

# Summarization - Round 1

## Team E2JK++

Eric Lindberg

John Greve

Josh Mathias

Kekoa Riggin

Presentation requirements (from the D2 reqs)

[added 4/22 jgreve, thought a check list to work on would be helpful]

NOTE: for document creation. Remove (or at least make hidden) before shipping.

Figured we write this first and back-port into a paper.

git location: `../doc/D2_presentation.pdf`

Your presentation may be prepared in any computer-projectable format, including HTML, PDF, PPT, and Word. Your presentation should take about 10 minutes to cover your main content, including:

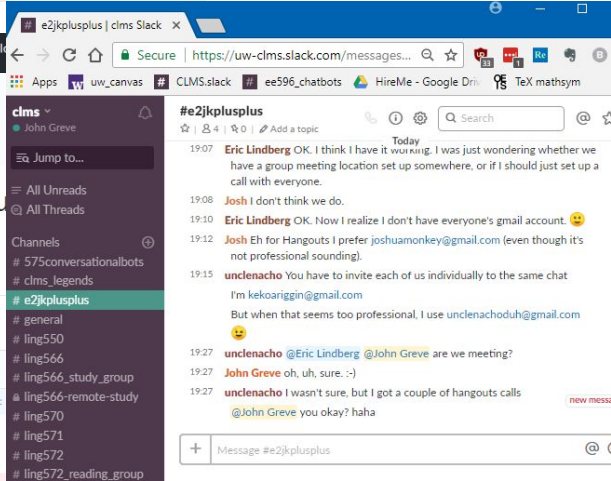
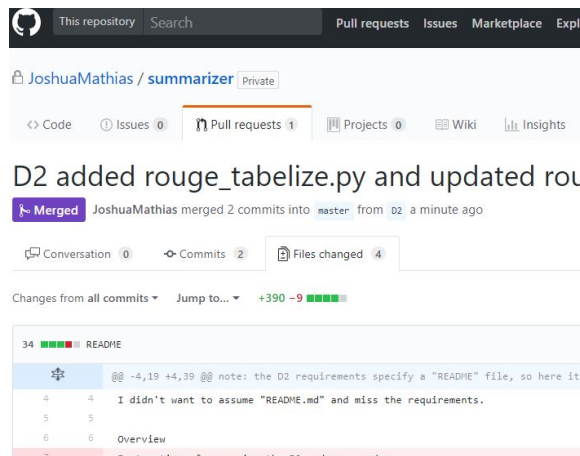
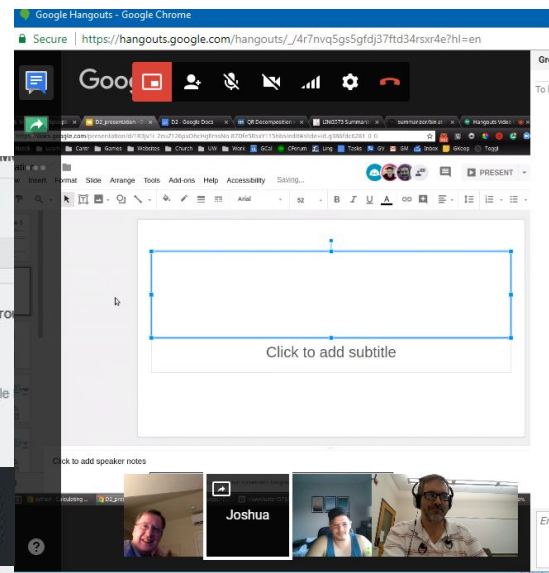
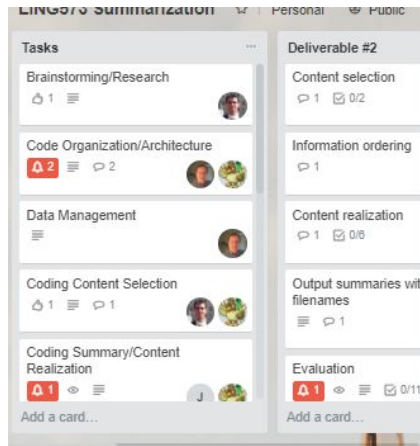
- System architecture
- Content selection
- Information Ordering
- Content Realization
- Issues and successes
- Related reading which influenced your approach

Your presentation should be deposited in your doc directory, but it is not due until the actual presentation time.

