Eleyon
SOUTH ASIAN JOURNAL OF ENGINEERING AND TECHNOLOGY

Full Length Article

# Cosdes: A Collaborative Spam Detection System With A Novel E-Mail Abstraction Scheme

**J. Nirmala Gandhi [a], K. Mohanraj [b] , K.Nandhini [b], K. Poovarasan [b], K.Prakashraj [b]**

*a Associate Professor, Department of Computer Science of Engineering ,K.S.R. College of Engineering, Tiruchengode.*
*b Student, Department of Computer Science of Engineering ,K.S.R. College of Engineering ,Tiruchengode.*
*b Student, Department of Computer Science of Engineering ,K.S.R. College of Engineering, Tiruchengode.*
*b Student, Department of Computer Science of Engineering ,K.S.R. College of Engineering, Tiruchengode.*
*b Student, Department of Computer Science of Engineering ,K.S.R. College of Engineering ,Tiruchengode.*

***Corresponding Author**
J. Nirmala Gandhi

**ABSTRACT:** E-mail communication is indispensable nowadays, but the e-mail spam problem continues growing drastically. In recent years, the notion of collaborative spam filtering with near-duplicate similarity matching scheme has been widely discussed. The primary idea of the similarity matching scheme for spam detection is to maintain a known spam database, formed by user feedback, to block subsequent near-duplicate spams. On purpose of achieving efficient similarity matching and reducing storage utilization, prior works mainly represent each e-mail by a succinct abstraction derived from e-mail content text. However, these abstractions of e-mails cannot fully catch the evolving nature of spams, and are thus not effective enough in near-duplicate detection. This paper proposes a novel e-mail abstraction scheme, which considers e-mail layout structure to represent e-mails. A procedure is presented to generate the e-mail abstraction using HTML content in e-mail, and this newly devised abstraction can more effectively capture the near-duplicate phenomenon of spams. Moreover, a complete spam detection system Cosdes (standing for Collaborative Spam Detection System) is designed, which possesses an efficient near-duplicate matching scheme and a progressive update scheme. The progressive update scheme enables system Cosdes to keep the most up-to-date information for near-duplicate detection.

## 1 Introduction

Data mining is the process of extracting patterns from data. Data mining is seen as an increasingly important tool by modern business to transform data into an informational advantage. It is currently used in a wide range of profiling practices, such as marketing, surveillance, fraud detection, and scientific discovery.

Clustering is an automatic learning technique aimed at grouping a set of objects into subsets or clusters. The goal is to create clusters that are coherent internally, but substantially different from each other. In plain words, objects in the same cluster should be as similar as possible, whereas objects in one cluster should be as dissimilar as possible from objects in the other clusters.Automatic document clustering has played an important role in many fields like information retrieval, data mining, etc. The aim of this thesis is to improve the efficiency and accuracy of document clustering. In this proposed system two clustering algorithms and the fields where these perform better than the known standard clustering algorithms.

Clustering is a division of data into groups of similar objects. Each group, called cluster, consists of objects that are similar between themselves and dissimilar to objects of other groups. In other words, the goal of a good document clustering scheme is to minimize intra-cluster distances between documents, while maximizing inter-cluster distances (using an appropriate distance measure between documents). A distance measure (or, dually, similarity measure) thus lies at the heart of document clustering.

Clustering is the most common form of unsupervised learning and this is the major difference between clustering and classification. No super-vision means that there is no human expert who has assigned documents to classes. In clustering, it is the distribution and makeup of the data that will determine cluster membership. Clustering is sometimes erroneously referred to as automatic classification; however, this is inaccurate, since the clusters found are not known prior to processing whereas in case of classification the classes are pre-defined. Document clustering is being studied from many decades but still it is far from a trivial and solved problem. The challenges are:

- Selecting appropriate features of the documents that should be used for clustering.

- Selecting an appropriate similarity measure between documents.
- Selecting an appropriate clustering method utilising the above similarity measure.
- Implementing the clustering algorithm in an efficient way that makes it feasible in terms of required memory and CPU resources.
- Finding ways of assessing the quality of the performed clustering.

