Blog to Microblog

Report - Project CSCE - 670

Vandana Bachani

April 29th, 2012

Blog2Microblog is a tool that can summarize articles, based on the social context of the article. It is developed as a python web service interfaced by a simple Web UI. Given a link for a web article, it generates a summary (5 important sentences from the article) and a tweet text. The tool also provides the statistics and reasoning as to why it feels the sentences chosen are important (the support). I look at the problem from the perspective of a recommender system which can recommend the 5 most important (popular) sentences in the article/blog.

1. Summarization Background

Text Summarization is the process of identifying the most salient information in a document or set of related documents and conveying it in less space than the original text. In principle, summarization is possible because of the naturally occurring redundancy in text and because important (salient) information is spread unevenly in textual documents.

Taxonomically one can distinguish among the following types of summaries: extractive/non-extractive, generic/query-based, single-document/multi-document, and monolingual/multilingual/cross-lingual. Most existing summarizers work in an extractive fashion, selecting portions of the input documents (e.g., sentences) that are believed to be more salient. Non-extractive summarization includes dynamic reformulation of the extracted content, involving a deeper understanding of the input text, and is therefore limited to small domains. Query-based summaries are produced in reference to a user query (e.g., summarize a document about an international summit focusing only on the issues related to the environment) while generic summaries attempt to identify salient information in text without the context of a query. Some types of problems that occur in MDS are qualitatively different from the ones observed in SDS: e.g., addressing redundancy across information sources and dealing with contradictory and complementary information. No true multilingual summarization systems exist yet.

Text Summarization is an active field of research in both the IR and NLP communities. Summarization is important for IR since it is a means to provide access to large repositories of data in an efficient way. It shares some basic techniques with indexing, since both indexing and summarization are concerned with identifying the essence of a document. In particular, for domains in which the aspects of interest can be pre-specified, summarization looks very much like Information extraction.

The future vision of the summarization research community is about producing purposeful summaries for a number of textual and semi-structured sources in a right way for a specific user, given the task and the user profile. [10]

2. Motivation and Uses

Why is text summarization important in current scenario?

The following reasons explain the need for text summarization and the motivation for the approach of incorporating user feedback in text summarization:

- The ever-increasing and overwhelming amount of digital content we are required to consume each day necessitates one to develop means to effectively cruise through the information.
- With the advent of smart phones, ipad, netbook and other similar devices, most people are spending more of their online time on smaller devices. As per a recent study 93% of users are going online on their smart devices in US alone. Smaller devices entail smaller screen sizes and bad user experience if too much content is pushed to the screen at a time. Hence there is a need for ways to shorten content, to provide for better user experience along with rich content on these devices.
- It is an active area of research with lots of tools being developed for the purpose but there is still lot of scope for improvement and avenues to be explored.
- The future of text summarization looks bright as we may have many more applications requiring this functionality in future like audio/video summarization, which can benefit from the techniques we invest in today.
- The existing summarization tools do not consider the user feedback on the articles for summarization. These days ample user feedback is available in form of comments, tweets or discussions about a particular article or an event or phenomenon, which can aid the summarization process producing a summary tailored to a user's need in a particular scenario.

Uses:

1. Blog2Microblog can be exploited as a new feature for twitter or any blog/news site wherein user can create content and then generate a summary for the audience. It can improve user experience for twitter users (by providing a small expandable summary section for a given url) and encourage user engagement and content creation.
2. Article summaries (if accurate) may be used by search engines as an extra link under the result to give the reader a gist of the content. These may also be used to assist in retrieval and text classification tasks.
3. Blog digests can be improved to include summaries for the followers.
4. The principles used for summarization in this project can be used for other summarization tasks such as summarization of social updates, etc. and many apps like Siri can benefit from the same.