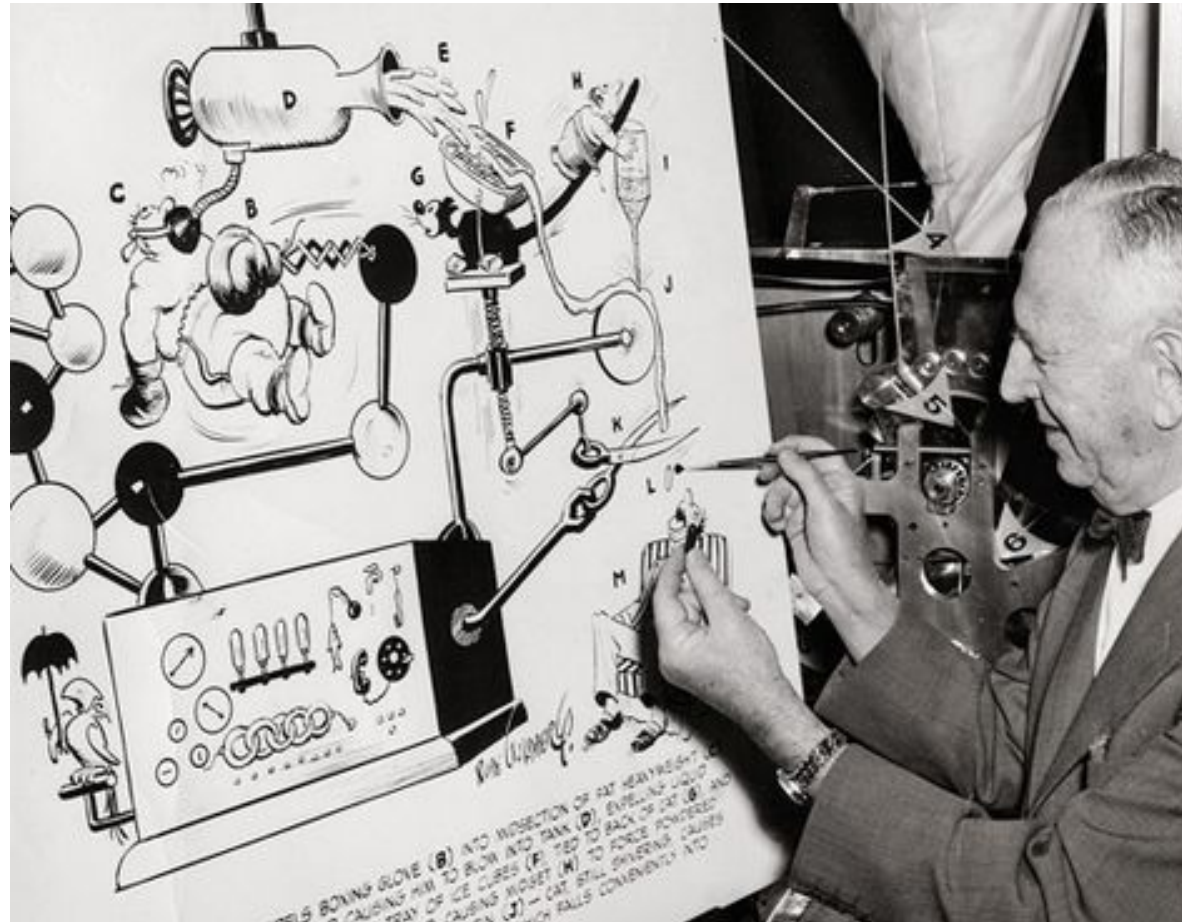
\*\*\* You may continue working on it after the main deliverable is due.

# Communication

- Trello
  - Tasks
- Slack
  - Chat
- Hangouts
  - Video calls
- Git
  - Pull requests
- Bi-weekly standups
  - In-person reporting, planning