## II.RELATED WORKS

Weize Kong and James Allan Faceted search helps users by offering drill-down options as a complement to the keyword input box, and it has been used successfully for many vertical applications, including ecommerce and digital libraries. However, this idea is not well explored for general web search, even though it holds great potential for assisting multi-faceted queries and exploratory search. In this paper, explore this potential by extending faceted search into the open-domain web setting, which is call Faceted Web Search. To tackle the heterogeneous nature of the web, propose to use query-dependent automatic facet generation, which generates facets for a query instead of the entire corpus. To incorporate user feedback on these query facets into document ranking, we investigate both Boolean filtering and soft ranking models. The authors evaluated Faceted Web Search systems by their utility in assisting users to clarify search intent and subtopic information. The authors described how to build reusable test collections for such tasks, and propose an evaluation method that considers both gain and cost for users. Faceted search enables users to navigate a multi-faceted information space by combining text search with drill-down options in each facet. For example, when searching \computer monitor" in an e-commerce site, users can select brands and monitor types from the the provided facets: fSamsung, Dell, Acer, ...g and f LET-Lit, LCD, OLEDg. This technique has been used successfully for many vertical applications, including e-commerce and digital libraries

Krisztian Balog, Edgar Meij and Maarten de Rijke study consider the task of entity search and examine to which extent state-of-art information retrieval (IR) and semantic web (SW) technologies are capable of answering information needs that focus on entities. We also explore the potential of combining IR with SW technologies to improve the end-to-end performance on a specific entity search task. We arrive at and motivate a proposal to combine text-based entity models with semantic information from the Linked Open Data cloud. The problem of entity search has been and is being looked at by both the Information Retrieval (IR) and Semantic Web (SW) communities and is, in fact, ranked high on the research agendas of the two communities. The entity search task comes in several flavors. One is known as entity ranking (given a query and target category, return a ranked list of relevant entities), another is list completion (given a query and example entities, return similar entities), and a third is related entity finding (given a source entity, a relation and a target type, identify target entities that enjoy the specified relation with the source entity and that satisfy the target type constraint.

Chengkai Li, Ning Yan et al. This paper proposes Facetedpedia, a faceted retrieval system for information discovery and exploration in Wikipedia. Given the set of Wikipedia articles resulting from a keyword query, Facetedpedia generates a faceted interface for navigating the result articles. Compared with other faceted retrieval systems, Facetedpedia is fully automatic and dynamic in both facet generation and hierarchy construction, and the facets are based on the rich semantic information from Wikipedia.The essence of our approach is to build upon the collaborative vocabulary in Wikipedia, more specifically the intensive internal structures and folksonomy. Given the sheer size and complexity of this corpus, the space of possible choices of faceted interfaces is prohibitively large. Authors propose metrics for ranking individual facet hierarchies by user's navigational cost, and metrics for ranking interfaces (each with facets) by both their average pairwise similarities and average navigational costs. Thus, develop faceted interface discovery algorithms that optimize the ranking metrics.

Wisam Dakka, Panagiotis G. Ipeirotis Databases of text and text-annotated data constitute a significant fraction of the information available in electronic form. Searching and browsing are the typical ways that users locate items of interest in such databases. Faceted interfaces represent a new powerful paradigm that proved to be a successful complement to keyword searching. Thus far, the identification of the facets was either a manual procedure, or relied on apriori knowledge of the facets that can potentially appear in the underlying collection. In this paper, we present an unsupervised technique for automatic extraction of facets useful for browsing text databases. In particular, observed through a pilot study, that facet terms rarely appear in text documents, showing that we need external resources to identify useful facet terms. For this, first identify important phrases in each document. Then, expand each phrase with "context" phrases using external resources, such as WordNet and Wikipedia, causing facet terms to appear in the expanded database

Amaç Herdagdelen et al describe the authors of this paper, presents a novel approach to query reformulation which combines syntactic and semantic information by means of generalized Levenshtein distance algorithms where the substitution operation costs are based on probabilistic term rewrite functions.

We investigate unsupervised, compact and efficient models, and provide empirical evidence of their effectiveness. Further it explores a generative model of query reformulation and supervised combination methods providing improved performance at variable computational costs. Among other desirable properties, our similarity measures incorporate information-theoretic interpretations of taxonomic relations such as specification and generalization. Query reformulation is the process of iteratively modifying a query to improve the quality of search engine results, in order to satisfy one's information need. Search engines support users in this task explicitly; e.g., by suggesting related queries or query completions, and implicitly; e.g., by expanding the query to improve quality and recall of organic and sponsored results. Successful refinements are closely related to the original query. This is not surprising as reformulations involve spelling corrections, morphological variants, and tend to reuse parts of the previous query. More precisely, reformulations are close to the previous query both syntactically, as sequences of characters or terms,1 and semantically, often involving transparent taxonomic relations

Query reformulation modifies the original query with the aim of better matching the vocabulary of the relevant documents, and consequently improving ranking effectiveness. Previous models typically generate words and phrases related to the original query, but do not consider how these words and phrases would fit together in new queries. In this paper, a novel framework is proposed that models reformulation as a distribution of queries, where each query is a variation of the original query. This approach considers a query as a basic unit and can capture important dependencies between words and phrases in the query. Previous reformulation models are special cases of the proposed framework by making certain assumptions.

