Personalization and Recommendation of web pages using Semi-Supervised Approach

Nirnika L
Nivedita M
Sree Mownika Guntur

SSN College of Engineering, Chennai

April 1, 2014

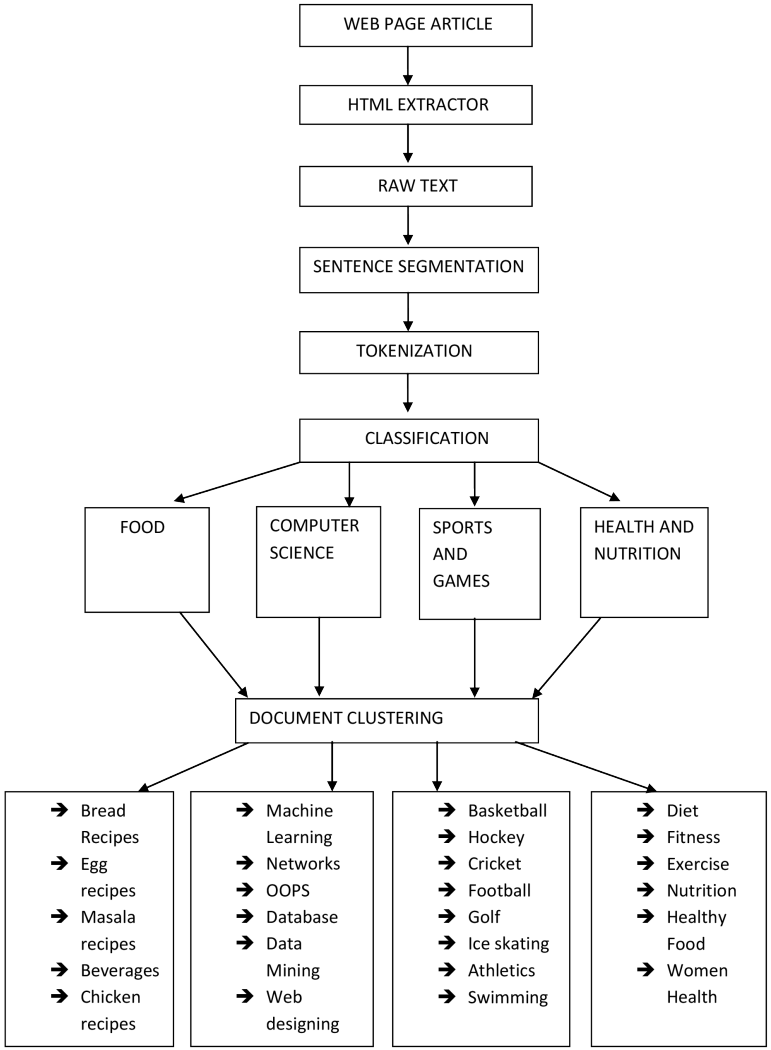
OVERVIEW

- Introduction
- Proposed System
- Module Split-Up
- Methods
- Personalization
- Recommendation
- References

Introduction

- Due to large volume of unstructured data in the web there rises a necessity to categorize and structure the documents
- Document classification is the task of labeling documents with a set of predefined thematic categories.
- Document clustering is an unsupervised approach which groups similar articles together in one cluster

Proposed System



Proposed System

- A Semi-supervised method is proposed to address the problem of personalization and recommendation.
- Document classification and Document clustering methods are combined to achieve this.
- A real time web-application is developed which pools in web articles according to users more specific interest from various websites.

Module Split-Up

Training the classifier

- Features are extracted from the documents.
- The classifier is trained over the various pre-defined categories.
- Classifier uses logistic regression algorithm.

Document Pre-Proessing

- The web page document in the form of a raw HTML format is parsed and content is extracted
- The extracted content is processed by removing stop words and stemming to root form.

Module Split-Up

Document Classification

- The trained classifier returns a set of probability for all the pre-defined categories for the test document
- The categorized document are then further clustered together to find sub-categories

Document Clustering

- Clustering is done with the help of Incremental clustering algorithm .
- This algorithm makes use of similarity and distance metric in order to find the correlation distance between two documents.

