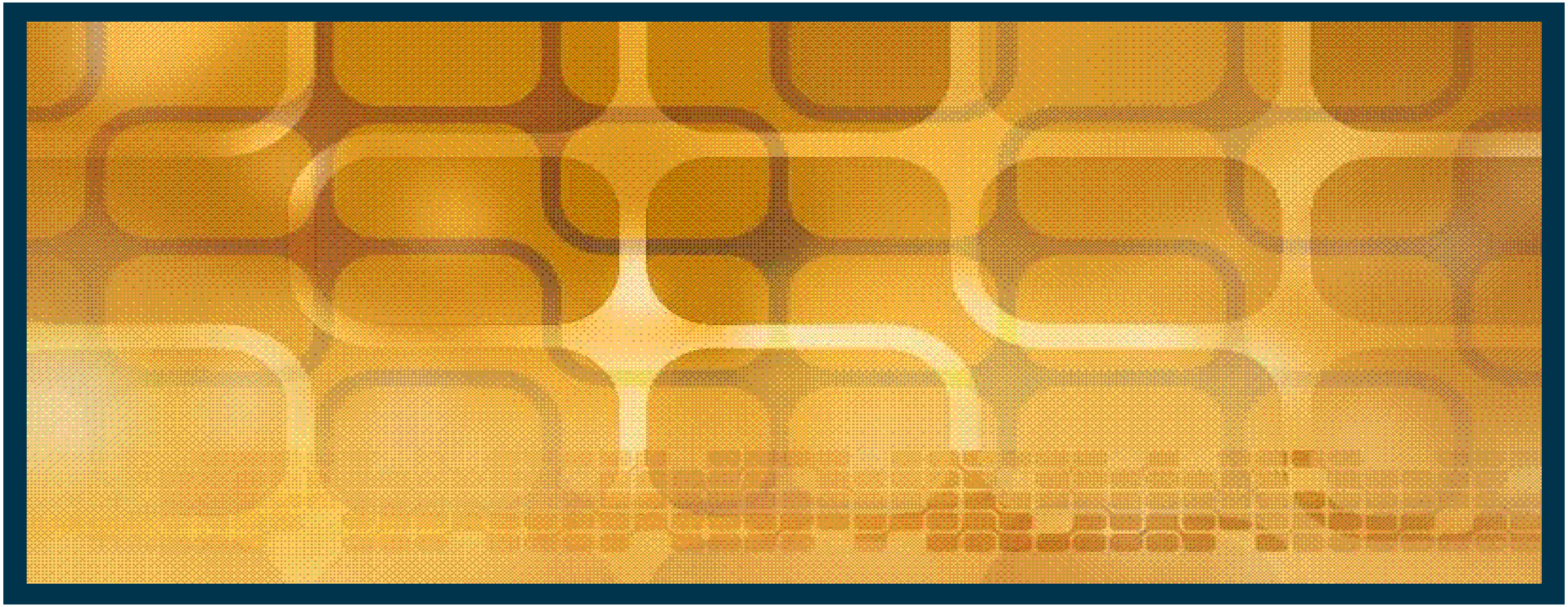


Experiments with and Implementation of a Context Sensitive Text Summarizer

Charles Bocage

Agenda

- The thesis goal
- Automatic Text Summarization (ATS)
- The Yioop search engine
- Recall-Oriented Understudy for Gisting Evaluation (ROUGE)
- My experiments
- Conclusion

Thesis Goal

- To improve the summaries that the Yioop search engine generates with regard to the ROUGE metric by
  - improving the existing summarizers or
  - creating new summarizers or
  - find a solution that can assist the summarizers produce better results

Automatic Text Summarization (ATS)

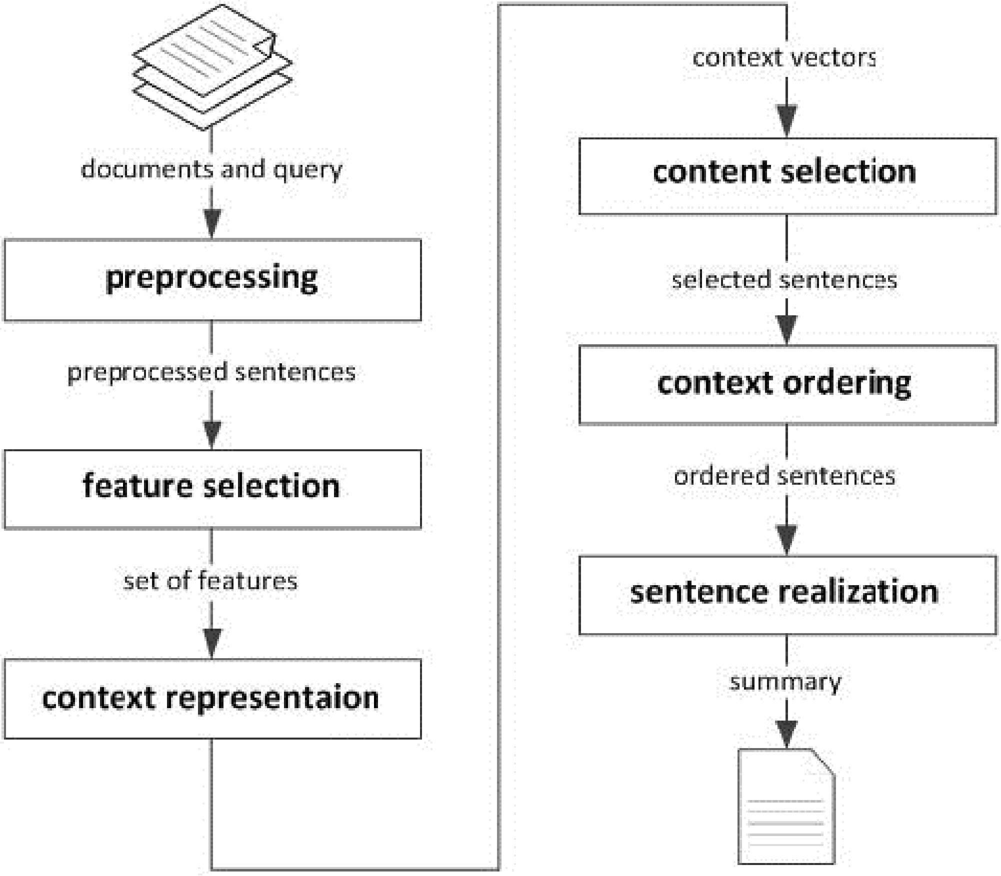
- ATS is trying to solve the problem of getting a computer to summarize content as well or better than humans can

- Summed up as the ability to obtain key ideas from a text passage using as little words a possible.

- Summarization Components
  - Type
    - extract-based *
    - abstract-based
  - Number of documents summarized at one time
    - single-document *
    - multi-document
  - Intelligence
    - knowledge-poor *
    - knowledge-rich
  - Can be extended for a specific use case
    - query-focused
    - identify new pieces of information
    - guided

Automatic Text Summarization (ATS)

- Summarization Phases



The Yioop search engine

- Open source search engine created by Dr. Christopher Pollett at San Jose State University
- Development began in November 2009 but not made public until August 2010
- Its name comes from the Vietnamese word for help (giúp) and its pronunciation yoop
- Since its inception, it has grown to have 30 contributors
- It contains a crawler, a news service, social groups, blogs, wikis, and can host websites.

Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

- Until recently, testing summarizer evaluation was very difficult and expensive to do frequently on a large scale
- It is a PERL script whose purpose is to determine the quality of a summary by comparing it to human summaries developed by Chin-Yew Lin
- ROUGE is the tool in my project used to analyze the results of the summarizers I have created and/or modified
- ROUGE uses various metrics to calculate its statistics: Recall (R), Precision (P), and F measures
- Tests ROUGE performs
  - ROUGE-L
  - ROUGE-S
  - ROUGE-W
  - ROUGE-SU
  - ROUGE-N

$$\mathrm{F} = 2 * \frac{precision * recall}{precision + recall}$$

$$recall = \frac{|\{relevant\ documents\} \cup \{retrieved\ documents\}|}{|\{relevant\ documents\}|}$$

$$precision = \frac{|\{relevant\ documents\} \cap \{retrieved\ documents\}|}{|\{retrieved\ documents\}|}$$

Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

- ROUGE-L: measures sentence-to-sentence similarity based on the longest common subsequence (LCS) statistics between a candidate translation and a set of reference translations [Lin 2004]

- ROUGE-S: computes skip-bigram co-occurrence statistics [Lin 2004]

- ROUGE-W: is an extended version of ROUGE-L. The only difference is ROUGE-W weights the LCS statistics and favors contiguous occurrences

- ROUGE-SU: is an extended version of ROUGE-S. ROUGE-SU considers skip-bigrams and unigrams, hence the addition of U in the name

- ROUGE-N: is an N-gram recall between a candidate summary and the reference summaries. N is the length of the n-gram [Lin 2004] usually between one and four

Experiments Performed

1. Create a Dutch Stemmer for the Yioop Search Engine
2. Compare the Basic Summarizer to the Centroid-Based Summarizer using ROUGE
3. Create a New Summarizer for the Yioop Search Engine
4. Term Frequency Weighting in the Centroid-Based Summarizer
5. Improve the ROUGE Results for Centroid-based Weighting Summarizer
6. A Numerically Stable Lanczos Text Summarization Algorithm
7. Test Yioop Summarizers Against a Large Data Set
8. Improving Text Summarization using Automatic Sentence Compression

Background for the Experiments

- There were two existing summarizers
- Two new summarizers were created
  - the new summarizers were written in PHP and integrated into the Yioop search engine
- Each summarizer was tested thoroughly with ROUGE either using
  - ten websites created during CS200W or
  - Document Understanding Conference (DUC) 2007 documents
- I will demonstrate each summarizer using the below example sentences
  - I went to the store.
  - I bought some chips from the store.
  - I ate the chips on my ride home.

Create a Dutch Stemmer for the Yioop Search Engine Experiment

- Goal of this experiment
  - to gain an understanding of stemmers by adding a stemmer that does not already exist in the Yioop search engine

- A stemmer is an algorithm that reduces all words with the same stem to a common form [Lovins 1968]

- For example, connect, connected, connecting, connection and connections [Porter 1980] would all conflate to connect

Create a Dutch Stemmer for the Yioop Search Engine Experiment

- Installed Yioop and performed a small crawl
- Leveraged the Porter stemmer and the Simplicity Lab PHP Dutch Stemmer
- Wrote the Dutch stemmer in PHP and integrated it into the Yioop Search engine
- Tested the Dutch stemmer against 49000 vocabulary words provided by Martin Porter
- A locale was also created to support the Dutch stemmer

Create a Dutch Stemmer for the Yioop Search Engine Experiment

Compare the BASIC to the CBS using ROUGE Experiment

- Goal of this experiment
  - to compare the Yioop search engine's basic summarizer (BASIC) and centroid-based summarizer (CBS) results
  - Get familiar with the ROUGE software package
  - Analyze the ROUGE results

Compare the BASIC to the CBS using ROUGE Experiment

- Basic Summarizer
  - one of the existing summarizers
  - grabs sentences in a fixed order from beginning to end

- Example sentences
  1. I went to the store.
  2. I bought some chips from the store.
  3. I ate the chips on my ride home.

- The summary would be generated in the order [1, 2, 3]

Compare the BASIC to the CBS using ROUGE Experiment

- Centroid-based Summarizer
  - another one of the existing summarizers
  - hinges its summary generation on using a centroid to get the main idea for the document
    - a centroid is a set of words that are statistically important to the document
    - Computes the Inverse Document Frequency (IDF) for the terms
    - Uses the IDF and term frequencies to calculate the cosine similarity to centroid to rank the sentences
    - Grabs sentences based on the cosine similarity value until it reaches the limit of allowed characters in the summary

Compare the BASIC to the CBS using ROUGE Experiment

- Gets a list of all of the terms

I	ate	bought	chips	from	home	my	on	ride	some	store	the	to	went

- Calculate the number of sentences the term is found in

I:3	ate:1	bought:1	chips:2	from:1	home:1	my:1	on:1	ride:1	some:1	store:2	the:3	to:2	went:1

- Calculate weights using Inverse Document Frequency (IDF)

I:0	ate:1.1	bought:1.1	chips:0.41	from:1.1	home:1.1	my:1.1	on:1.1	ride:1.1	some:1.1	store:0.41	the:0	to:2	went:1.1

- Multiplies the number of sentences the term is found in by the IDF

I:0	ate:1.1	bought:1.1	chips:0.81	from:1.1	home:1.1	my:1.1	on:1.1	ride:1.1	some:1.1	store:0.81	the:0	to:1.21	went:1.1

- Get the top 5 terms for the centroid (word cloud)

to	went	bought	some	from

Compare the BASIC to the CBS using ROUGE Experiment

- Calculate the cosine similarity between the centroid and each sentence
  1. I went to the store = 0.36
  2. I bought some chips from the store = 0.59
  3. I ate the chips on my ride home = 0.74

- The summary would be generated in the order [3, 2, 1] but sorts them in the order they appear in the document

Compare the BASIC to the CBS using ROUGE Experiment

- In general the Basic summarizer produced better results than the Centroid-Based Summarizer

BS ROUGE Result

```
---------------------------------------------
11 ROUGE-1 Average_R: 0.80587 (95%-conf.int. 0.68571 - 0.92333)
11 ROUGE-1 Average_P: 0.70494 (95%-conf.int. 0.55714 - 0.85833)
11 ROUGE-1 Average_F: 0.74742 (95%-conf.int. 0.61115 - 0.88264)
---------------------------------------------
11 ROUGE-2 Average_R: 0.70543 (95%-conf.int. 0.54000 - 0.86333)
11 ROUGE-2 Average_P: 0.61554 (95%-conf.int. 0.43571 - 0.80857)
11 ROUGE-2 Average_F: 0.65227 (95%-conf.int. 0.47596 - 0.83047)
---------------------------------------------
11 ROUGE-3 Average_R: 0.60484 (95%-conf.int. 0.39167 - 0.82833)
11 ROUGE-3 Average_P: 0.52748 (95%-conf.int. 0.31095 - 0.76429)
11 ROUGE-3 Average_F: 0.55747 (95%-conf.int. 0.34470 - 0.78985)
---------------------------------------------
11 ROUGE-4 Average_R: 0.46413 (95%-conf.int. 0.20000 - 0.76667)
11 ROUGE-4 Average_P: 0.42445 (95%-conf.int. 0.16000 - 0.71333)
11 ROUGE-4 Average_F: 0.43933 (95%-conf.int. 0.17500 - 0.73333)
---------------------------------------------
11 ROUGE-L Average_R: 0.56185 (95%-conf.int. 0.42162 - 0.71773)
11 ROUGE-L Average_P: 0.70494 (95%-conf.int. 0.55714 - 0.85833)
11 ROUGE-L Average_F: 0.60646 (95%-conf.int. 0.47795 - 0.75049)
---------------------------------------------
11 ROUGE-W-1.2 Average_R: 0.37926 (95%-conf.int. 0.27541 - 0.49500)
11 ROUGE-W-1.2 Average_P: 0.65106 (95%-conf.int. 0.49107 - 0.82321)
11 ROUGE-W-1.2 Average_F: 0.46368 (95%-conf.int. 0.34814 - 0.59179)
---------------------------------------------
11 ROUGE-S* Average_R: 0.66482 (95%-conf.int. 0.47238 - 0.85524)
11 ROUGE-S* Average_P: 0.53184 (95%-conf.int. 0.32500 - 0.75000)
11 ROUGE-S* Average_F: 0.57648 (95%-conf.int. 0.37678 - 0.78848)
---------------------------------------------
11 ROUGE-SU* Average_R: 0.71369 (95%-conf.int. 0.53984 - 0.88286)
11 ROUGE-SU* Average_P: 0.57277 (95%-conf.int. 0.38142 - 0.78056)
11 ROUGE-SU* Average_F: 0.62202 (95%-conf.int. 0.43843 - 0.81393)
```

