

Blog to Microblog

Report Checkpoint 2 - Project CSCE – 670

Vandana Bachani

April 5th, 2012

1. Baseline Algorithm

The baseline algorithm is to summarize a blog based on the structural and data-driven features of the blog itself. The text of a blog is divided into sentences, and a rating or score for each sentence is calculated based on the features of the content of the individual sentences. I try to fit a weighted linear model for the features which can be used to predict the ratings for the sentences of any new blog.

Step1: Data Processing

The data was collected using the twitter search api, for selected/famous twitter bloggers (who post links to their articles on twitter) as mentioned in the previous checkpoint report. Blogs related tweets were also collected. The blog urls collected from the twitter search api are crawled and parsed using the *BeautifulSoup* library. The text of the blog is extracted and divided into paragraphs and sentences and stored into the database and also on files. The sentence files were processed into items with features.

The following features were calculated per sentence:

1. The position of the sentence in the document.
2. The paragraph a sentence belongs to (paragraph number)
3. The position of the sentence in the paragraph.
4. The average tf-idf score of the terms in the sentence (without the stop words).
5. The cosine similarity score of the sentence with the title of the blog.
6. The length of the sentence.

For calculating the idf, the entire blog corpus (that was collected) was considered comprising of approx. 3600 blogs. A dictionary of approximately 40000 unique terms (after removing the stop words) was found from the blog corpus. Each sentence was considered as a document in this application and hence the idf is calculated based on the total number of sentences in the whole blog corpus and the number of sentences a term appears in. The idfs were stored on a text file as key, value pairs for using later.

Division of text into sentences, removal of stopwords, and tokenization into words is done using functions and data models of *nlTK library* as they proved to be much more accurate.

The sentences were thus converted into item objects and saved for later use for learning the model.

****Similarly the features of the tweets were extracted. The tweet features are not discussed here as they are not used for implementation of the baseline algorithm.**

Step 2: Learning

The sentences which were saved in the file were annotated manually (by myself) with a rating between 1 and 5 to denote the worthiness of a sentence to be part of a summary. Due to time constraints only a small portion of the blogs were annotated. Hence the following model was trained on very less data, i.e. approximately 900 items/sentences, and the accuracy of the learner was poor. I intend to improve it in the later phases of the project. However, the low accuracy did not affect the results much because the relative ratings of sentences still favored the sentences which were more worthy of being summary sentences.

The item objects created as part of the data processing steps are 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 but the mean square error of the prediction was huge.

The errors can be attributed to two reasons:

1. Less training data
2. Linear Regression Bias

I plan to annotate more data for learning and will also try svm based classification approach with linear or polynomial kernels to find the best model. An interesting observation was that the relative ratings predicted by the model were close to correct, i.e. the sentences which were good summary candidates were rated higher than the others.

Step 3: The Web App and the End to End call

The tool was implemented as a web App. I implemented a python web service comprising 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. A request to the python webservice is made after the form is submitted with the url entered by the user. The flow of request in the pipeline is as follows:

1. The url received is crawled and the content is parsed into sentences, using *BeautifulSoup* library.
2. The sentences are converted into item objects, by calculating the features as described in Step 1.
3. The item vectors are then passed to the model to predict the ratings.
4. Top 5 rated sentences are chosen and displayed in the order of their occurrence in the blog text in order to preserve the flow of dialogue in English language.
5. 140 characters for the highest rated sentence are converted to the tweet text.
6. The sentences with the ratings are packaged in a json and sent back to the client (the php), which displays it to the user.


2. Intermediate Results and Screenshots

The screenshot shows a web browser window with the title "Blog2Microblog - User Context Based Summarization Tool - Google Chrome". The address bar shows "localhost/summary.php". The page content includes a form with a "URL:" label and a text input field containing "http://www.treehugger.com/c", followed by a "Summarize!" button. Below the form, a "Summary:" section displays a list of five sentences with their corresponding ratings. The first sentence is highlighted in red. Below the summary, a "Tweet Text:" section shows a truncated version of the first sentence, preceded by a blue Twitter bird icon.

URL:

Summary: [Smart Meter Opt Out Requirements Spreading Through California : TreeHugger](#)