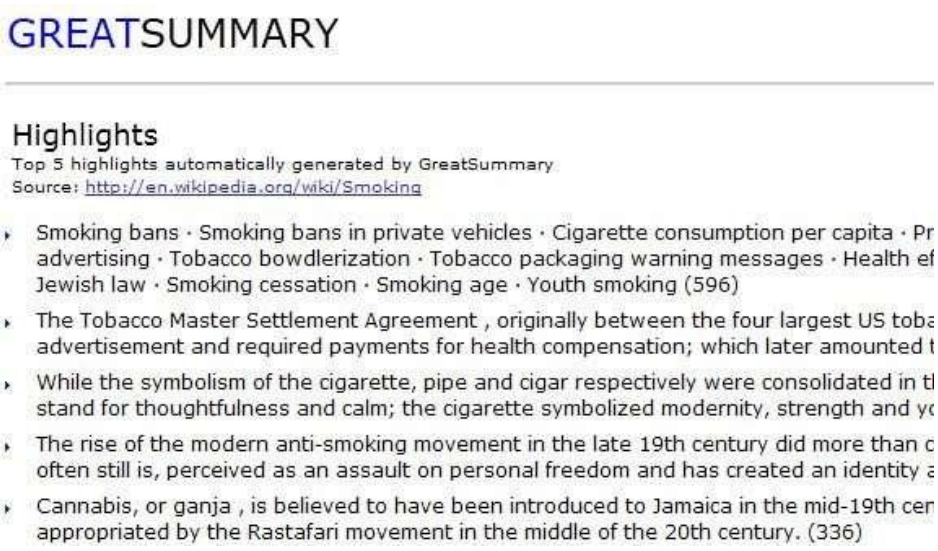
3. Related Work

Good amount of research has been done in the field of text summarization before and a variety of tools exist for the purpose. Some of the existing summarization systems are:

1. **GreatSummary:**

   http://greatsummary.com, which is based on a patent pending research conducted by Yihong Gong, Researcher at NEC Laboratories America, provides a simple web interface where in one can paste the text to be summarized or the url of a web page with the ability for a user to select the size of the summary in terms of number of sentences.

   The algorithm behind this particular approach is simple and effective. The text (as entered by the user or crawled from the url), is divided into sentences. Using a mathematical technique called "Singular Value Decomposition", the system identifies the words that capture the key threads of the text. The process is repeated until the number of sentences requested by the user is reached. GreatSummary ranks the sentences according to the words and returns the results thus obtained to the user.



Highlights

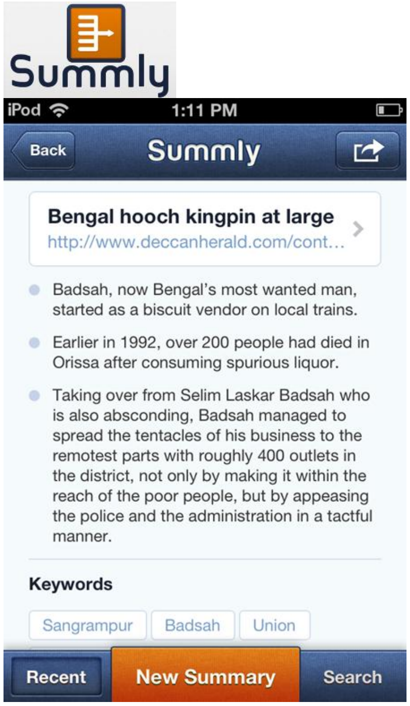
Top 5 highlights automatically generated by GreatSummary
Source: http://en.wikipedia.org/wiki/Smoking

- Smoking bans · Smoking bans in private vehicles · Cigarette consumption per capita · Pr advertising · Tobacco bowdlerization · Tobacco packaging warning messages · Health ef Jewish law · Smoking cessation · Smoking age · Youth smoking (596)
- The Tobacco Master Settlement Agreement , originally between the four largest US toba advertisement and required payments for health compensation; which later amounted t
- While the symbolism of the cigarette, pipe and cigar respectively were consolidated in tl stand for thoughtfulness and calm; the cigarette symbolized modernity, strength and yc
- The rise of the modern anti-smoking movement in the late 19th century did more than c often still is, perceived as an assault on personal freedom and has created an identity a
- Cannabis, or ganja , is believed to have been introduced to Jamaica in the mid-19th cer appropriated by the Rastafari movement in the middle of the 20th century. (336)

2. **Summly:**

“Summly”, an iPhone App developed recently by a 16 year old kid, Brit Nick D’Aloisio, gained lot of attention because of its young inventor. MIT and some venture capitalists have taken keen interest in the research. MIT even conducted some tests on the app and has stated that it is very effective when compared to existing summarization tools (as per the tech blogs). The App takes a URL as an input and produces a summary bounded by the number of words.

The tool utilizes ontological detection and machine learning techniques for summarization (As mentioned in http://www.summly.com/en/technology.html). The tool claims to be language independent. There is a mention on some of the tech blogs that the tool uses some genetic algorithms under the hood to process and train massive collection of web documents. (http://www.crazyengineers.com/summly-app-launched-by-16-year-old-ios-developer-to-summarize-web-1462/)



Bengal hooch kingpin at large
http://www.deccanherald.com/cont...

- Badsah, now Bengal’s most wanted man, started as a biscuit vendor on local trains.
- Earlier in 1992, over 200 people had died in Orissa after consuming spurious liquor.
- Taking over from Selim Laskar Badsah who is also absconding, Badsah managed to spread the tentacles of his business to the remotest parts with roughly 400 outlets in the district, not only by making it within the reach of the poor people, but by appeasing the police and the administration in a tactful manner.