CBS ROUGE Result

```
---------------------------------------------
11 ROUGE-1 Average_R: 0.76663 (95%-conf.int. 0.63762 - 0.89333)
11 ROUGE-1 Average_P: 0.67857 (95%-conf.int. 0.53214 - 0.84286)
11 ROUGE-1 Average_F: 0.71577 (95%-conf.int. 0.57970 - 0.86190)
---------------------------------------------
11 ROUGE-2 Average_R: 0.70201 (95%-conf.int. 0.54667 - 0.86333)
11 ROUGE-2 Average_P: 0.61664 (95%-conf.int. 0.43809 - 0.80762)
11 ROUGE-2 Average_F: 0.65163 (95%-conf.int. 0.48667 - 0.83636)
---------------------------------------------
11 ROUGE-3 Average_R: 0.60484 (95%-conf.int. 0.39167 - 0.82833)
11 ROUGE-3 Average_P: 0.53236 (95%-conf.int. 0.31917 - 0.76250)
11 ROUGE-3 Average_F: 0.56079 (95%-conf.int. 0.35098 - 0.79039)
---------------------------------------------
11 ROUGE-4 Average_R: 0.46413 (95%-conf.int. 0.20000 - 0.76667)
11 ROUGE-4 Average_P: 0.42445 (95%-conf.int. 0.16000 - 0.71333)
11 ROUGE-4 Average_F: 0.43933 (95%-conf.int. 0.17500 - 0.73333)
---------------------------------------------
11 ROUGE-L Average_R: 0.54120 (95%-conf.int. 0.39743 - 0.70309)
11 ROUGE-L Average_P: 0.67857 (95%-conf.int. 0.53214 - 0.84286)
11 ROUGE-L Average_F: 0.58289 (95%-conf.int. 0.44879 - 0.72838)
---------------------------------------------
11 ROUGE-W-1.2 Average_R: 0.39129 (95%-conf.int. 0.28736 - 0.50728)
11 ROUGE-W-1.2 Average_P: 0.67325 (95%-conf.int. 0.52159 - 0.83758)
11 ROUGE-W-1.2 Average_F: 0.47857 (95%-conf.int. 0.36753 - 0.60219)
---------------------------------------------
11 ROUGE-S* Average_R: 0.60686 (95%-conf.int. 0.40381 - 0.81667)
11 ROUGE-S* Average_P: 0.49785 (95%-conf.int. 0.28373 - 0.74167)
11 ROUGE-S* Average_F: 0.53466 (95%-conf.int. 0.32789 - 0.76667)
---------------------------------------------
11 ROUGE-SU* Average_R: 0.66706 (95%-conf.int. 0.48421 - 0.84841)
11 ROUGE-SU* Average_P: 0.54722 (95%-conf.int. 0.34603 - 0.76429)
11 ROUGE-SU* Average_F: 0.58952 (95%-conf.int. 0.40240 - 0.79922)
```

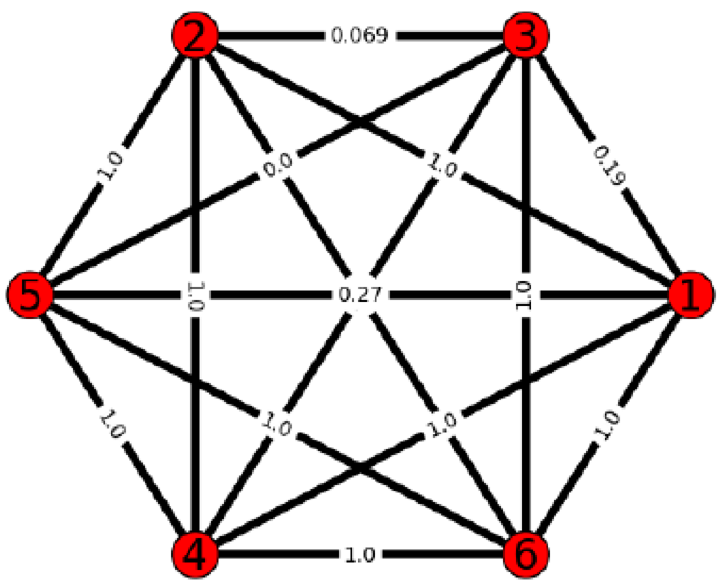
Create a New Summarizer for the Yioop Search Engine Experiment

- Goal of this experiment
  - to find a paper that covers an algorithm used for summarization
  - implement the algorithm in PHP and integrated into the Yioop search engine
  - Analyze the ROUGE results

- The algorithm chosen was the Graph Based summarizer (GBS)

Create a New Summarizer for the Yioop Search Engine Experiment

- It constructs a weighted graph or adjacency matrix out of the text where the sentences are the nodes
- The weights are determined by a distortion measure
  - represents the relation between two nodes
- The sentences are ranked based on the weights to generate the summary



Create a New Summarizer for the Yioop Search Engine Experiment

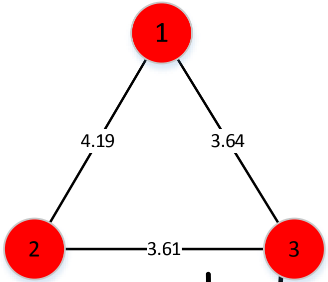
- Adjacency Matrix Computation
  - Each sentence is compared to each other to get their distortion measure

$$Distortion = \frac{Sum}{non\ common\ words}$$

  - Check each word in sentence1 to see if it exists in sentence2. If the word X of sentence1 does not exist in sentence2, square the score of word X and add to the sum and increase the number of not-common words by one
  - In case the word X is common between sentence1 and sentence2, calculate its frequency in sentence2 and subtract it from the score of word X, then square and add to sum
  - Then check the sentence2 to find its not-common words with sentence1, in case the word Y is not in sentence1, square the score of word Y and add to sum and increase the number of not-common words by one
  - At the end, calculate the distortion between sentence1 and sentence2 by dividing sum by the number of not-common words

Create a New Summarizer for the Yioop Search Engine Experiment

- Calculates the adjacency matrix as follows



	Sentence 1	Sentence 2	Sentence 3
Sentence 1	0	4.19	3.64
Sentence 2	4.19	0	3.61
Sentence 3	3.64	3.61	0

- The values are ranked using form of the famous page rank algorithm

$$TR(V_i) = 1 - d + d \sum_{V_j \in I} \frac{TR(Vj)}{Out(Vj)}$$

  - In laments terms, we multiply the adjacency matrix by a even probability vector 10 times

1. I went to the store = 226643006
2. I bought some chips from the store = 226166276
3. I ate the chips on my ride home = 215062659

- The summary would be generated in the order [1, 2, 3]

Create a New Summarizer for the Yioop Search Engine Experiment

- In general the Graph-Based summarizer produced inferior results compared to the Basic summarizer and the Centroid-Based Summarizer

Graph Based ROUGE Result

```
---------------------------------------------
11 ROUGE-1 Average_R: 0.32603 (95%-conf.int. 0.16429 - 0.49286)
11 ROUGE-1 Average_P: 0.30169 (95%-conf.int. 0.15000 - 0.45119)
11 ROUGE-1 Average_F: 0.31165 (95%-conf.int. 0.15619 - 0.46519)
---------------------------------------------
11 ROUGE-2 Average_R: 0.19028 (95%-conf.int. 0.04000 - 0.35667)
11 ROUGE-2 Average_P: 0.18023 (95%-conf.int. 0.04000 - 0.34667)
11 ROUGE-2 Average_F: 0.18470 (95%-conf.int. 0.04000 - 0.35111)
---------------------------------------------
11 ROUGE-3 Average_R: 0.12849 (95%-conf.int. 0.02500 - 0.24500)
11 ROUGE-3 Average_P: 0.12012 (95%-conf.int. 0.02500 - 0.24000)
11 ROUGE-3 Average_F: 0.12370 (95%-conf.int. 0.02500 - 0.24000)
---------------------------------------------
11 ROUGE-4 Average_R: 0.00000 (95%-conf.int. 0.00000 - 0.00000)
11 ROUGE-4 Average_P: 0.00000 (95%-conf.int. 0.00000 - 0.00000)
11 ROUGE-4 Average_F: 0.00000 (95%-conf.int. 0.00000 - 0.00000)
---------------------------------------------
11 ROUGE-L Average_R: 0.20862 (95%-conf.int. 0.08138 - 0.36597)
11 ROUGE-L Average_P: 0.28501 (95%-conf.int. 0.14167 - 0.44405)
11 ROUGE-L Average_F: 0.23625 (95%-conf.int. 0.10295 - 0.39301)
---------------------------------------------
11 ROUGE-W-1.2 Average_R: 0.14754 (95%-conf.int. 0.05208 - 0.27167)
11 ROUGE-W-1.2 Average_P: 0.26469 (95%-conf.int. 0.11303 - 0.43571)
11 ROUGE-W-1.2 Average_F: 0.18541 (95%-conf.int. 0.07116 - 0.33047)
---------------------------------------------
11 ROUGE-S* Average_R: 0.13775 (95%-conf.int. 0.05000 - 0.24191)
11 ROUGE-S* Average_P: 0.11769 (95%-conf.int. 0.04000 - 0.22095)
11 ROUGE-S* Average_F: 0.12457 (95%-conf.int. 0.04019 - 0.22895)
---------------------------------------------
11 ROUGE-SU* Average_R: 0.19791 (95%-conf.int. 0.07333 - 0.33667)
11 ROUGE-SU* Average_P: 0.17406 (95%-conf.int. 0.06500 - 0.31556)
11 ROUGE-SU* Average_F: 0.18243 (95%-conf.int. 0.06702 - 0.32222)
```