Methods

Document Classification

- HTML tags removal
- Feature Extraction
- Logistic Regression

Document Clustering

- Incremental Clustering Algorithm
- Cosine Similarity
- TF-IDF
- Pearson Correlation

Document Classification

- HTML Tags Removal
  - The web page document which is in the form of a raw HTML format is parsed.
  - All the HTML tags are removed from the HTML document using a Python-Goose wrapper.

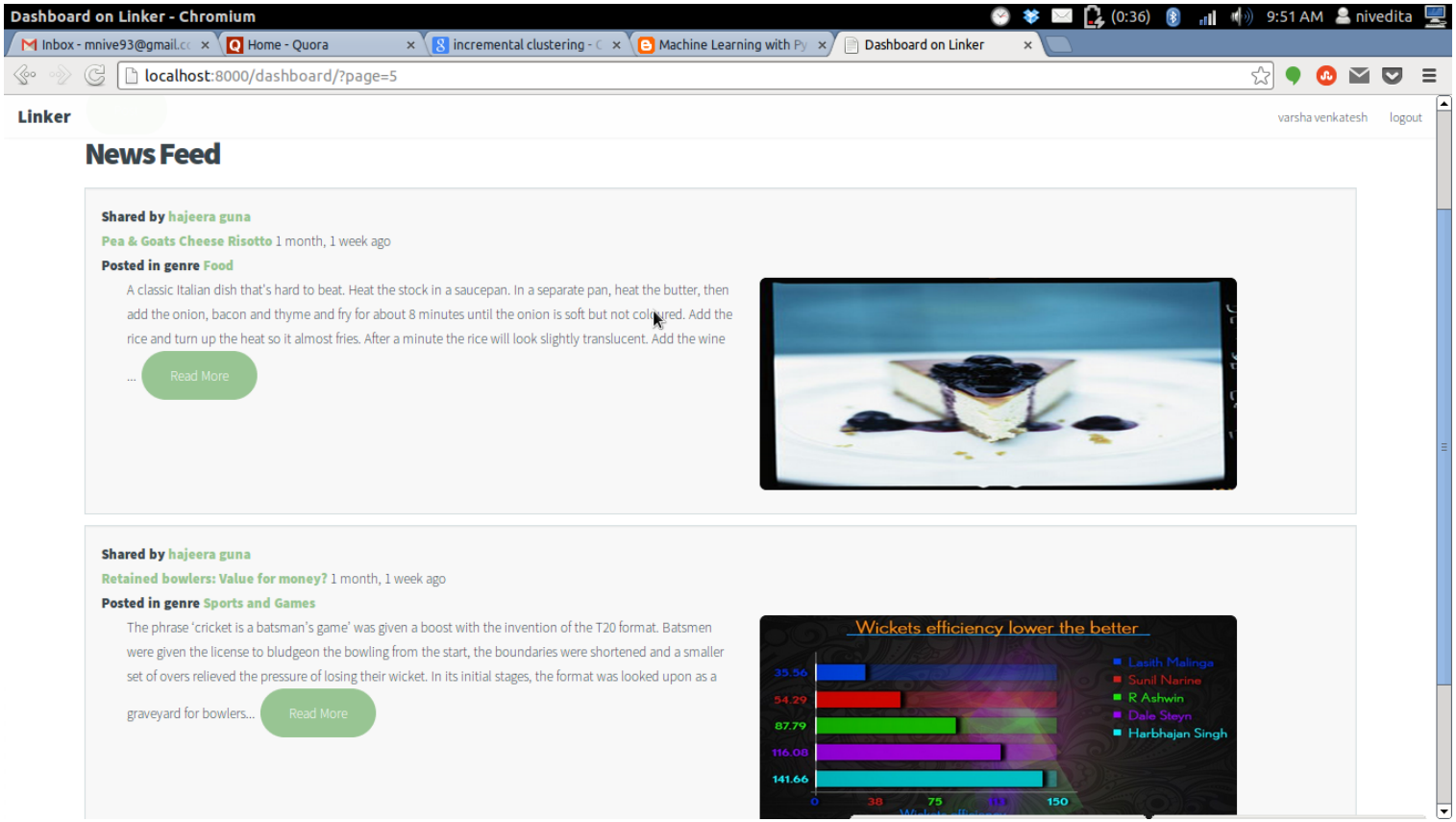
```
def parse_url(url):
 print url
 g = Goose()
 article = g.extract(url=url)
 print article.title
 print article.cleaned_text
 if article.title:
    if(len(article.cleaned_text)>200):
        return article.cleaned_text
```