Keywords

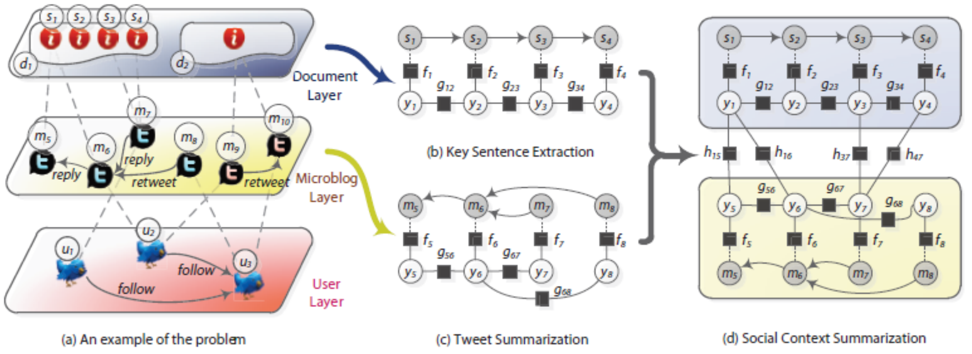
Sangrampur | Badsah | Union

3. **Skimthat:**

http://skimthat.com/, is a recent effort to crowd-based summarization of news articles or events. The members of the “skimthat” community/network are encouraged to write summaries for the articles so that they are available for quick read for a casual reader.

4. **Recent Research Paper:**

With the rapid growth of online social networks, abundant user generated content associated with web documents is also available. A recent approach to summarizing documents comprises of leveraging associated social context. A recent paper by Yang et al. at SIGIR 2011 (http://dl.acm.org/citation.cfm?id=2009916.2009954) explores an approach which models the web documents and social contexts into a unified framework called Dual Wing Factor Graph Model, as is mentioned in the figure below.


Document Layer
Microblog Layer
User Layer
reply
reply
retweet
retweet
follow
follow
d_1
d_2
s_1 s_2 s_3 s_4
m_5 m_6 m_7 m_8 m_9 m_{10}
u_1 u_2 u_3
f_1 f_2 f_3 f_4 f_5 f_6 f_7 f_8
g_{12} g_{23} g_{34} g_{56} g_{67} g_{68}
y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8
h_{15} h_{16} h_{37} h_{47}

(a) An example of the problem (b) Key Sentence Extraction (c) Tweet Summarization (d) Social Context Summarization

Figure 2: An example of the problem and factor graph representations for summarization tasks. In (b), (c), and (d), each gray circle with s_i indicates a sentence in the Web document; its associated white circle with y_i denotes whether the sentence should be included in the document summary. Each gray circle with m_i indicates a tweet and its associated white circle denotes whether the tweet would be included in the tweet summary.

The problem is formulated with an objective function which is a normalized product of three kinds of linear objective functions, i.e. the contribution of features specific to the page's content (eg. length of the sentence, position in document, etc.), the contribution or importance of the tweets about that page's content, and the relation between the author and other users who generated the tweets. The parameters of this objective function are estimated using a maximum likelihood procedure on training instances. This is a non-linear optimization problem and a standard method is used to solve the same. The authors use an inference algorithm to infer the important sentences in the page's content using the parameters and evaluate their results on the test examples using different metrics like precision, recall, Rouge-N, etc. The results were effective with marginal improvement over existing techniques.

How is Blog2Microblog different from the existing techniques?

- The recent research paper mentioned above inspired me to work on this project. Hence, Blog2Microblog is built around the principle of summarizing web blogs based on user generated content for the article but using a simpler approach i.e. formulating the problem as a recommendation system.
- The approach mentioned by the authors is complicated, less scalable and not so easy to implement in comparison to the approach followed by Blog2Microblog.
- Blog2Microblog can also assist in providing user-specific summaries using the power of collaborative filtering (no cold-start problem because of document context).
- The results observed for 'mashable' domain, which is common between the paper approach and Blog2Microblog, show that the latter performs significantly better. Though the dataset and ground truth in the two cases is different, the results differ by a considerable margin (approx. 30%), which suggests that the recommendation approach maybe better. Please refer results section.
- Some blogs from "Skimthat" were used in an experiment for this project as "ground truth" as that summary is written by willing human contributors. Please refer results section.

4. System Design and Architecture

1. Data and Baseline Algorithm

The data for the tool comprises of:

a. Links of blogs or articles from mined tweets were collected using the twitter search api.

b. The whole text of blogs or articles was crawled and fetched.
c. Related Tweets and Comments were collected using the twitter search api.
d. Authors' Social Features, tweet users social features were collected using twitter users/lookup api.