CBS ROUGE Result

```
---------------------------------------------
11 ROUGE-1 Average_R: 0.76663 (95%-conf.int. 0.63762 - 0.89333)
11 ROUGE-1 Average_P: 0.67857 (95%-conf.int. 0.53214 - 0.84286)
11 ROUGE-1 Average_F: 0.71577 (95%-conf.int. 0.57970 - 0.86190)
---------------------------------------------
11 ROUGE-2 Average_R: 0.70201 (95%-conf.int. 0.54667 - 0.86333)
11 ROUGE-2 Average_P: 0.61664 (95%-conf.int. 0.43809 - 0.80762)
11 ROUGE-2 Average_F: 0.65163 (95%-conf.int. 0.48667 - 0.83636)
---------------------------------------------
11 ROUGE-3 Average_R: 0.60484 (95%-conf.int. 0.39167 - 0.82833)
11 ROUGE-3 Average_P: 0.53236 (95%-conf.int. 0.31917 - 0.76250)
11 ROUGE-3 Average_F: 0.56079 (95%-conf.int. 0.35098 - 0.79039)
---------------------------------------------
11 ROUGE-4 Average_R: 0.46413 (95%-conf.int. 0.20000 - 0.76667)
11 ROUGE-4 Average_P: 0.42445 (95%-conf.int. 0.16000 - 0.71333)
11 ROUGE-4 Average_F: 0.43933 (95%-conf.int. 0.17500 - 0.73333)
---------------------------------------------
11 ROUGE-L Average_R: 0.54120 (95%-conf.int. 0.39743 - 0.70309)
11 ROUGE-L Average_P: 0.67857 (95%-conf.int. 0.53214 - 0.84286)
11 ROUGE-L Average_F: 0.58289 (95%-conf.int. 0.44879 - 0.72838)
---------------------------------------------
11 ROUGE-W-1.2 Average_R: 0.39129 (95%-conf.int. 0.28736 - 0.50728)
11 ROUGE-W-1.2 Average_P: 0.67325 (95%-conf.int. 0.52159 - 0.83758)
11 ROUGE-W-1.2 Average_F: 0.47857 (95%-conf.int. 0.36753 - 0.60219)
---------------------------------------------
11 ROUGE-S* Average_R: 0.60686 (95%-conf.int. 0.40381 - 0.81667)
11 ROUGE-S* Average_P: 0.49785 (95%-conf.int. 0.28373 - 0.74167)
11 ROUGE-S* Average_F: 0.53466 (95%-conf.int. 0.32789 - 0.76667)
---------------------------------------------
11 ROUGE-SU* Average_R: 0.66706 (95%-conf.int. 0.48421 - 0.84841)
11 ROUGE-SU* Average_P: 0.54722 (95%-conf.int. 0.34603 - 0.76429)
11 ROUGE-SU* Average_F: 0.58952 (95%-conf.int. 0.40240 - 0.79922)
```

Term Frequency Weighting in the Centroid-Based Summarizer Experiment

- Goal of this experiment
  - to add the appropriate frequency weights to the CBS to improve its summary results
  - Analyze the ROUGE results

Term Frequency Weighting in the Centroid-Based Summarizer Experiment

- Utilizing regular expression matching on six categories
  - A
  - H1, H2
  - H3, H4, H5, H6
  - STRONG, B, EM, I, U, DL, OL, UL
  - Title
  - Plain Text = None of the above
- Each category has its own weight
- Extended Centroid-Based summarizer to leverage the weighting method

Term Frequency Weighting in the Centroid-Based Summarizer Experiment

- In general the Centroid-Based summarizer with weighting produced inferior results compared to the Basic summarizer and the Centroid-Based Summarizer

CBS Weighted ROUGE Result

```
---------------------------------------------
11 ROUGE-1 Average_R: 0.24689 (95%-conf.int. 0.12667 - 0.37333)
11 ROUGE-1 Average_P: 0.21218 (95%-conf.int. 0.11607 - 0.31071)
11 ROUGE-1 Average_F: 0.21092 (95%-conf.int. 0.11915 - 0.30427)
---------------------------------------------
11 ROUGE-2 Average_R: 0.14330 (95%-conf.int. 0.05000 - 0.25000)
11 ROUGE-2 Average_P: 0.08230 (95%-conf.int. 0.02857 - 0.14286)
11 ROUGE-2 Average_F: 0.10366 (95%-conf.int. 0.03636 - 0.18182)
---------------------------------------------
11 ROUGE-3 Average_R: 0.06350 (95%-conf.int. 0.03333 - 0.13333)
11 ROUGE-3 Average_P: 0.03175 (95%-conf.int. 0.01667 - 0.06667)
11 ROUGE-3 Average_F: 0.04233 (95%-conf.int. 0.02222 - 0.08889)
---------------------------------------------
11 ROUGE-4 Average_R: 0.00000 (95%-conf.int. 0.00000 - 0.00000)
11 ROUGE-4 Average_P: 0.00000 (95%-conf.int. 0.00000 - 0.00000)
11 ROUGE-4 Average_F: 0.00000 (95%-conf.int. 0.00000 - 0.00000)
---------------------------------------------
11 ROUGE-L Average_R: 0.18470 (95%-conf.int. 0.09365 - 0.29961)
11 ROUGE-L Average_P: 0.21218 (95%-conf.int. 0.11607 - 0.31071)
11 ROUGE-L Average_F: 0.18124 (95%-conf.int. 0.10222 - 0.26444)
---------------------------------------------
11 ROUGE-W-1.2 Average_R: 0.11842 (95%-conf.int. 0.05341 - 0.20543)
11 ROUGE-W-1.2 Average_P: 0.17366 (95%-conf.int. 0.08750 - 0.27679)
11 ROUGE-W-1.2 Average_F: 0.12677 (95%-conf.int. 0.06425 - 0.20894)
---------------------------------------------
11 ROUGE-S* Average_R: 0.07818 (95%-conf.int. 0.03000 - 0.14000)
11 ROUGE-S* Average_P: 0.02791 (95%-conf.int. 0.01071 - 0.05000)
11 ROUGE-S* Average_F: 0.04038 (95%-conf.int. 0.01579 - 0.07427)
---------------------------------------------
11 ROUGE-SU* Average_R: 0.13299 (95%-conf.int. 0.05656 - 0.22476)
11 ROUGE-SU* Average_P: 0.09831 (95%-conf.int. 0.03429 - 0.20513)
11 ROUGE-SU* Average_F: 0.08041 (95%-conf.int. 0.03828 - 0.12862)
```