Document Classification

- Feature Extraction
  - The document is pre-processed by removing stop words from the document and stemming the word to its root form.
  - After pre-processing unique features are selected from the document by using bi-grams.
- Logistic Regression
  - Logistic regression is used for predicting the outcome of a categorical dependent variable.
  - It is based on one or more predictor variables.

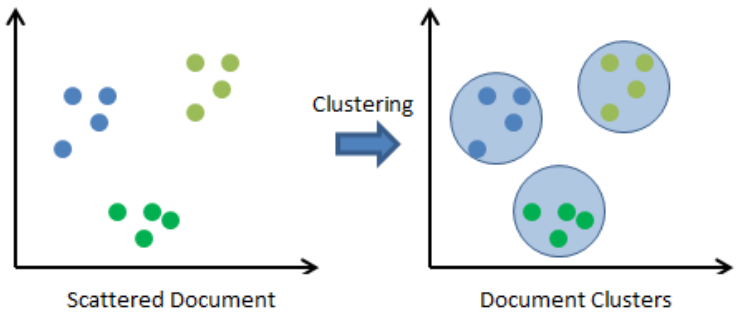
Personalized user feed

- The user feed is customized to suit the user's particular interests thus filtering out web pages that does not appeal to the user.
- This is achieved with the help of classification methods which assigns a category to the web page thus helping to generate a personalized feed.



Document Clustering

- Clustering can be considered the most important unsupervised learning problem deals with finding a structure in a collection of unlabeled data.



- Two or more objects belong to the same cluster if they are close according to a given distance (in this case geometrical distance).

Document Clustering

- Incremental Clustering Algorithm
  - Incremental Clustering algorithm is implemented due to the dynamic nature of the web documents.
  - In Incremental clustering the documents are clustered using a pair wise similarity.
  - If the pair wise similarity falls below a particular threshold,the documents are clustered together
  - Threshold is calculated by finding out the average between highest and lowest pair wise similarity.

