Text Summarization

Youn, Kim
CS 298
San Jose State University

Outline

- Introduction
- Theory and Concepts
- Design and Implementation
- Summary Evaluation
- Conclusion

Introduction

- Motivation
  - Need for an efficient text summarizer due to the overwhelming amount of textual information available on the Web.
  - Useful for a reader to have access to a concise summary tailored to his or her interests to quickly browse through a large number of blog sites.
  - The utilization of SQL, which can be easily converted into Pig Latin, an SQL-like data transformation language developed at Yahoo and allows for massive parallel processing of large data sets across clusters by compiling queries into Map Reduce jobs and executing them in Hadoop.
- Goal
  - Create a text summarizer to provide condensed versions of original text by identifying the best approximation of original text.

Theory and Concepts

- The Lanczos Algorithm
  - Can determine the eigenvalues for a large sparse matrix efficiently through the employment of the Lanczos recursion, which converts the original matrix A into tridiagonal matrix T through a finite number of orthogonal similarity transformations.
  - The eigenvalues for tridiagonal matrix T are approximate to those of the original matrix, A.
  - The eigenvectors of A can be found through the multiplication of the eigenvectors of T by the Lanczos vectors acquired from the recursion.
  - The number of arithmetical operations required to generate a tridiagonal matrix is proportional to the number of nonzero entries of A, which saves running time for a large sparse matrix.

Theory and Concepts (cont..)

- SVD (singular value decomposition)
  - The SVD theorem is usually presented as:

    $A_{nxp} = U_{nxn} S_{nxp} V^T_{pxp}$

    where $UU^T = U^T U = I$ and $V^T V = VV^T = I$

  - Based on the linear algebra theorem that a rectangular matrix A can be decomposed into the product of three matrices (an orthogonal matrix U, a diagonal matrix S, and the transpose of an orthogonal matrix V), and reconstructed by multiplying the three matrices together.

Theory and Concepts (cont…)

- SVD (cont…)
  - Calculating SVD consists of finding eigenvalues and AA^T eigenvectors of A^TA and AA^T. The eigenvectors of A^TA make up the columns of U and the eigenvectors of make up the columns of V. The singular values in S contain the square roots of the eigenvalues from AA^T or A^TA and are placed along the diagonal of S in descending order.

Theory and Concepts (cont…)

- SVD (cont…)
  - It is a method for reducing a high-dimensional set of data to a lower-dimensional set, which allows us to identify which data exhibit the most variation through an ordering of the dimensions.
  - It gives us the best approximation of the original data by simply ignoring dimensions below certain thresholds, and in doing so this approach reduces the volume of content, while maintaining the main relationships that are present.

Theory and Concpets (cont…)

- Eigenvalues and Eigenvectors
  - If a nonzero vector satisfies the equation below, vector v is called an eigenvector, and scalar λ is called an eigenvalue.

    $A\vec{v} = \lambda\vec{v}$ where A is a square matrix.
  - Eigenvectors and eigenvalues are important in many areas of mathematics and physics.
  - Eigenvectors and eigenvalues can tell us something important about the matrix such as underlying or hidden structure of matrix.

Theory and Concepts (cont…)

- For my project:
  - Worked with a word-by-sentence matrix A, where A_{ij} represents the frequency of a particular word appearing in each sentence.
  - Using SVD, A can be decomposed into three matrices (U, S, and V^T)
  - Each number U_{ij} indicates how strongly related a word is to the topic or concept represented by semantic dimension , while each number V_{ij} indicates how strongly related sentence is to the topic represented by semantic dimension. Each number on the diagonal of S indicates the importance of the corresponding semantic dimension.

Theory and Concepts (cont…)

- To extract sentence for each topic for the summary in my project:

Each entry in the matrix V^T gives information about how closely the sentence is related to the given concept. A higher value means that the sentence is more closely related this concept. Thus, the sentence that is most related to each concept is chosen for the summary until a predefined number of sentences is extracted.