CBS ROUGE Result

```
---------------------------------------------
11 ROUGE-1 Average_R: 0.76663 (95%-conf.int. 0.63762 - 0.89333)
11 ROUGE-1 Average_P: 0.67857 (95%-conf.int. 0.53214 - 0.84286)
11 ROUGE-1 Average_F: 0.71577 (95%-conf.int. 0.57970 - 0.86190)
---------------------------------------------
11 ROUGE-2 Average_R: 0.70201 (95%-conf.int. 0.54667 - 0.86333)
11 ROUGE-2 Average_P: 0.61664 (95%-conf.int. 0.43809 - 0.80762)
11 ROUGE-2 Average_F: 0.65163 (95%-conf.int. 0.48667 - 0.83636)
---------------------------------------------
11 ROUGE-3 Average_R: 0.60484 (95%-conf.int. 0.39167 - 0.82833)
11 ROUGE-3 Average_P: 0.53236 (95%-conf.int. 0.31917 - 0.76250)
11 ROUGE-3 Average_F: 0.56079 (95%-conf.int. 0.35098 - 0.79039)
---------------------------------------------
11 ROUGE-4 Average_R: 0.46413 (95%-conf.int. 0.20000 - 0.76667)
11 ROUGE-4 Average_P: 0.42445 (95%-conf.int. 0.16000 - 0.71333)
11 ROUGE-4 Average_F: 0.43933 (95%-conf.int. 0.17500 - 0.73333)
---------------------------------------------
11 ROUGE-L Average_R: 0.54120 (95%-conf.int. 0.39743 - 0.70309)
11 ROUGE-L Average_P: 0.67857 (95%-conf.int. 0.53214 - 0.84286)
11 ROUGE-L Average_F: 0.58289 (95%-conf.int. 0.44879 - 0.72838)
---------------------------------------------
11 ROUGE-W-1.2 Average_R: 0.39129 (95%-conf.int. 0.28736 - 0.50728)
11 ROUGE-W-1.2 Average_P: 0.67325 (95%-conf.int. 0.52159 - 0.83758)
11 ROUGE-W-1.2 Average_F: 0.47857 (95%-conf.int. 0.36753 - 0.60219)
---------------------------------------------
11 ROUGE-S* Average_R: 0.60686 (95%-conf.int. 0.40381 - 0.81667)
11 ROUGE-S* Average_P: 0.49785 (95%-conf.int. 0.28373 - 0.74167)
11 ROUGE-S* Average_F: 0.53466 (95%-conf.int. 0.32789 - 0.76667)
---------------------------------------------
11 ROUGE-SU* Average_R: 0.66706 (95%-conf.int. 0.48421 - 0.84841)
11 ROUGE-SU* Average_P: 0.54722 (95%-conf.int. 0.34603 - 0.76429)
11 ROUGE-SU* Average_F: 0.58952 (95%-conf.int. 0.40240 - 0.79922)
```

Improve the ROUGE Results for Centroid-based Weighting Summarizer Experiment

- Goal of this experiment
  - to create a Centroid-based weighting summarizer to produce better results
  - implement the algorithm in PHP and integrated into the Yioop search engine
  - Analyze the ROUGE results

- Created a new summarizer called Centroid-based Weighting Summarizer (CBWS)

Improve the ROUGE Results for Centroid-based Weighting Summarizer Experiment

- The main idea is that the sentences closes to the average sentence are the most important

- Find the sentences that are closest to the average sentence
  - Treating each sentence's terms like a matrix
  - Add each term frequency column value and divide each value by the total number of rows

- Compare all of the other sentences to it by calculating the dot product of the sentences and the average sentence

Improve the ROUGE Results for Centroid-based Weighting Summarizer Experiment

- Get the term frequencies in each sentence

I:1	went:1	to:1	the:1	store:1			
I:1	bought:1	some:1	chips:1	from:1	the:1	store:1	
I:1	ate:1	the:1	chips:1	on:1	my:1	ride:1	home:1

- Array of all frequencies summed up

I:3	ate:1	bought:1	chips:2	from:1	home:1	my:1	on:1	ride:1	some:1	store:2	the:3	to:1	went:1

- This represents the average sentence

I:1	ate:1/3	bought:1/3	chips:2/3	from:1/3	home:1/3	my:1/3	on:1/3	ride:1/3	some:1/3	store:2/3	the:1	to:1/3	went:1/3

Improve the ROUGE Results for Centroid-based Weighting Summarizer Experiment

- Average sentence dot product values of each sentence
  1. I went to the store = 3 1/3
  2. I bought some chips from the store = 4 1/3
  3. I ate the chips on my ride home = 4 1/3

- The summary would be generated in the order [2, 3, 1]

Improve the ROUGE Results for Centroid-based Weighting Summarizer Experiment

- In general the Centroid-Based Weighted summarizer produced did not perform as well as the Basic summarizer or the Centroid-Based Summarizer

CBS Weighted ROUGE Result

```
---------------------------------------------
11 ROUGE-1 Average_R: 0.24689 (95%-conf.int. 0.12667 - 0.37333)
11 ROUGE-1 Average_P: 0.21218 (95%-conf.int. 0.11607 - 0.31071)
11 ROUGE-1 Average_F: 0.21092 (95%-conf.int. 0.11915 - 0.30427)
---------------------------------------------
11 ROUGE-2 Average_R: 0.14330 (95%-conf.int. 0.05000 - 0.25000)
11 ROUGE-2 Average_P: 0.08230 (95%-conf.int. 0.02857 - 0.14286)
11 ROUGE-2 Average_F: 0.10366 (95%-conf.int. 0.03636 - 0.18182)
---------------------------------------------
11 ROUGE-3 Average_R: 0.06350 (95%-conf.int. 0.03333 - 0.13333)
11 ROUGE-3 Average_P: 0.03175 (95%-conf.int. 0.01667 - 0.06667)
11 ROUGE-3 Average_F: 0.04233 (95%-conf.int. 0.02222 - 0.08889)
---------------------------------------------
11 ROUGE-4 Average_R: 0.00000 (95%-conf.int. 0.00000 - 0.00000)
11 ROUGE-4 Average_P: 0.00000 (95%-conf.int. 0.00000 - 0.00000)
11 ROUGE-4 Average_F: 0.00000 (95%-conf.int. 0.00000 - 0.00000)
---------------------------------------------
11 ROUGE-L Average_R: 0.18470 (95%-conf.int. 0.09365 - 0.29961)
11 ROUGE-L Average_P: 0.21218 (95%-conf.int. 0.11607 - 0.31071)
11 ROUGE-L Average_F: 0.18124 (95%-conf.int. 0.10222 - 0.26444)
---------------------------------------------
11 ROUGE-W-1.2 Average_R: 0.11842 (95%-conf.int. 0.05341 - 0.20543)
11 ROUGE-W-1.2 Average_P: 0.17366 (95%-conf.int. 0.08750 - 0.27679)
11 ROUGE-W-1.2 Average_F: 0.12677 (95%-conf.int. 0.06425 - 0.20894)
---------------------------------------------
11 ROUGE-S* Average_R: 0.07818 (95%-conf.int. 0.03000 - 0.14000)
11 ROUGE-S* Average_P: 0.02791 (95%-conf.int. 0.01071 - 0.05000)
11 ROUGE-S* Average_F: 0.04038 (95%-conf.int. 0.01579 - 0.07427)
---------------------------------------------
11 ROUGE-SU* Average_R: 0.13299 (95%-conf.int. 0.05656 - 0.22476)
11 ROUGE-SU* Average_P: 0.09831 (95%-conf.int. 0.03429 - 0.20513)
11 ROUGE-SU* Average_F: 0.08041 (95%-conf.int. 0.03828 - 0.12862)
```