Performance Analysis between Distance Metrics

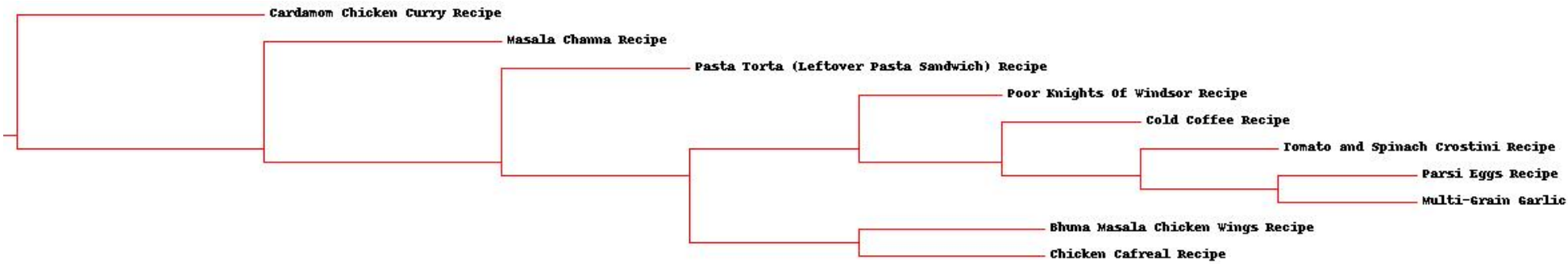


Figure: Euclidean distance with cosine similarity

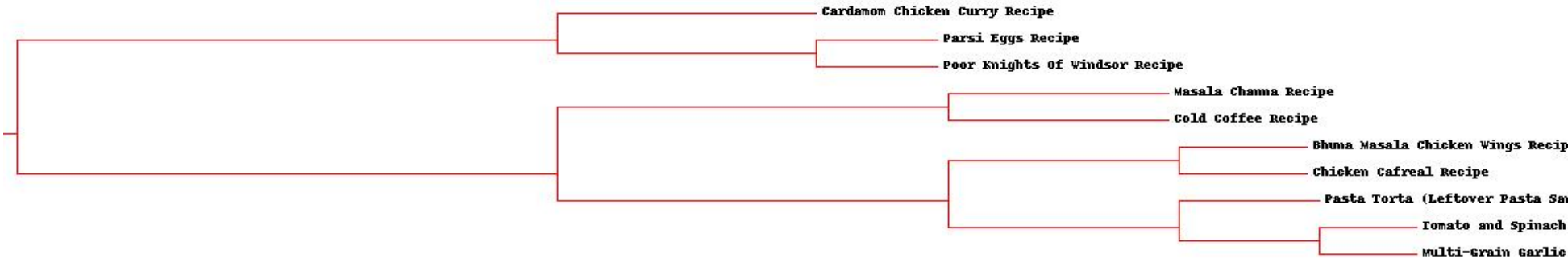


Figure: Hamming distance with cosine similarity

Performance Analysis between Distance Metrics

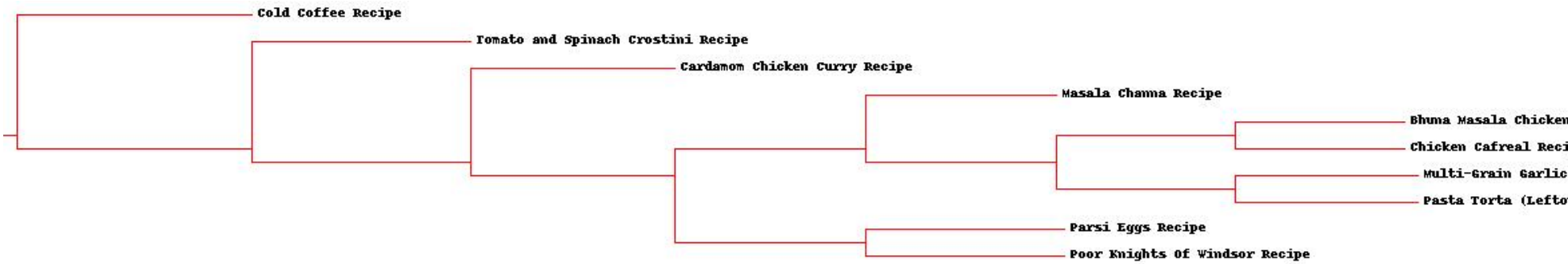


Figure: Tanamoto distance with cosine similarity

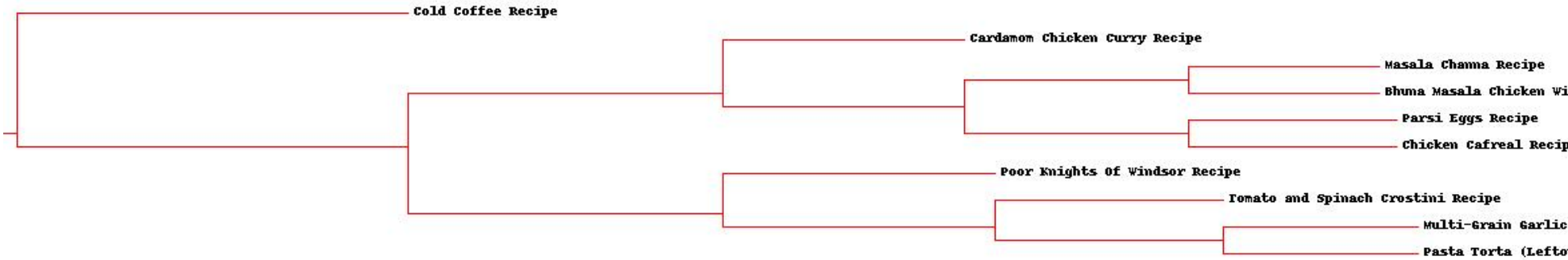


Figure: Pearson distance with cosine similarity

Document Clustering

- Cosine Similarity
  - This metric is frequently used when trying to determine similarity between two documents.
  - In this similarity metric, the attributes is used as a vector to find the normalized dot product of the two documents.

```
def cosim(v1, v2):
    dot_product = sum(n1 * n2 for n1,n2 in zip(v1, v2) )
    magnitude1 = sqrt (sum(n ** 2 for n in v1))
    magnitude2 = sqrt (sum(n ** 2 for n in v2))
    return dot_product / (magnitude1 * magnitude2)
```

Document Clustering

- TF-IDF
  - Term FrequencyInverse Document Frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.

```
from nltk.corpus import stopwords
from math import log
from math import sqrt
def tfidf(alldocument):
   wc = {}
   idf = {}
   tot_tfidf = []
   for content in alldocument:
    temp = []
    normalizedoc = content.lower().split()
    for word in normalizedoc:
      if word not in stopwords.words('english') and len(word) >1:
          wc[word] = normalizedoc.count(word.lower()) / float(len(normalizedoc))
          count = 0
          for c in alldocuments:
             if word.lower() in c:
                count+=1
          if count > 0:
            idf[word] = 1 + log(float(len(alldocument))/count)
          else:
            idf[word] = 1
          temp.append(wc[word]*idf[word])
    tot_tfidf.append(temp)

   return tot_tfidf
```