Idan Szpektor, Aristides Gionis, Yoelle Maarek describe the ability to aggregate huge volumes of queries over a large population of users allows search engines to build precise models for a variety of query-assistance features such as query recommendation, correction, etc. Yet, no matter how much data is aggregated, the long-tail distribution implies that a large fraction of queries are rare. As a result, most query assistance services perform poorly or are not even triggered on long-tail queries. We propose a method to extend the reach of query assistance techniques (and in particular query recommendation) to long-tail queries by reasoning about rules between query templates rather than individual query transitions, as currently done in query-flow graph models. As a simple example, if we recognize that 'Montezuma' is a city in the rare query "Montezuma surf" and if the rule '<city> surf → <city> beach' has been observed, we are able to offer "Montezuma beach" as a recommendation, even if the two queries were never observed in a same session. We conducted experiments to validate our hypothesis, first via traditional small-scale editorial assessments but more interestingly via a novel automated large scale evaluation methodology.

In this paper, focused on related query recommendation, one of the tasks for which the long-tail issue is the most visible. We propose to address the long-tail problem by leveraging query templates, which are query constructs that abstract and generalize queries. Our key idea is to identify rules between templates as means for suggesting related queries. The rationale for our approach is based on the fact that many individual queries share the same query intent while focusing on different entities. Hence, their related queries also share similar structures.

Mariana Damova, Ivan Koychev paper presents a survey of recent extractive query-based summarization techniques. We explore approaches for single document and multi-document summarization. Knowledge-based and machine learning methods for choosing the most relevant sentences from documents with respect to a given query are considered. Further, expose tailored summarization techniques for particular domains like medical texts. The most recent developments in the feld are presented with opinion summarization of blog entries.

This survey is motivated by the idea of making e-books more intelligent, in particular enabling them to "answer" users' queries. To find the needed information in books users usually do not want to spend a long time searching, browsing or skimming them. They will be happy to have a "guru" nearby that can provide them with the right answer almost simultaneously. For this purpose to had a close look at the area of automated text summarization. Recently, with the increasing of information available online, those approaches have been developed very extensively. In the realm of automatic summarization different kinds of summarization have been attempted. Along with the study distinguish between the following types of summaries according to specific criteria.

## III. METHODOLOGY

### 1. ADD SPAM MAIL CONTENT

In this module, the email content string in HTML format is keyed in or taken from any HTML file and are saved in 'KnownSPAM' table. The details are viewed and modified using data grid view control.

### 2. ADD GMAIL ACCOUNT

In this module, the gmail pop server. Username and password of the given user is keyed in and are saved in 'Users' table. The details are viewed and modified using data grid view control.

### 3. DOWNLOAD GMAIL FOR GIVEN USER

In this module, the gmail pop server name along with Username and password of the given user is fetched and gmail inbox is accessed. The records are saved in 'Inbox' folder of the project. The details are viewed in listview control and web browser control. These mails (HTML content) is then fetched and given for successive modules.

## 4. PROCEDURE SAG EXECUTION

**Procedure SAG is composed of three major phases,**

**Tag Extraction Phase,**
**Tag Reordering Phase, and**
**<anchor> Appending Phase.**

In Tag Extraction Phase, the name of each HTML tag is extracted, and tag attributes and attribute values are eliminated. In addition, each paragraph of text without any tag embedded is transformed to <mytext=>. In lines 4-5, <anchor> tags are then inserted into AnchorSet, and the first 1,023 valid tags are concatenated to form the tentative e-mail abstraction. Note that we retain only the first 1,023 tags as the tag sequence. The main reason is that the rear part of long e-mails can be ignored without affecting the effectiveness of near-duplicate matching.

## 5. SPTREE CONSTRUCTION

In this module, an e-mail abstraction is segmented into several subsequences, and these subsequences are consecutively put into the corresponding nodes from low levels to high levels. As such, an e-mail abstraction is stored in one path from the root node to a leaf node of SpTree, and hence the matching between a testing e-mail and known spams is processed from root to leaf.

The primary goal of applying the tree data structure for storage is to reduce the number of tags required to be matched when processing from root to leaf. Since only subsequences along the matching path from root to leaf should be compared, the matching efficiency is substantially increased.