CBS ROUGE Result

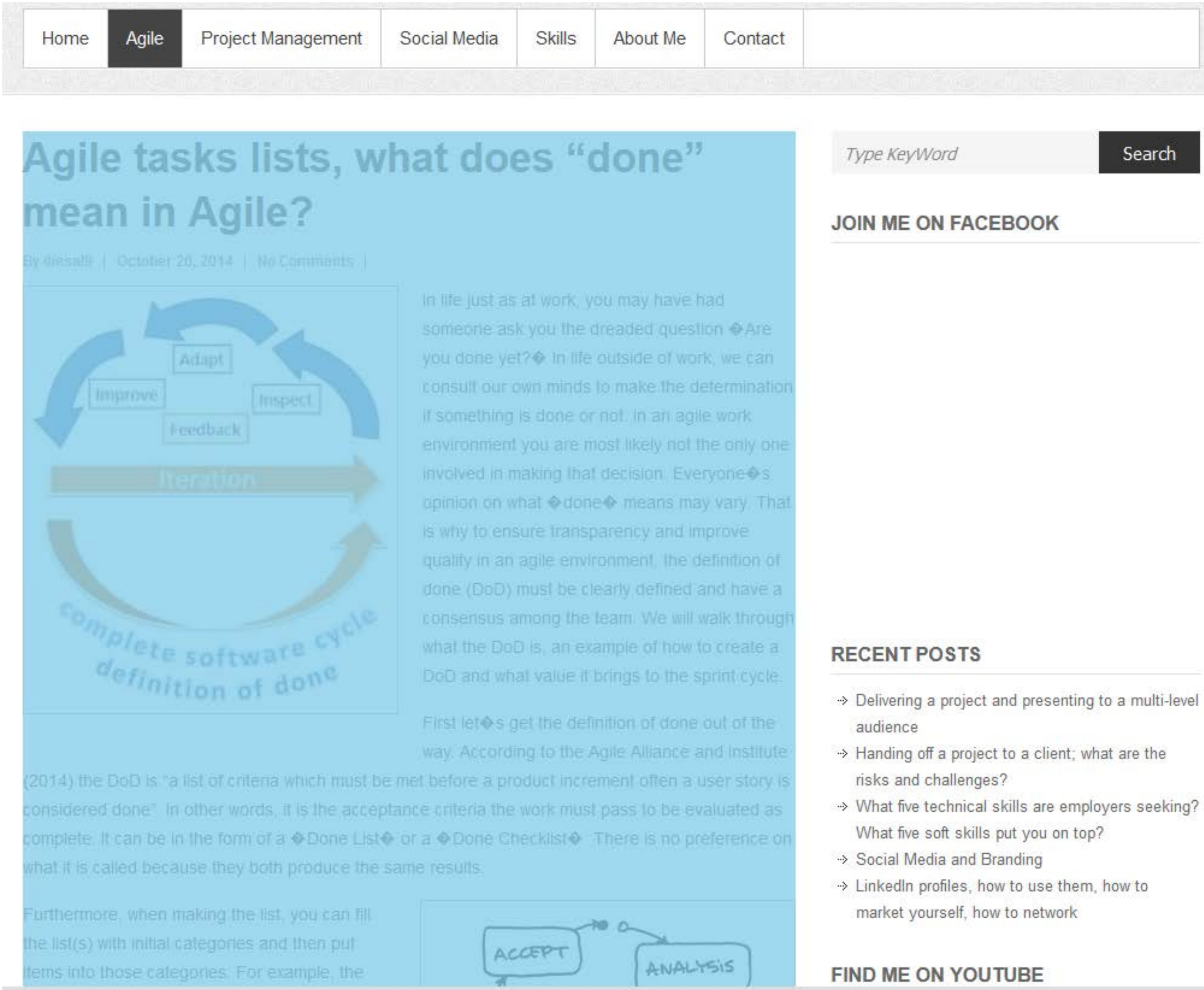
```
---------------------------------------------
11 ROUGE-1 Average_R: 0.76663 (95%-conf.int. 0.63762 - 0.89333)
11 ROUGE-1 Average_P: 0.67857 (95%-conf.int. 0.53214 - 0.84286)
11 ROUGE-1 Average_F: 0.71577 (95%-conf.int. 0.57970 - 0.86190)
---------------------------------------------
11 ROUGE-2 Average_R: 0.70201 (95%-conf.int. 0.54667 - 0.86333)
11 ROUGE-2 Average_P: 0.61664 (95%-conf.int. 0.43809 - 0.80762)
11 ROUGE-2 Average_F: 0.65163 (95%-conf.int. 0.48667 - 0.83636)
---------------------------------------------
11 ROUGE-3 Average_R: 0.60484 (95%-conf.int. 0.39167 - 0.82833)
11 ROUGE-3 Average_P: 0.53236 (95%-conf.int. 0.31917 - 0.76250)
11 ROUGE-3 Average_F: 0.56079 (95%-conf.int. 0.35098 - 0.79039)
---------------------------------------------
11 ROUGE-4 Average_R: 0.46413 (95%-conf.int. 0.20000 - 0.76667)
11 ROUGE-4 Average_P: 0.42445 (95%-conf.int. 0.16000 - 0.71333)
11 ROUGE-4 Average_F: 0.43933 (95%-conf.int. 0.17500 - 0.73333)
---------------------------------------------
11 ROUGE-L Average_R: 0.54120 (95%-conf.int. 0.39743 - 0.70309)
11 ROUGE-L Average_P: 0.67857 (95%-conf.int. 0.53214 - 0.84286)
11 ROUGE-L Average_F: 0.58289 (95%-conf.int. 0.44879 - 0.72838)
---------------------------------------------
11 ROUGE-W-1.2 Average_R: 0.39129 (95%-conf.int. 0.28736 - 0.50728)
11 ROUGE-W-1.2 Average_P: 0.67325 (95%-conf.int. 0.52159 - 0.83758)
11 ROUGE-W-1.2 Average_F: 0.47857 (95%-conf.int. 0.36753 - 0.60219)
---------------------------------------------
11 ROUGE-S* Average_R: 0.60686 (95%-conf.int. 0.40381 - 0.81667)
11 ROUGE-S* Average_P: 0.49785 (95%-conf.int. 0.28373 - 0.74167)
11 ROUGE-S* Average_F: 0.53466 (95%-conf.int. 0.32789 - 0.76667)
---------------------------------------------
11 ROUGE-SU* Average_R: 0.66706 (95%-conf.int. 0.48421 - 0.84841)
11 ROUGE-SU* Average_P: 0.54722 (95%-conf.int. 0.34603 - 0.76429)
11 ROUGE-SU* Average_F: 0.58952 (95%-conf.int. 0.40240 - 0.79922)
```

Improve the ROUGE Results for Centroid-based Weighting Summarizer Experiment

- Pondered an idea about using CMS detection to focus on the important content instead
  - The theory was most CMS generated web pages are full of unnecessary content
  - If we target the important content, the ROUGE results would improve
  - For example all of the WordPress pages used for my blog have the important content in a DIV tag with the id of content
  - All other content is not important to summarization
  - Extending even further, we developed a mechanism to pull unnecessary content from within the target area

- CMS detection works by regular expression matching on unique patterns in the head tag of the HTML document

Improve the ROUGE Results for Centroid-based Weighting Summarizer Experiment



Improve the ROUGE Results for Centroid-based Weighting Summarizer Experiment

- Configured Yioop with the each summarizer and generated summaries from the set of ten web pages again
- Ran ROUGE on the summaries to evaluate their quality
- Compared the ROUGE results

Improve the ROUGE Results for Centroid-based Weighting Summarizer Experiment

- The test scores went up across the board.

CBWS ROUGE Result Before

```
---------------------------------------------
11 ROUGE-1 Average_R: 0.66665 (95%-conf.int. 0.56143 - 0.78024)
11 ROUGE-1 Average_P: 0.59174 (95%-conf.int. 0.44167 - 0.75417)
11 ROUGE-1 Average_F: 0.61670 (95%-conf.int. 0.49659 - 0.74103)
---------------------------------------------
11 ROUGE-2 Average_R: 0.59220 (95%-conf.int. 0.47333 - 0.72667)
11 ROUGE-2 Average_P: 0.52933 (95%-conf.int. 0.35952 - 0.71357)
11 ROUGE-2 Average_F: 0.54518 (95%-conf.int. 0.41064 - 0.68843)
---------------------------------------------
11 ROUGE-3 Average_R: 0.47023 (95%-conf.int. 0.32833 - 0.63333)
11 ROUGE-3 Average_P: 0.44098 (95%-conf.int. 0.24762 - 0.65000)
11 ROUGE-3 Average_F: 0.43500 (95%-conf.int. 0.27667 - 0.60444)
---------------------------------------------
11 ROUGE-4 Average_R: 0.21573 (95%-conf.int. 0.00000 - 0.45000)
11 ROUGE-4 Average_P: 0.20247 (95%-conf.int. 0.00000 - 0.42667)
11 ROUGE-4 Average_F: 0.20307 (95%-conf.int. 0.00000 - 0.40952)
---------------------------------------------
11 ROUGE-L Average_R: 0.46695 (95%-conf.int. 0.35840 - 0.59688)
11 ROUGE-L Average_P: 0.59174 (95%-conf.int. 0.44167 - 0.75417)
11 ROUGE-L Average_F: 0.50737 (95%-conf.int. 0.38835 - 0.65018)
---------------------------------------------
11 ROUGE-W-1.2 Average_R: 0.27444 (95%-conf.int. 0.16971 - 0.37699)
11 ROUGE-W-1.2 Average_P: 0.43489 (95%-conf.int. 0.25417 - 0.63333)
11 ROUGE-W-1.2 Average_F: 0.32801 (95%-conf.int. 0.19819 - 0.46248)
---------------------------------------------
11 ROUGE-S* Average_R: 0.43092 (95%-conf.int. 0.28524 - 0.60095)
11 ROUGE-S* Average_P: 0.38901 (95%-conf.int. 0.19166 - 0.60952)
11 ROUGE-S* Average_F: 0.37903 (95%-conf.int. 0.22536 - 0.54553)
---------------------------------------------
11 ROUGE-SU* Average_R: 0.52030 (95%-conf.int. 0.38368 - 0.67646)
11 ROUGE-SU* Average_P: 0.44916 (95%-conf.int. 0.25394 - 0.65976)
11 ROUGE-SU* Average_F: 0.45378 (95%-conf.int. 0.30683 - 0.60927)
```