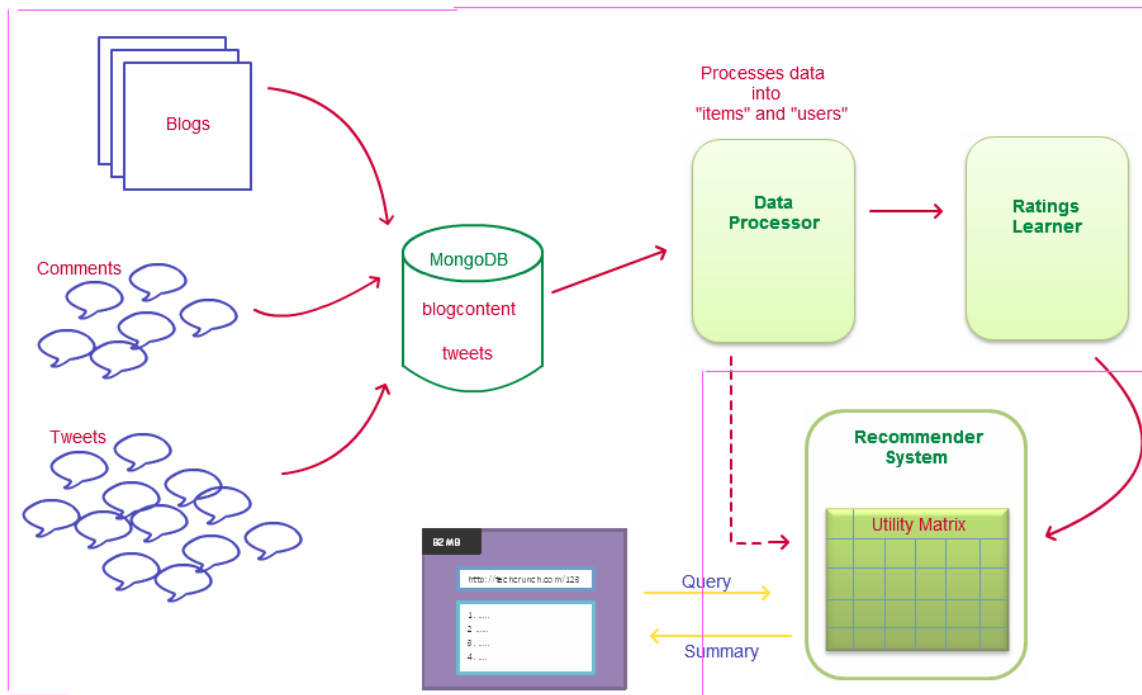
1. Just last month, the California Public Utility Commission ruled that PG&E; must allow customers to opt out of smart meter installations after groups had formed to protest the new wireless meters, but those customers will have to pay a fine and a monthly fee for keeping their analog meters. *Rating:2.7040536542118*
2. Now San Diego Gas & Electric (SDG&E;) and Southern California Edison (SCE) have been handed the same orders from the commission, spreading the opt-out option across the whole state of California. *Rating:2.0085409854288*
3. SDG&E; and SCE can charge up to a \$75 fine and \$10 a month, the same that PG&E; will charge its customers, to cover the cost of sending meter readers out to read the old meters and other costs. *Rating:1.904880975059*
4. The reasons for protesting the smart meters remain the same: health concerns related to the radiation emitted from the devices (which studies have found emit far less than a cell phone) and concerns about the security of the devices that send energy use information to utilities over Wi-Fi. *Rating:2.5347732099562*
5. But while the opposition groups that PG&E; face are mainly community activists, the groups in Southern California seem to have a specific political bent, with many local Tea Party organizations represented, meaning this could become a national cause for the conservative party. *Rating:2.4394317092617*

 **Tweet Text:** Just last month, the California Public Utility Commission ruled that PG&E; must allow customers to opt out of smart meter installations ...

The above screenshot depicts a completed request flow from the tool. The user enters the url in the search box and the server computes a summary and returns the response which is rendered on the page as is shown. The computed ratings are shown for support reasons. The design of the app is not finalized and is subject to changes in later phases.

3. Remaining Work

The remaining work in the project includes improving the current implementation of the baseline algorithm and adding the social context feedback to the summarization process. The following picture gives a sense of progress of the project so far with respect to the proposed design. The “pink” bordered portion in the below image is approximately complete except for the tweet ratings learner.



The following picture shows the portion of utility matrix (marked red) of the recommender system which can be realized with the current implementation. The remaining social elements of the matrix will be added as part of the enhancement to the baseline algorithm.

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

Evaluation:

1. Precision and Recall measures for the predicted ratings.
2. A measure based on intersection between human generated summary and the summary generated by the tool.
3. The measure of information gain of the tweet text produced in contrast with the title of the page.

4. Problems Or Shortcomings

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

1. The evaluation metric is still primarily dependent on human judge and comparison with human annotated labels.
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 blogs which have been considered for training in the project, because of the limitations of the crawler as it is not perfected to crawl and get blog text from any web url.

5. Related Work

Good amount of research has been done in the field of text summarization before. Traditional summarization research has focused on extracting informative sentences from standard documents. Few examples of such systems are <http://greatsummary.com>, which is based on patent pending research conducted by Yihong Gong, Researcher at NEC Laboratories America, and “Summly”, an iPhone App created by 16-year-old, Brit Nick D’Aloisio.