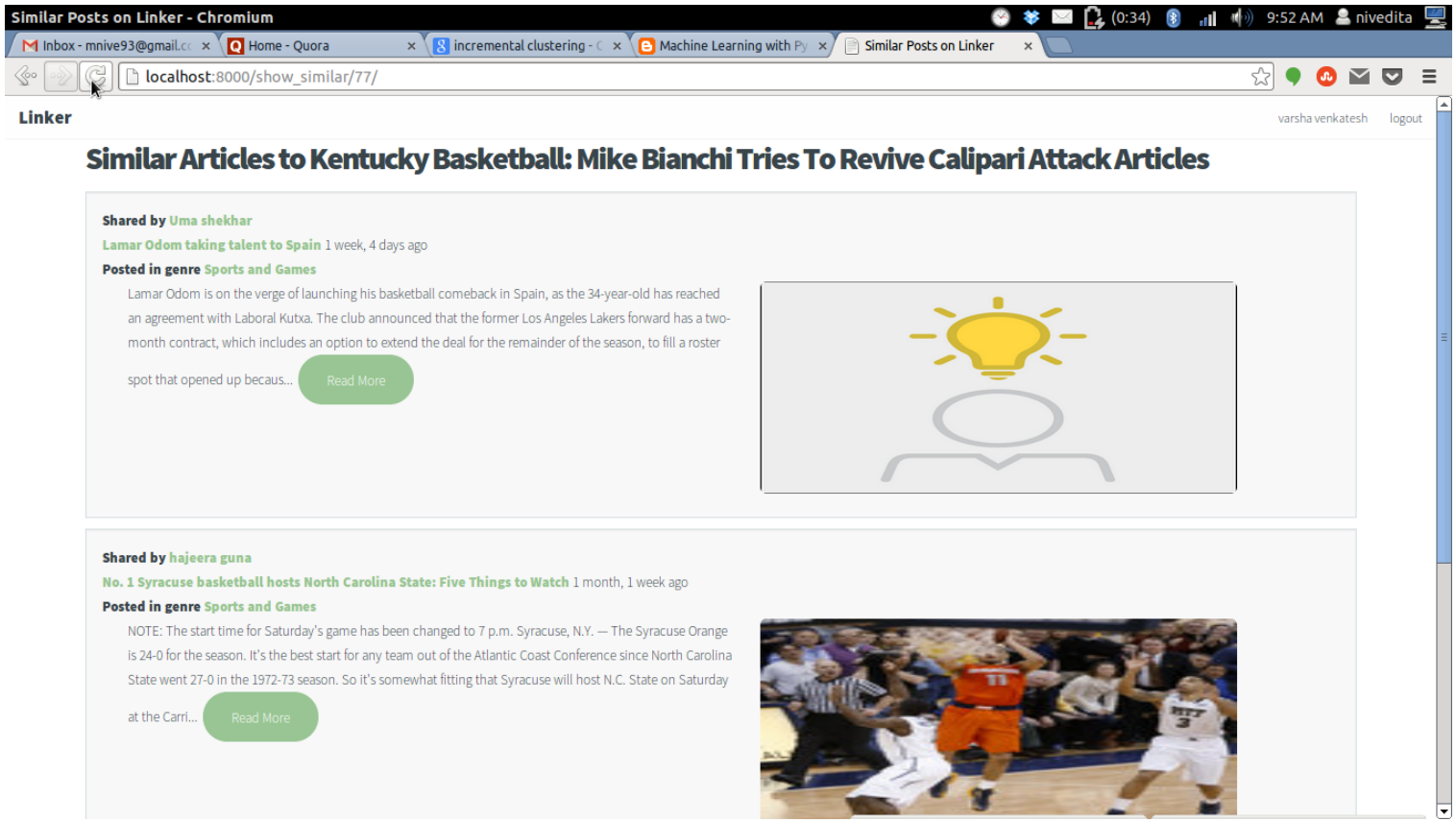
Document Clustering

- Pearson-Correlation
  - This metric measures how highly correlated are two variables and is measured from -1 to +1.

```
def pearson(v1,v2):
  # Simple sums
  sum1=sum(v1)
  sum2=sum(v2)

  # Sums of the squares
  sum1Sq=sum([pow(v,2) for v in v1])
  sum2Sq=sum([pow(v,2) for v in v2])

  # Sum of the products
  pSum=sum([v1[i]*v2[i] for i in range(len(v1))])

  # Calculate r (Pearson score)
  num=pSum-(sum1*sum2/len(v1))
  den=sqrt((sum1Sq-pow(sum1,2)/len(v1))*(sum2Sq-pow(sum2,2)/len(v1)))
  if den==0: return 0

  return 1.0-num/den
```

Recommendation of Similar Articles

- The user's specific interests are targeted by showing similar articles to a particular article.
- This is done with the help of clustering methods.



Conclusion

- We have finally developed a personalized dashboard for a user which streams web documents depending on the users interests .
- The classified documents are then further clustered together using an unsupervised algorithm which is used in identifying the users specific interests.
- The web application supports many concurrent users and the users can share their favourite posts simultaneously which can be viewed by other users depending upon the genre of the article.


Mita K.Dalal and Mukesh A.Zaveri, *Automatic Text Classification:A Technical Review* ,International Journal of Computer Applications 0975-8887,Volume-28-No.2


King-Ip Lin, Ravikumar Kondadadi, *A Similarity-based soft-clustering algorithm for documents*,Department of Mathematical Sciences,The University of Memphis,Memphis, TN 38152, USA.


Shui-Lung Chuang and Lee-Feng Chien,*A Practical Web-based Approach to Generating Topic Hierarchy for Text Segments*,Institute of Information Science ,Academia Sinica ,Taiwan, R.O.C.


Michael Steinbach , George Karypis and Vipin Kumar,*A Comparison of Document Clustering Techniques*,Department of Computer Science and Egineering,University of Minnesota ,Technical Report 00-034


Kentaro Suzuki and Hyunwoo Park ,*NLP-based Course Clustering and Recommendation*,UC Berkley.