CBWS ROUGE Result After

```
---------------------------------------------
11 ROUGE-1 Average_R: 0.80863 (95%-conf.int. 0.70071 - 0.89833)
11 ROUGE-1 Average_P: 0.79195 (95%-conf.int. 0.69143 - 0.88214)
11 ROUGE-1 Average_F: 0.79771 (95%-conf.int. 0.69705 - 0.88261)
---------------------------------------------
11 ROUGE-2 Average_R: 0.71814 (95%-conf.int. 0.58000 - 0.83667)
11 ROUGE-2 Average_P: 0.70826 (95%-conf.int. 0.57000 - 0.83000)
11 ROUGE-2 Average_F: 0.71019 (95%-conf.int. 0.57272 - 0.82778)
---------------------------------------------
11 ROUGE-3 Average_R: 0.63218 (95%-conf.int. 0.45833 - 0.78833)
11 ROUGE-3 Average_P: 0.62551 (95%-conf.int. 0.45167 - 0.79000)
11 ROUGE-3 Average_F: 0.62487 (95%-conf.int. 0.45143 - 0.78135)
---------------------------------------------
11 ROUGE-4 Average_R: 0.45574 (95%-conf.int. 0.20000 - 0.68333)
11 ROUGE-4 Average_P: 0.46348 (95%-conf.int. 0.21667 - 0.71667)
11 ROUGE-4 Average_F: 0.45503 (95%-conf.int. 0.21333 - 0.68952)
---------------------------------------------
11 ROUGE-L Average_R: 0.55198 (95%-conf.int. 0.44967 - 0.67684)
11 ROUGE-L Average_P: 0.79195 (95%-conf.int. 0.69143 - 0.88214)
11 ROUGE-L Average_F: 0.63472 (95%-conf.int. 0.55068 - 0.73380)
---------------------------------------------
11 ROUGE-W-1.2 Average_R: 0.35800 (95%-conf.int. 0.27424 - 0.46042)
11 ROUGE-W-1.2 Average_P: 0.68562 (95%-conf.int. 0.57119 - 0.79762)
11 ROUGE-W-1.2 Average_F: 0.45893 (95%-conf.int. 0.36787 - 0.56848)
---------------------------------------------
11 ROUGE-S* Average_R: 0.65210 (95%-conf.int. 0.48857 - 0.79953)
11 ROUGE-S* Average_P: 0.62371 (95%-conf.int. 0.46000 - 0.78143)
11 ROUGE-S* Average_F: 0.62855 (95%-conf.int. 0.47200 - 0.77133)
---------------------------------------------
11 ROUGE-SU* Average_R: 0.72396 (95%-conf.int. 0.57849 - 0.84921)
11 ROUGE-SU* Average_P: 0.69407 (95%-conf.int. 0.55905 - 0.82167)
11 ROUGE-SU* Average_F: 0.70080 (95%-conf.int. 0.56488 - 0.81945)
```

A Numerically Stable Lanczos Text Summarization Algorithm Experiment

- Goal of this experiment
  - to create a numerically stable version of the Lanczos text summarization method
  - implement the algorithm in PHP and integrated into the Yioop search engine
  - Analyze the ROUGE results

  - Numeric stability
    - how does it respond with small numerical errors

A Numerically Stable Lanczos Text Summarization Algorithm Experiment

- Lanczos is a fast method for computing SVD. The Lanczos method is "a technique that can be used to solve certain large, sparse, symmetric eigenproblems $Ax = \lambda x$" [Golub and Van Loan 2012].

- The Lanczos algorithm solves these eigenproblems by splitting the initial frequency matrix into three matrices. "An orthogonal matrix U, a diagonal matrix S, and the transpose of an orthogonal matrix V" [Kim 2010].

A Numerically Stable Lanczos Text Summarization Algorithm Experiment

- Create a term frequency matrix per sentence

	Sentence 1	Sentence 2	Sentence 3
I	1	1	1
ate	0	0	1
bought	0	1	0
chips	0	1	1
from	0	1	0
home	0	0	1
my	0	0	1
on	0	0	1
ride	0	0	1
some	0	1	0
store	1	1	0
the	1	1	1
to	1	1	0
went	1	0	0

A Numerically Stable Lanczos Text Summarization Algorithm Experiment

- Creates a transpose matrix

	I	ate	bought	chips	fro m	home	my	on	ride	some	stor e	the	to	went
Sentence 1	1	0	0	0	0	0	0	0	0	0	1	1	1	1
Sentence 2	1	0	1	1	1	0	0	0	0	1	1	1	1	1
Sentence 3	1	1	0	1	0	1	1	1	1	0	0	1	0	0

A Numerically Stable Lanczos Text Summarization Algorithm Experiment

- Multiples the original matrix with the transpose matrix to create an orthogonal matrix

3	1	1	2	1	1	1	1	1	1	2	3	2	1
1	1	0	1	0	1	1	1	1	0	0	1	0	0
1	0	1	1	1	0	0	0	0	1	1	1	1	0
2	1	1	2	1	1	1	1	1	1	1	2	1	0
1	0	1	1	1	0	0	0	0	1	1	1	1	0
1	1	0	1	0	1	1	1	1	0	0	1	0	0
1	1	0	1	0	1	1	1	1	0	0	1	0	0
1	1	0	1	0	1	1	1	1	0	0	1	0	0
1	1	0	1	0	1	1	1	1	0	0	1	0	0
1	0	1	1	1	0	0	0	0	1	1	1	1	0
2	0	1	1	1	0	0	0	0	1	2	2	2	1
3	1	1	2	1	1	1	1	1	1	2	3	2	1
2	0	1	1	1	0	0	0	0	1	2	2	2	1
1	0	0	0	0	0	0	0	0	0	1	1	1	1

A Numerically Stable Lanczos Text Summarization Algorithm Experiment

- An orthogonal matrix is converted into a tridiagonal matrix

3	5.477	0	0	0	0	0	0	0	0	0	0	0	0
5.477	10.267	0.722	0	0	0	0	0	0	0	0	0	0	0
0	0.722	2.271	0.451	0	0	0	0	0	0	0	0	0	0
0	0	0.451	5.463	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

A Numerically Stable Lanczos Text Summarization Algorithm Experiment

- The Lanczos algorithm also creates a "q" matrix

1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0.183	-0.63	-0.344	-0.190	-0.091	-0.10	-0.148	0.148	-0.148	0.148	-0.148	0.148	-0.148
0	0.183	0.410	-0.112	-0.197	-0.237	0.339	-0.338	0.338	-0.338	-0.338	0.338	-0.338	0.338
0	0.365	0.347	-0.203	0.704	-0.133	0.430	-0.077	0.076	-0.076	0.076	-0.076	0.076	-0.076
0	0.183	0.410	0.141	-0.107	-0.206	-0.230	0.438	-0.437	0.437	-0.437	0.437	-0.437	0.437
0	0.183	-0.063	-0.344	-0.185	-0.100	-0.003	-0.049	0.049	-0.049	0.049	-0.049	0.049	-0.049
0	0.183	-0.063	-0.344	-0.185	-0.100	-0.003	-0.049	0.049	-0.049	0.049	-0.049	0.049	-0.049
0	0.183	-0.063	-0.344	-0.185	-0.100	-0.003	-0.049	0.049	-0.049	0.049	-0.049	0.049	-0.049
0	0.183	-0.063	-0.344	-0.168	-0.131	0.020	0.297	-0.297	0.297	-0.297	0.297	-0.297	-0.049
0	0.183	0.410	0.141	-0.167	-0.100	-0.311	-0.749	0.750	-0.750	0.750	-0.750	0.750	-0.750
0	0.365	-0.126	0.360	-0.264	-0.009	0.389	-0.014	0.013	-0.013	0.013	-0.013	0.013	-0.013
0	0.548	-0.189	0.016	0.209	0.655	-0.430	0.077	-0.076	0.076	-0.076	0.076	-0.076	0.076
0	0.365	-0.126	0.360	-0.264	-0.009	0.389	-0.014	0.013	-0.013	0.013	-0.013	0.013	-0.013
0	0.183	-0.536	0.219	0.319	-0.637	-0.348	-0.050	0.050	-0.050	0.050	-0.050	0.050	-0.050