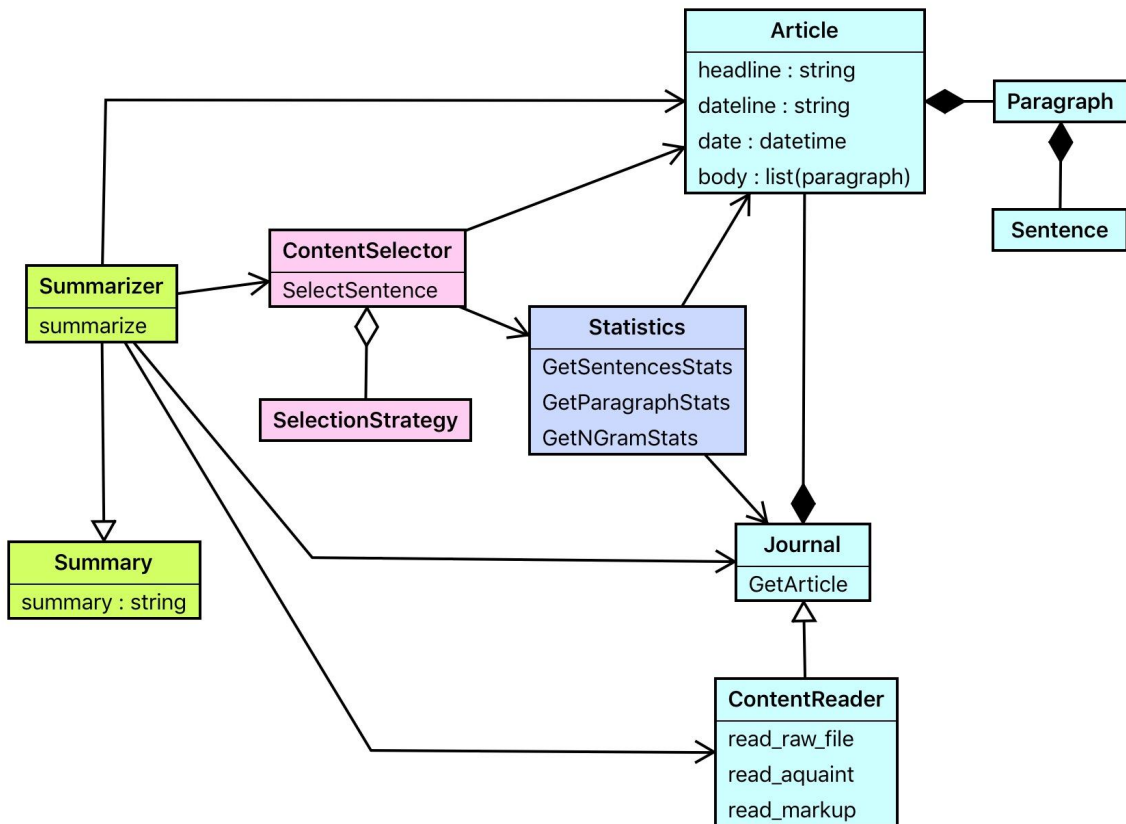
# System Architecture



# System Architecture

## System Architecture Diagram (the right one this time)

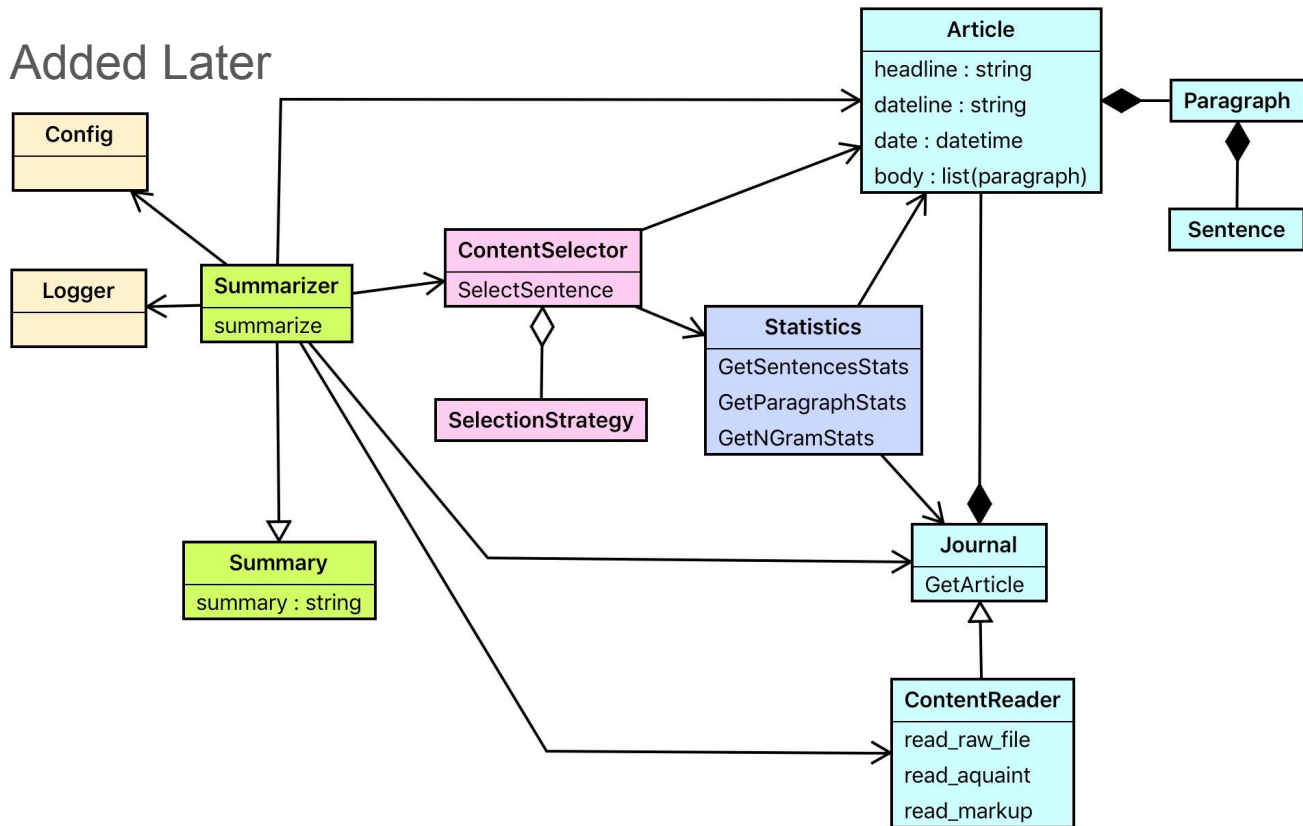
- **Summarization**  
Invokes reading of content and summary sentence selection
- **Content Access**  
Reads and iterates over articles for each document set
- **Sentence Selection**  
Extracts/Creates and Orders summary sentences based on selected articles.
- **Statistics**  
Provides details for sentence selection and for logging.



