We present results on real data sets illustrating the effectiveness of our approach. The results show that the use of side-information can greatly enhance the quality of text. The results show that the use of side-information can greatly enhance the quality of text clustering and classification, while maintaining a high level of efficiency.

**REFERENCE**

1. C. C. Aggarwal and H. Wang, *Managing and Mining Graph Data.*New York, NY, USA: Springer, 2010.
2. C. C. Aggarwal, *Social Network Data Analytics. New York, NY,*USA: Springer, 2011.
3. C. C. Aggarwal and C.-X. Zhai, *Mining Text Data. New York, NY,*USA: Springer, 2012.
4. C. C. Aggarwal and C.-X. Zhai, "A survey of text classification algorithms," in *Mining Text Data. New York, NY, USA: Springer,* 2012.
5. C. C. Aggarwal and P. S. Yu, "A framework for clustering massive text and categorical data streams," in *Proc. SIAM Conf. Data Mining, 2006, pp. 477–481.*
6. C. C. Aggarwal, S. C. Gates, and P. S. Yu, "On using partial supervision for text categorization," *IEEE Trans. Knowl. Data Eng.,*vol. 16, no. 2, pp. 245–255, Feb. 2004.
7. C. C. Aggarwal and P. S. Yu, "On text clustering with side information," in *Proc. IEEE ICDE Conf., Washington, DC, USA,* 2012.
8. J. Chang and D. Blei, "Relational Topic Models for Document Networks," in AISTASIS , pp. 81–88, 2009.
9. D. Cutting, D. Karger, J. Pedersen, and J. Tukey, "Scatter/Gather: A luster-based Approach to Browsing Large Document Collections," in ACM SIGIR Conf. , pp. 318–329, 1992.
10. I. Dhillon, "Co-clustering Documents and Words using bipartite spectral graph partitioning," in ACM KDD Conf. , pp. 269–274, 2001.
11. I. Dhillon, S. Mallela and D. Modha , "Information-theoretic Co-Clustering," in ACM KDD Conf. , pp. 89–98, 2003.
12. M. Franz, T. Ward, J. S. McCarley, and W. J. Zhu, "Unsupervised and supervised clustering for topic tracking," in ACM SIGIR Conf., pp. 310–317, 2001.
13. G. P. C. Fung, J. X. Yu, and H. Lu, "Classifying text streams in thepresence of concept drifts," in PAKDD Conf. , pp. 373–383, 2004.
14. H. Frigui and O. Nasraoui, "Simultaneous clustering and dynamic keyword weighting for text documents, Survey of text mining," Michael Berry, Ed,Springer , pp. 45–70, 2004.
15. S. Guha, R. Rastogi, and K. Shim, "CURE: An Efficient Clustering Algorithm for Large Databases," in ACM SIGMOD Conf., pp. 73–84, 1998.

16. S. Guha, R. Rastogi, and K. Shim, "ROCK: A Robust Clustering Algorithm for Categorical Attributes," in Inf. Syst. , vol. 25(5), pp. 345–366, 2000.
17. Q. He, K. Chang, E.-P. Lim and J. Zhang, "Bursty feature representation for clustering text streams," in SDM Conf. , pp. 491–496, 2007.
18. A. Jain and R. Dubes, Algorithms for clustering data, Prentice-Hall, Inc., 1988.
19. T. Liu, S. Liu, Z. Chen, and W.-Y. Ma, "An evaluation of feature selection for text clustering," in ICML Conf., pp. 488–495, 2003.
20. A. McCallum, "Bow: A Toolkit for Statistical Language Modeling, Text Retrieval, Classification and Clustering," http://www.cs.cmu.edu/ mccallum/bow