9 popular twitter blogs from different fields such as technology, news, nature, fun, etc. in terms of number of followers and follower activity were selected:
[engadget, techcrunch, nytimes, mashable, businessinsider, fakingnews, espn, treehugger, huffingtonpost].
For initial data analysis and understanding the user reply pattern on blogs, 1336 blogs were successfully parsed (content retrieved) and 23305 related tweets were collected.
Some graphs to assess blog popularity and tweets per blog:

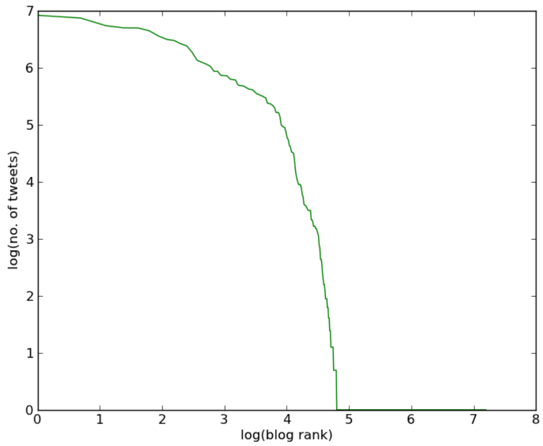


Figure 1

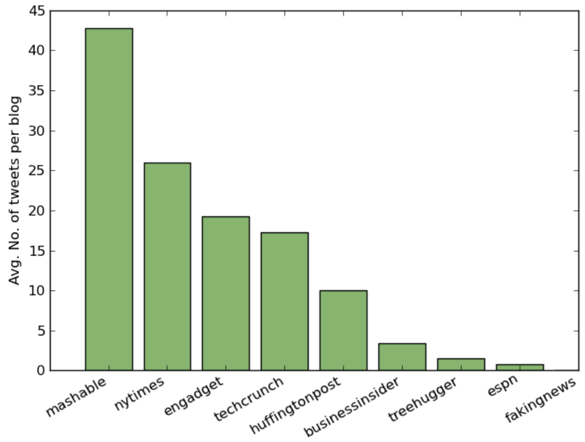


Figure 2

The tweets distribution by blog rank in the collected sample does not follow power law strictly as can be seen in Figure 1 (or maybe the corpus is small). These graphs give a starting point in terms of which blogs to start working with as more users tweet on certain blogs.
To address some of the technical issues faced with the data, additional data was collected, with the addition of 'ndtv' blog as it was difficult to fetch blog content for nytimes. Finally the size of data collected was 3610 blogs and 304599 tweets related to those blogs.

Baseline Algorithm

The baseline algorithm was to summarize a blog based on the structural and data-driven features of the blog itself. The text of a blog was divided into sentences, and a rating or score for each sentence was calculated based on the features of the content of the individual sentences. An SVM based linear regression model was learned to predict the ratings for the sentences of any new blog.

Step1: Data Processing
The following features were calculated per sentence:
a. The position of the sentence in the document.
b. The paragraph a sentence belongs to (paragraph number)
c. The position of the sentence in the paragraph.
d. The average tf-idf score of the terms in the sentence (without the stop words).
e. The cosine similarity score of the sentence with the title of the blog.
f. The length of the sentence.

The idf was calculated using the entire blog corpus (approx. 3600 blogs) and approx. 50000 unique terms were found after 'stopword removal'. Division of text into sentences, removal of stopwords, and tokenization into words is done using functions and data models of *nltk library* as they were more accurate. The sentences were thus converted into item objects and saved for later use for learning the model.

Step 2: Learning
The sentences for each blog were annotated manually (by myself and with the help of a friend) with a rating between 1 and 5 to denote the worthiness of a sentence to be part of the summary. The item objects created as part of the data processing steps were converted into training data vectors, each comprising of the above mentioned 6 feature values and a target rating as annotated. The data was shuffled and divided into training (80%) and test (20%) sets. A SVM based linear regression model was learnt using the training set. *Scikit-learn* library was used for SVM Regression with linear kernel for learning the model.
The model was used to predict ratings for the test set and had a **Root Mean Square Error (RMSE)** of **1.2187**. However, the relatively high error did not affect the results much because the relative ratings of sentences still favored the sentences which were more worthy of being summary sentences.
Top 5 rated sentences are recommeneded as the summary for the article.