A Numerically Stable Lanczos Text Summarization Algorithm Experiment

- Performs QR factorization to expose the eigenvalues
  - 13.248
  - 5.25
  - 2.27
  - 11 zeros

13.248	0	0	0	0	0	0	0	0	0	0	0	0	0
0	5.52	0	0	0	0	0	0	0	0	0	0	0	0
0	0	2.27	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

A Numerically Stable Lanczos Text Summarization Algorithm Experiment

- Creates a new tridiagonal matrices by subtracting the eigenvalues from each diagonal value from the tridiagonal matrix created earlier and then inverts it

- Then it takes the first row of each matrix and puts them into the columns of another matrix

A Numerically Stable Lanczos Text Summarization Algorithm Experiment

0.470	0.014	-0.173	-0.450	-0.450	-0.450	-0.450	-0.450	-0.450	-0.450	-0.450	-0.450	-0.450	-0.450
00880	0.006	0.024	-0.877	-0.877	-0.877	-0.877	-0.877	-0.877	-0.877	-0.877	-0.877	-0.877	-0.877
0.062	-0.136	0.976	-0.165	-0.165	-0.165	-0.165	-0.165	-0.165	-0.165	-0.165	-0.165	-0.165	-0.165
0.004	-0.991	-0.136	-0.032	-0.032	-0.032	-0.032	-0.032	-0.032	-0.032	-0.032	-0.032	-0.032	-0.032
0	-0	-0	0	0	0	0	0	0	0	0	0	0	0
0	-0	-0	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001	-0.001
0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
0	-0	-0	0.023	0.023	0.023	0.023	0.023	0.023	0.023	0.023	0.023	0.023	0.023
0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0
0	-0	-0	0	0	0	0	0	0	0	0	0	0	0
0	-0	-0	0	0	0	0	0	0	0	0	0	0	0
0	-0	-0	0	0	0	0	0	0	0	0	0	0	0
0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0	-0

A Numerically Stable Lanczos Text Summarization Algorithm Experiment

- The U, S and V matrices are calculated
  - U = matrix q times the inverted eigenvalue matrix
  - S = tridiagonal matrix of the square root of the eigenvalues
  - V = a transpose matrix of the original matrix times and U

A Numerically Stable Lanczos Text Summarization Algorithm Experiment

- The highest values from the top row of the V matrix reflect the sentence order
    1. I went to the store = 0.467
    2. I bought some chips from the store = 0.679
    3. I ate the chips on my ride home = 0.566

- The summary would be generated in the order [2, 3, 1]

Sentence 1	Sentence 2	Sentence 3
0.467	0.679	0.566
-0.336	-0.456	0.824
-0.818	0.575	-0.015
-0.442	-0.715	-0.542
-0.442	-0.715	-0.542
-0.442	-0.715	-0.542
-0.442	-0.715	-0.542
-0.442	-0.715	-0.542
-0.442	-0.715	-0.542
-0.442	-0.715	-0.542
-0.442	-0.715	-0.542
-0.442	-0.715	-0.542
-0.442	-0.715	-0.542
-0.442	-0.715	-0.542

A Numerically Stable Lanczos Text Summarization Algorithm Experiment

- The problem:
  - "The central problem is a loss of orthogonality among the Lanczos vectors that the iteration produces" [Golub and Van Loan 2012].
- Researched using a modified version of Latent Semantic Analysis (LSA)
  - All of the approaches found failed to address the Single Value Decomposition (SVD) problem
- Researched using Block Lanczos
  - Create mutually orthogonal matrices provided none of them are rank-deficient
  - If any are rank-deficient then the matrices produced will lose orthogonality and become numerically unstable too
- Unable to find a numerically stable Lanczos algorithm, it is determined that it is not a viable approach for summarization

Test Yioop Summarizers Against a Large Data Set Experiment

- Goal of this experiment
  - to find a large document set
  - write code to automate the testing of all summarizers against the large data set
  - Analyze the ROUGE results

- The large document set we used is from the Document Understanding Conference (DUC) in 2007

Test Yioop Summarizers Against a Large Data Set Experiment

- Document Understanding Conference (DUC)
  - event organized by the NIST that consist of a many summarization evaluations
  - "Its goal is to further progress in automatic text summarizations and enable researchers to participate in large-scale experiments in both the development and evaluations of summarization systems" [NIST 2011]
  - Held between 2001-2007, now it is called Text Analytic Conference (TAC)
  - Chose the latest DUC data from 2007
  - The DUC data is broken down into 10 topics with 25 documents per topic
  - The documents are broken in to 3 sets: A, B, C
  - Set A has 10 documents
  - Set B has 8 documents
  - Set C has 7 documents
  - 4 human summaries per document set

Test Yioop Summarizers Against a Large Data Set Experiment

- Organized the DUC documents into a format that could be summarized
- Configured Yioop with the each summarizer and generated summaries from the set of DUC documents
- Ran ROUGE on the summaries to evaluate their quality
- Compared the ROUGE results

Test Yioop Summarizers Against a Large Data Set Experiment

- Each set {A, B, C} contained 38 summaries (4 ours, 4 human and 30 candidates)

	CBS	BS	CBWS	GBS
Lowest Rank	18	21	24	22
Lowest Rank Test	R1R	RLF;RWF	R3F	RLF
Median Rank	37	38	36	34
Average Rank	32.0952381	33.28571429	32.42857143	31.85714286
Highest Rank	37	38	36	35

Improving Text Summarization using Automatic Sentence Compression Experiment

- Goal of this experiment
  - to find a simple automatic sentence compression framework to improve text summarization
  - implement the algorithm in PHP and integrated into the Yioop search engine
  - Analyze the ROUGE results

Improving Text Summarization using Automatic Sentence Compression Experiment

- Chose to implement the sentence trimming method mentioned in *Back to Basics: CLASSY 2006*
  - look for specific words, phrases or clauses and remove them
  - Chose to implement four of the seven

- I remove many adverbs and all conjunctions, including phrases such as "As a matter of fact," and "At this point," that occur at the start of a sentence

- We remove a small selections of words that occur in the middle of a sentence, such as ", however," and ", also," (not always requiring the commas)

- We remove ages such as ", 51," or ", aged 24,“

- We remove relative clause attributives (clauses beginning with "who(m)", "which", "when", and "where") wherever possible.

Improving Text Summarization using Automatic Sentence Compression Experiment

- The DUC data has 120 documents to summarize and seven ROUGE tests to perform
- A total of 840 tests for each summarizer
- Out of the 840 tests some of them had different results
  - 236 tests for the Basic Summarizer (BASIC)
  - 350 tests for the Centroid-based Summarizer (CBS)
  - 197 tests for the Centroid-based Weighted Summarizer (CBWS)
  - 266 tests for the Graph-based Summarizer (GBS)

Improving Text Summarization using Automatic Sentence Compression Experiment

- Out of the 236 tests for the BASIC automatic sentence compression lost 134 to 82
- Out of the 350 tests for the CBS automatic sentence compression lost 215 to 135
- Out of the 197 tests for the CBWS automatic sentence compression lost 143 to 54
- Out of the 266 tests for the GBS automatic sentence compression lost 175 to 91

Conclusion

- I was able to
  - created a stemmer in Dutch
  - created two new summarizers
  - evaluated the summarizers against a large data set
  - created a basic sentence compression framework
  - improve the ROUGE results using CMS Detection
  - become very knowledgeable in the topic of ATS.

  - The factor that effected the ROUGE results the most was detecting the CMS that generated the content
    - detecting the CMS improved the results by 10% to 20%
    - the graph based or the average sentence approach was fun but did not improve results

Questions and Comments

Demo