Here is a brief discussion about the algorithms and success of the two products:

GreatSummary:

GreatSummary provides a simple web interface where in one can paste the text to be summarized or the url of a web page. The tool lets one select the number of sentences to be included in the summary. How effective it is? Give example and screenshot.

GREATSUMMARY

Highlights

Top 5 highlights automatically generated by GreatSummary

Source: <http://en.wikipedia.org/wiki/Smoking>

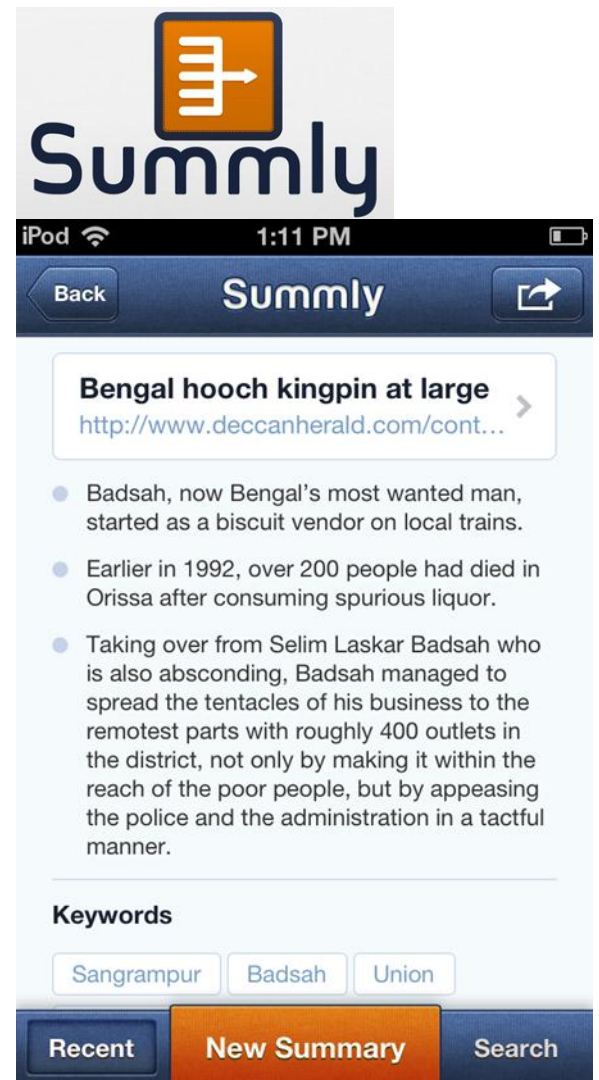
- ✦ Smoking bans · Smoking bans in private vehicles · Cigarette consumption per capita · Pr
advertising · Tobacco bowdlerization · Tobacco packaging warning messages · Health ef
Jewish law · Smoking cessation · Smoking age · Youth smoking (596)
- ✦ The Tobacco Master Settlement Agreement , originally 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 cen
appropriated by the Rastafari movement in the middle of the 20th century. (336)

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.

Summly:

Summly, an iPhone App was 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/>)



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 the 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.

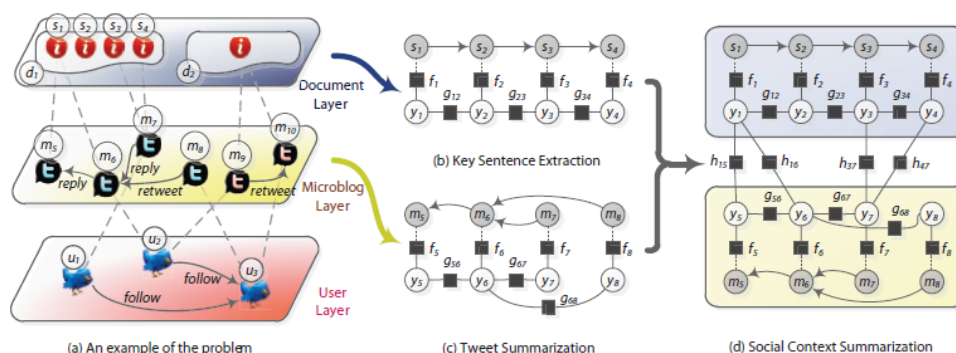


Figure 2: An example of the problem and factor graph representations for summarization tasks. In (b), (c), and (d), each gray circle with s_i indicates a sentence in the Web document; its associated white circle with y_i denotes whether the sentence should be included in the document summary. Each gray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

Further 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. I was inspired by this concept and am trying to develop a system which summarizes web blogs based on user generated content for the blog but using a simpler approach i.e. formulating the problem as a recommendation system.

6. 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>
4. 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/>
5. scikit-learn
<http://scikit-learn.org/stable/>
6. Chapter 3: Processing Raw Text
<http://nltk.googlecode.com/svn/trunk/doc/book/ch03.html>