2. System Design – Improvement over Baseline

Blog2Microblog is implemented as a recommender system in which the concept of users and items is different from the traditional recommendation system. The tool has been developed as a hybrid recommendation system with dynamic utility matrix and machine learned ratings.
The utility matrix for this recommender system is small enough to fit in memory (unless the article to be summarized is huge and/or there are millions of users who commented on it).
The Utility Matrix comprises of:

a. Items: The sentences in the article.
b. Users: The twitter users and blog commenters, who comment on the article and two hypothetical users which represent the other features of the article (refer the diagram below). The two hypothetical users are the author and the content of the blog. These hypothetical users represent the minimalistic social and document context of the article, which is present when the article is generated and no users have commented on the article.
c. User Ratings: The users' comments and document features are symbolic of the ratings for the various items (sentences) of the article. Objective numerical ratings are calculated for each item as obtained from a regression model which is learned using training data as described later.

Users	Item1	Item2	Item3	Item4	Item5	Item6	Item7	Item8	Item9…
Author	Ra1	Ra2	Ra3	..	..	..	..	..	..
Document Content	Rc1	Rc2	Rc3	..	..	..	..	..	..
User1	Ru1			Ru4					Ru9
User2	Ru1		Ru3						
User3									
…									
Aggregate	2	0	2	1	0	0	0	0	1

Once the utility matrix is constructed, the aggregate ratings are calculated for each sentence using the expression: $\lambda*\mathbf{r_d} + (1-\lambda)*\mathbf{r_u}$, where λ drives the contribution of user tweet ratings in the overall rating of the items and hence the summary and $\mathbf{r_d}$, $\mathbf{r_u}$ are the document and average user/tweet ratings respectively. Top 5 rated sentences are recommeneded as the summary for the article.

3. Implementation

We aid the baseline summarization algorithm with tweets in reply to the blog post by twitter users*. Tweets were converted to tweet objects with each tweet representing a user row in the utility matrix. Item rating per item was calculated for each tweet signifying the items it covers in its content representing a measure of user interest in the particular sentences of the blog.

Step1: Data Processing

The following features were calculated per tweet:

a. Cosine similarity score of the tweet text with each sentence in the article.
b. The length of the tweet.
c. The average tf-idf of terms in the tweet text.
d. Number of followers of the tweet author.

The tweets were thus converted into tweet objects and saved for later use for learning the model.

Step2: Learning

The sentences for each blog were annotated manually with a rating between 1 and 5 to denote the worthiness of a sentence to be part of a summary and a user interest rating. The item model had been trained as part of the baseline process. The tweets model was trained using tweet objects created as part of the data processing steps, by converting into training data vectors, each comprising of the above mentioned 4 feature values and a user interest rating as annotated. The data was shuffled and divided into training (80%) and test (20%) sets. A SVM based linear regression model was learnt using the training set. *Scikit-learn* library was used for SVM Regression with linear kernel for learning the model. The model was used to predict ratings for the test set and had a **Root Mean Square Error (RMSE)** of **0.9186**.

The item and tweet models were saved in memory for recommender system use.

Step3: Recommender System

For summarizing a new blog, the item and tweet objects are created and the utility matrix is generated on the fly containing ratings based on the predicted models and aggregate ratings are computed using the formula $\lambda*\mathbf{r_d} + (1-\lambda)*\mathbf{r_u}$, as explained in the system design.

Top 5 rated sentences are recommeneded as the summary for the article.

Web App and Request Pipeline:

The tool is implemented as a python service called by a Web App. The python web service comprised of a request pipeline which handled an incoming web request and returned a json response. The PHP Web App allows a user to enter a URL for which the summary is sought and a value for λ. The flow of request in the pipeline is as follows:

a. The URL received is crawled and the content is parsed into sentences, using *BeautifulSoup* library.
b. The sentences are converted into item objects, by calculating the features as described in Baseline Algorithm Step 1.

c. Twitter search API is called to get the reply tweets for the given blog url.

d. Tweet objects are created as described in Step 1.

e. The item and tweet vectors along with the value of λ are passed to the recommender system to return top 5 aggregate rated sentences in the order of their occurrence in the blog text (in order to preserve the flow of dialogue in English language.) along with all the support information.