	Sentence 1	Sentence 2	Sentence 3	Sentence 4
Topic 1	0.22	1.51	2.11	7.67
Topic 2	5.33	3.22	1.10	2.43
Topic 3	3.43	5.34	0.74	0.71
Topic 4	2.11	1.31	9.54	2.33

Design And Implementation

- The front end was developed using JQuery, JavaScript and HTML.
  - Used JQuery to make Ajax requests, manipulate the DOM and CSS, and to add effects and animations.
- The back end was developed using PHP and SQL.
  - Used PHP to respond to JQuery Ajax requests, execute SQL queries, dynamically create pages and control the execution of SQL statements when calculating SVD and Lanczos algorithm.
  - Used SQL for calculating SVD and Lanczos algorithm.
- XAMPP was installed for the development environment for the project.

Design and Implementation (cont…)

- My project is implemented in seven major steps.
  1. Crawl blogs for text.
  2. Parse text into sentences and words using stop words and regular expressions.
  3. Calculate the frequencies of the base forms of the words in each sentence and the use of the TF-IDF weighting scheme to scale frequencies.
  4. Compute SVD Using the Lanczos algorithm.
  5. Extract a Sentence for Each Topic for the Summary
  6. Implement back-end functions and interfaces for posting to Twitter and for responding to AJAX requests.
  7. Create user interface functions and the front-end layer.

Design and Implementation (cont…)

- Tables created after calculating SVD and generating the summary.

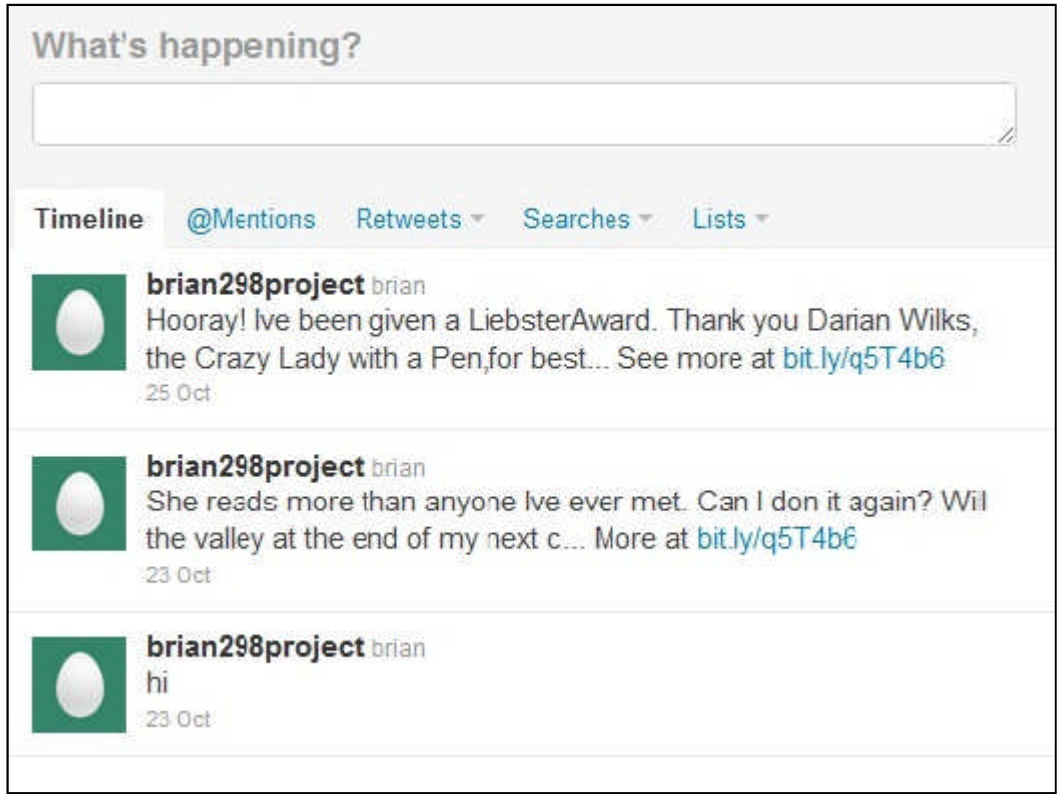
a							247	InnoDB	latin1_swedish_ci	16,0 KiB	-
a2							7,380	InnoDB	latin1_swedish_ci	320,0 KiB	-
alphas							41	InnoDB	latin1_swedish_ci	16,0 KiB	-
at							247	InnoDB	latin1_swedish_ci	16,0 KiB	-
ata							259	InnoDB	latin1_swedish_ci	16,0 KiB	-
betas							41	InnoDB	latin1_swedish_ci	16,0 KiB	-
d							42	InnoDB	latin1_swedish_ci	16,0 KiB	-
e							42	InnoDB	latin1_swedish_ci	16,0 KiB	-
eigenvalues							41	InnoDB	latin1_swedish_ci	16,0 KiB	-
eigenvectors							1,681	InnoDB	latin1_swedish_ci	112,0 KiB	-
r							41	InnoDB	latin1_swedish_ci	16,0 KiB	-
s							1,681	InnoDB	latin1_swedish_ci	112,0 KiB	-
sentence							41	InnoDB	latin1_swedish_ci	16,0 KiB	-
summary							1	InnoDB	latin1_swedish_ci	16,0 KiB	-
temp							0	InnoDB	latin1_swedish_ci	16,0 KiB	-
temp2							0	InnoDB	latin1_swedish_ci	16,0 KiB	-
temp3							1,681	InnoDB	latin1_swedish_ci	112,0 KiB	-
temp4							0	InnoDB	latin1_swedish_ci	16,0 KiB	-
u							0	InnoDB	latin1_swedish_ci	16,0 KiB	-
v							1,681	InnoDB	latin1_swedish_ci	112,0 KiB	-
valideigenvalues							0	InnoDB	latin1_swedish_ci	16,0 KiB	-
word							180	InnoDB	latin1_swedish_ci	16,0 KiB	-