## 6. SYSTEM COSDES EXECUTION

In this module, three major work, Abstraction Generation, Database Maintenance, and Spam Detection, are included. With regard to Abstraction Generation Module, each e-mail is converted to an e-mail abstraction by Structure Abstraction Generator with procedure SAG. Three types of action handlers, Deletion Handler, Insertion Handler, and Error Report Handler, are involved in Database Maintenance Module. Note that although the term “database” is used, the collection of reported spams can be essentially stored in main memory to facilitate the process of matching. In addition, Matching Handler in Spam Detection Module takes charge of determining results.

## 7. FIND AND REMOVE RANDOM TAGS, SOPHISTICATED TAG PATTERNS AND COSDES EXECUTION

In this module, the scheme is based on HTML tag sequences, random HTML tags will be removed along with random text paragraphs. In addition, if tag patterns that do conform to syntax rules are inserted, they are also removed. Duplicated subsequence of tags are also found out and removed. Then COSDES execution is carried out.

## IV. CONCLUSION

The new system eliminates the difficulties in the existing system. It is developed in a user-friendly manner. The aim of the project is to achieve efficient similarity matching and reducing storage utilization, prior works mainly represent each e-mail by a succinct abstraction derived from e-mail content text.

- This software is very particular in finding SPAM even if sophisticated HTML tag insertion is made by spammers.
    - A good documentation of user-friendly features had been incorporated in the system.
    - The system has been introduced to eliminate human error.
    - To minimize the time consumption and design & development work.

It is believed that almost all the system objectives that have been planned at the commencement of the software development have been net with and the implementation process of the project is completed. A trial run of the system has been made and is giving good results the procedures for processing is simple and regular order.

The process of preparing plans had been a new experience, which was found use full in later phases of the project is completed. Efforts had been taken to make the system user friendly and as simple as possible. However at some points some features may have been missed out which might be considered for further modification of the application. The new system become useful if the below enhancements are made in future.

- ✓ Duplicate sequences of tag patterns are found efficiently.
- ✓ The SPAM record details can be mailed to administrator for immediate knowledge about those mails

The new system is designed such that those enhancements can be integrated with current modules easily with less integration work.

## References

[1] W. Kong and J. Allan, "Extending faceted search to the general web," in Proc. ACM Int. Conf. Inf. Knowl. Manage., 2014, pp. 839–848.

[2] K. Balog, E. Meij, and M. de Rijke, "Entity search: Building bridges between two worlds," in Proc. 3rd Int. Semantic Search Workshop, 2010, pp. 9:1–9:5.

[3] C. Li, N. Yan, S. B. Roy, L. Lisham, and G. Das, "Facetedpedia: Dynamic generation of query-dependent faceted interfaces for wikipedia," in Proc. 19th Int. Conf. World Wide Web, 2010, pp. 651–660.

[4] W. Dakka and P. G. Ipeirotis, "Automatic extraction of useful facet hierarchies from text databases," in Proc. IEEE 24th Int. Conf. Data Eng., 2008, pp. 466–475.

[5] A. Herdagdelen, M. Ciaramita, D. Mahler, M. Holmqvist, K. Hall, S. Riezler, and E. Alfonseca, "Generalized syntactic and semantic models of query reformulation," in Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. retrieval, 2010, pp. 283–290.

[6] X. Xue and W. B. Croft, "Modeling reformulation using query distributions," ACM Trans. Inf. Syst., vol. 31, no. 2, pp. 6:1–6:34, May 2013.

[7] L. Bing, W. Lam, T.-L. Wong, and S. Jameel, "Web query reformulation via joint modeling of latent topic dependency and term context," ACM Trans. Inf. Syst., vol. 33, no. 2, pp. 6:1–6:38, eb. 2015.

[8] I. Szpektor, A. Gionis, and Y. Maarek, "Improving recommendation for long-tail queries via templates," in Proc. 20th Int. Conf. World Wide Web, 2011, pp. 47–56.

[9] M. Damova and I. Koychev, "Query-based summarization: A survey," in Proc. S3T, 2010, pp. 142–146.

[10] K. Latha, K. R. Veni, and R. Rajaram, "Afgf: An automatic facet generation framework for document retrieval," in Proc.Int. Conf. Adv. Comput. Eng., 2010, pp. 110–114.

[11] [J. Pound, S. Paparizos, and P. Tsaparas, "Facet discovery for structured web search: A query-log mining approach," in Proc. ACM SIGMOD Int. Conf. Manage. Data, 2011, pp. 169–180.

[12] ] W. Kong and J. Allan, "Extracting query facets from search results," in Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval, 2013, pp. 93–102.

[13] Y. Liu, R. Song, M. Zhang, Z. Dou, T. Yamamoto, M. P. Kato, H. Ohshima, and K. Zhou, "Overview of the NTCIR-11 imine task," in Proc. NTCIR-11, 2014, pp. 8–23.