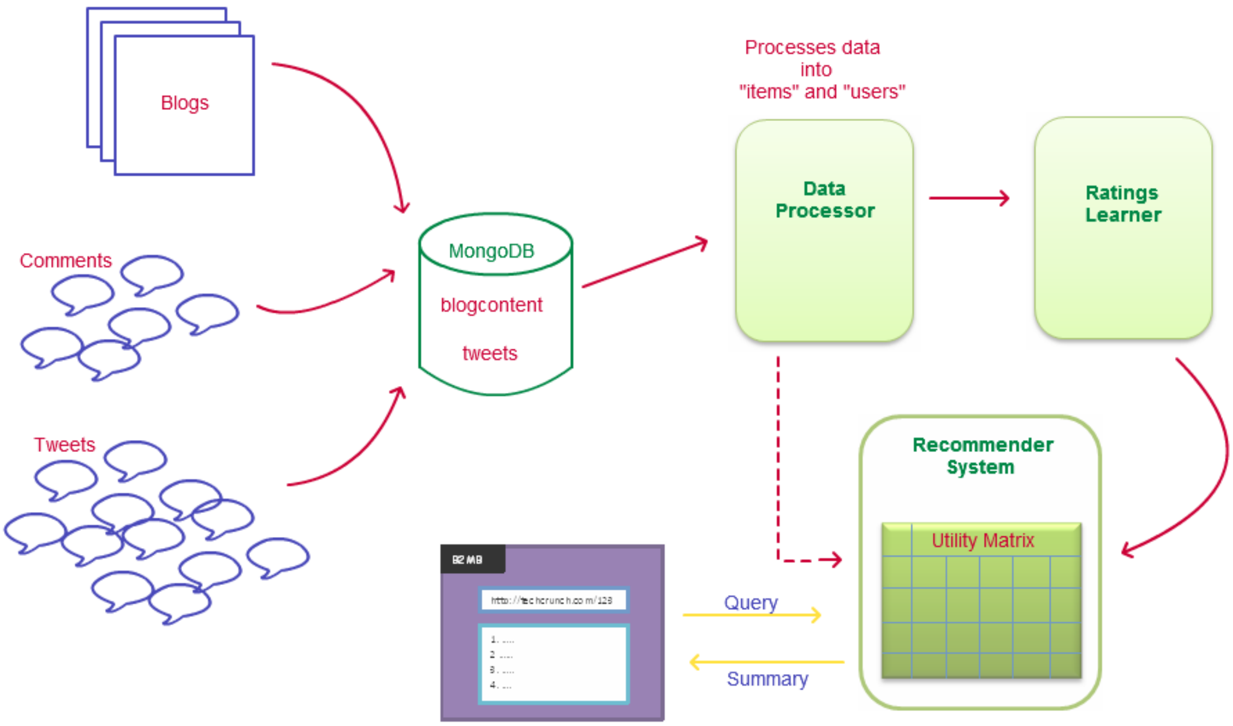
f. The supporting tweets and item ratings are also included as part of the response.

g. Information Gain measure for the tweet text is calculated and added to the response.

h. The response is packaged in a json and sent back to the client which displays it to the user.

*Did not use comments because of technical issues in crawling comments data from blog sites.

4. System Architecture Diagram



5. Experiments and Results

1. Experiment 1:

The ground truth for Experiment 1 was constructed using the manual annotations done as part of the Data Collection and Processing Phase. Approximately 200 blogs were annotated as a whole and 11% of the annotated blog corpus (22 blogs) was separated for ground truth and rest of it along with the related tweets was used for training the item and tweet rating models.

The ground truth was constructed by using the manual rating annotation of sentences in each blog and picking the top 5 sentences as per the manual annotation for the blog.

The system was tested on this ground truth and the following measures were used to evaluate the system.

Precision = $\dfrac{|S_g \cap S_h|}{|S_g|}$ and Recall = $\dfrac{|S_g \cap S_h|}{|S_h|}$

S_g is the set of sentences in the generated summary and S_h is the set of sentences in the ground truth summary.

As the size of both the sets is 5 the precision and recall values are the same.

Result:

Avg. Precision = Avg. Recall = 62.35% for ground truth.

I compared the results with one of the previous work on text summarization namely the "Social Context Summarization" paper at SIGIR'11. [1]

The results observed for 'mashable' domain, which is common between the paper approach and Blog2Microblog, show that the latter performs significantly better. Though the dataset and ground truth in the two cases is different, the results differ by a considerable margin (approx. 30%), which suggests that the recommendation approach maybe better.

Method	**F1 Score for 'Mashable' domain**
Blog2Microblog	65.0%
Social Context Summarization Paper	33.0%

2. **Experiment 2:**

The second evaluation metric that was used was the "Skimthat" summaries which are written by human authors. There were 10 summaries available for http://news.cnet.com/ domain in "Skimthat", which I used as my ground truth to test the tool.

A crawler was written for "cnetnews" domain and the blogs were summarized using the tool.

The summary thus obtained was compared with the human written summary using "cosine similarity score" between the two summaries as human summaries are succinct and subjective unlike the extractive summary produced by Blog2Microblog.

Result:

Average Cosine Similarity Score for the 10 blogs was found to be **0.48** with the **best score** being **0.58**, which shows that around 50-60% of important terms in the summary are captured by the Blog2Microblog summary. (This is after excluding stop words).