Design and Implementation (cont…)

6. Implement back-end functions and interfaces for posting to Twitter and for responding to AJAX requests.
   - All back-end functions are implemented in PHP, including adding blogs, removing blogs, displaying the summary for a blog, displaying individual posts from selected blogs, shortening long URLs, and posting to Twitter.
7. User Interface Functions and Front-End Layer
   - All Ajax requests were made using JQuery to provide a pleasant experience for the user.

Design and Implementation (cont…)

- End Result:



Summary Evaluation

- Evaluated a summary using the ROUGE evaluation toolkit.
  - ROUGE was developed by Chin-Yew Li and delivered to the research community in 2004.
  - By 2010, more than 150 research sites worldwide had downloaded this software package.
  - ROUGE was chosen as the official automatic evaluation tool by the Document Understanding Conference in 2004, 2005, and 2006.
  - The U.S. government has sponsored summarization evaluation efforts using this tool.

Summary Evaluation (cont…)

- To use ROUGE, the correct directories and formats must be set up beforehand.
  - One is the system-generated summary, also known as the candidate summary, and the other is the gold standard summary, also known as the model summary. The model summary is usually written by humans.
  - Once the proper directories are set up and the files are in the right format, we can run commands such as the one given below.

    ROUGE-1.5.5.pl -e data -u < project-name>/settings.xml.

Summary Evaluation (cont…)

- Test Plans and Datasets
  - Used a publicly available article from Yahoo News about a battle between Samsung Electronics and Apple over Samsung's attempt to get a preliminary injunction against the iPhone 4S in Paris, France, based on an alleged infringement of its 3G patents.
  - Prepared five summaries written by five different individuals. They were asked to pick the one and five most important sentences from the original text.
    - Four summaries by person A, person B, person C and person D are used as model summaries.
    - The candidate summary is then measured against the model summaries.
  - Compared a summary generated by my application with three baselines: a summary by a human (person E), a summary generated by MEAD, a well-known open-source summarizer, and a selection of one and five random sentences from the text.

Summary Evaluation (cont…)

- Results
  - ROUGE-1 scores for the most important sentence

Method	Average Recall	Average Precision	Average FScore
My Application	0.39394	0.34821	0.36967
Random	0.01010	0.03125	0.01527
MEAD	0.27273	0.35526	0.30857
Person E	0.46465	0.35938	0.40529

  - For the most important sentence, person E performed the best, and the random method performed the worst. My application performed better than did MEAD.

Summary Evaluation (cont…)

- Results
  - ROUGE-1 scores for the five most important sentence

System	Average Recall	Average Precision	Average FScore
My Application	0.78571	0.72600	0.75468
Random	0.26623	0.35756	0.30521
MEAD	0.58874	0.57627	0.58244
Person E	0.74892	0.73305	0.74090

  - For the five most important sentences, my application performed the best surprisingly. As expected, the random method performed the worst.

Demo and Conclusion

- Demo:
- Results:
  - My application extracted sentences reasonably well from the original text.
- Future work:
  - Sentence selection should be further refined.