But this measure cannot prove that it captured the same essence as the human written summary.

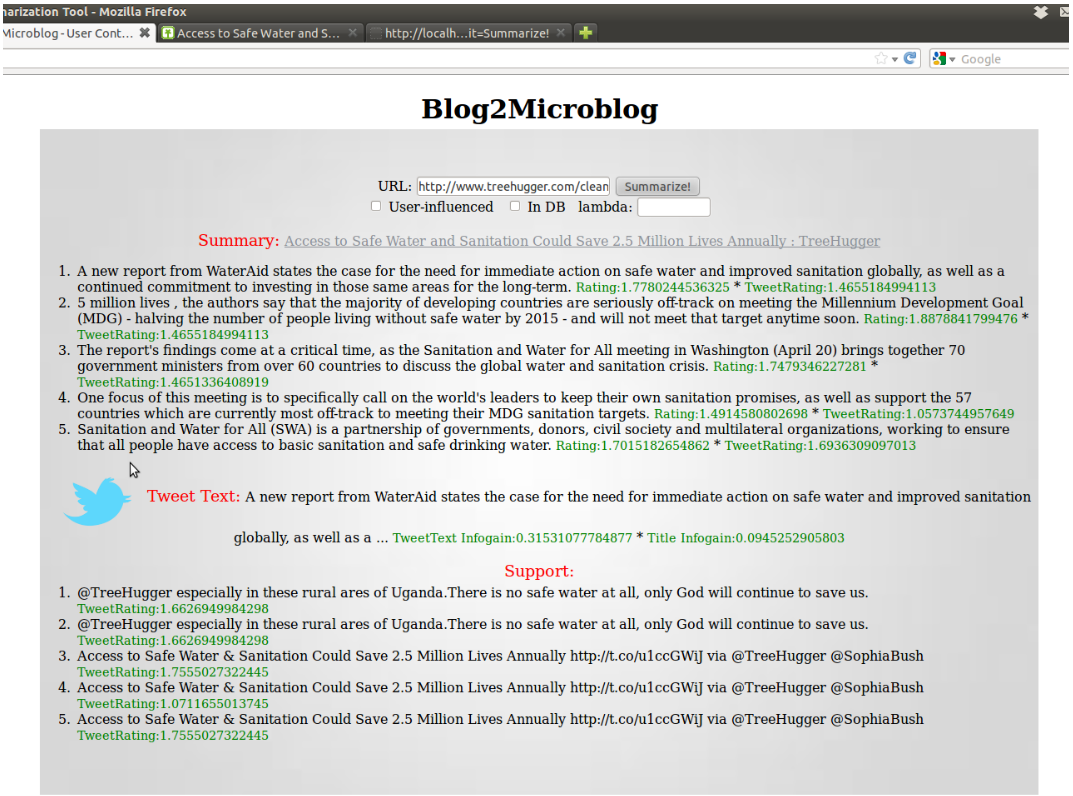
In future for evaluation one can post Blog2Microblog summaries on "Skimthat" and get user feedback in terms of likes and dislikes.

3. **Information Gain Measure for tweet text compared to title – Misleading**

The tweet text is generated as the 140 characters of the top rated summary sentence. The information gain is calculated for the same (after removing the stop words) and is displayed for every summary produced. But sometimes it is misleading as the title is sometimes more informative and still gets a lower information gain score.

4. **Screenshot of the tool**

The below screenshot gives a glimpse of the web app and the summary returned by the tool along with the support information:


arization Tool - Mozilla Firefox
Microblog - User Cont...
Access to Safe Water and S...
http://localh...it=Summarize!
Google
Summarize!

Blog2Microblog

URL: http://www.treehugger.com/clean
☐ User-influenced ☐ In DB lambda:

Summary: Access to Safe Water and Sanitation Could Save 2.5 Million Lives Annually : TreeHugger

1. A new report from WaterAid states the case for the need for immediate action on safe water and improved sanitation globally, as well as a continued commitment to investing in those same areas for the long-term. Rating:1.7780244536325 * TweetRating:1.4655184994113
2. 5 million lives , the authors say that the majority of developing countries are seriously off-track on meeting the Millennium Development Goal (MDG) - halving the number of people living without safe water by 2015 - and will not meet that target anytime soon. Rating:1.8878841799476 * TweetRating:1.4655184994113
3. The report's findings come at a critical time, as the Sanitation and Water for All meeting in Washington (April 20) brings together 70 government ministers from over 60 countries to discuss the global water and sanitation crisis. Rating:1.7479346227281 * TweetRating:1.4651336408919
4. One focus of this meeting is to specifically call on the world's leaders to keep their own sanitation promises, as well as support the 57 countries which are currently most off-track to meeting their MDG sanitation targets. Rating:1.4914580802698 * TweetRating:1.0573744957649
5. Sanitation and Water for All (SWA) is a partnership of governments, donors, civil society and multilateral organizations, working to ensure that all people have access to basic sanitation and safe drinking water. Rating:1.7015182654862 * TweetRating:1.6936309097013

Tweet Text: A new report from WaterAid states the case for the need for immediate action on safe water and improved sanitation globally, as well as a ... TweetText Infogain:0.31531077784877 * Title Infogain:0.0945252905803

Support:

1. @TreeHugger especially in these rural ares of Uganda.There is no safe water at all, only God will continue to save us. TweetRating:1.6626949984298
2. @TreeHugger especially in these rural ares of Uganda.There is no safe water at all, only God will continue to save us. TweetRating:1.6626949984298
3. Access to Safe Water & Sanitation Could Save 2.5 Million Lives Annually http://t.co/u1ccGWiJ via @TreeHugger @SophiaBush TweetRating:1.7555027322445
4. Access to Safe Water & Sanitation Could Save 2.5 Million Lives Annually http://t.co/u1ccGWiJ via @TreeHugger @SophiaBush TweetRating:1.0711655013745
5. Access to Safe Water & Sanitation Could Save 2.5 Million Lives Annually http://t.co/u1ccGWiJ via @TreeHugger @SophiaBush TweetRating:1.7555027322445

6. Learning from the Project

The following concepts have been very useful:

- The basic concepts of tf-idf and cosine similarity are core to the functioning of the project.
- The concept of Recommendation Systems is the defining and most important component of the project. The project itself presents a novel application of Recommender Systems.
- The concepts of learning and prediction from learning to rank and the papers that we read in class helped me devise a system which is mostly dependent on learning methods for predicting ratings and without this knowledge it would have been impossible to go ahead with this project.
- The evaluation metrics like Precision, Recall, F1 score and Cosine Similarity have been used to evaluate the project.

7. Problems/Shortcomings/Challenges

The following are some of the limitations and problem areas of the project:

1. The evaluation metric is still primarily dependent on human judgment which might vary from user to user.
2. Coverage: Whether the produced summary covers the entire content of the blog or not cannot be evaluated.
3. The tool currently only works for the blog domains for which crawling has been implemented, the domains which have been used for training in the project and "csnetnews", because the crawler is not generic and cannot get blog text from any other domain URL.
4. The tool currently caters only to English summaries though it can be extended to multi-lingual domains.

Challenges:

1. Data collection by crawling the web urls for blog content and search api limits did pose challenges as some of the blogs like 'techcrunch' and 'nytimes' allow only registered crawlers or scripts to get the full content

from their pages. Hence this tool is more suited to be embedded where user feedback is accessible at the same place as the content for better performance (within twitter, or within the blog site itself).

2. Some evaluation metric which captures the semantic information in a piece of text is what can revolutionize such a tool (search industry seems to be going that way).
3. The RMSE values for models learnt for tweets and item ratings cannot be compared to other systems as this is a novel application of Recommender Systems.
4. Poor quality user feedback is also a concern (some tweets just repeat the title or are too short like 'lol'; blogs comments are much better in that sense).

8. Acknowledgements

I would like to thank Prof. James Caverlee, Elham Khabiri and Krishna Kamath, for helping me with several concepts through the course of the project. I would also like to thank my friend Swayambhoo Jain for helping in manually annotating some blogs (assigning ratings to sentences) for the data preparation for my project.

9. References

1. Zi Yang, Keke Cai, Jie Tang, Li Zhang, Zhong Su, and Juanzi Li. 2011. Social context summarization. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (SIGIR '11). ACM, New York, NY, USA, 255-264. DOI=10.1145/2009916.2009954
   http://doi.acm.org/10.1145/2009916.2009954
2. How GreatSummary Works
   http://greatsummary.com/how.html
3. Summly.com
   http://www.summly.com/en/technology.html
5. Summly App Launched By 16 Year Old iOS Developer To Summarize Web
   http://www.crazyengineers.com/summly-app-launched-by-16-year-old-ios-developer-to-summarize-web-1462/
6. scikit-learn
   http://scikit-learn.org/stable/
7. Chapter 3: Processing Raw Text
   http://nltk.googlecode.com/svn/trunk/doc/book/ch03.html
8. SkimThat
   http://skimthat.com/
9. Summarization Research Resource
   http://www.summarization.com/
10. http://ciir.cs.umass.edu/irchallenges/presentations/summarization3.doc