# System Architecture

## Important Components Added Later

- Configuration
- Logging



# Content Selection

## First Sentence Summarization

There are broadly two types of extractive summarization tasks depending on what the summarization program focuses on. The first is *generic summarization*, which focuses on obtaining a generic summary or abstract of the collection (whether documents, or sets of images, or videos, news stories etc.). The second is *query relevant summarization*, sometimes called *query-based summarization*, which summarizes objects specific to a query. Summarization systems are able to create both query relevant text summaries and generic machine-generated summaries depending on what the user needs.

An example of a summarization problem is document summarization, which attempts to automatically produce an abstract from a given document. Sometimes one might be interested in generating a summary from a single source document, while others can use multiple source documents (for example, a cluster of articles on the same topic). This problem is called multi-document summarization. A related application is summarizing news articles. Imagine a system, which automatically pulls together news articles on a given topic (from the web), and concisely represents the latest news as a summary.

Image collection summarization is another application example of automatic summarization. It consists in selecting a representative set of images from a larger set of images.<sup>[1]</sup> A summary in this context is useful to show the most representative images of results in an image collection exploration system. Video summarization is a related domain, where the system automatically creates a trailer of a long video. This also has applications in consumer or personal videos, where one might want to skip the boring or repetitive actions. Similarly, in surveillance videos, one would want to extract important and suspicious activity, while ignoring all the boring and redundant frames captured.

## Pivoted QR Matrix Decomposition

There are broadly two types of extractive summarization tasks depending on what the summarization program focuses on. The first is *generic summarization*, which focuses on obtaining a generic summary or abstract of the collection (whether documents, or sets of images, or videos, news stories etc.). The second is *query relevant summarization*, sometimes called *query-based summarization*, which summarizes objects specific to a query. Summarization systems are able to create both query relevant text summaries and generic machine-generated summaries depending on what the user needs.

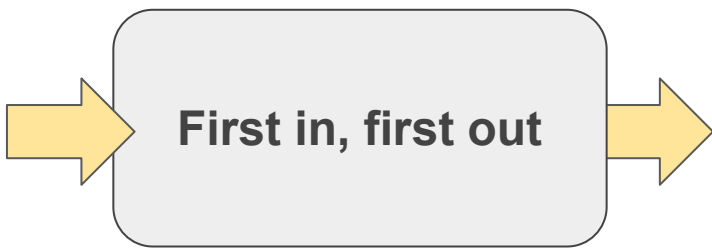
An example of a summarization problem is document summarization, which attempts to automatically produce an abstract from a given document. Sometimes one might be interested in generating a summary from a single source document, while others can use multiple source documents (for example, a cluster of articles on the same topic). This problem is called multi-document summarization. A related application is summarizing news articles. Imagine a system, which automatically pulls together news articles on a given topic (from the web), and concisely represents the latest news as a summary.

Image collection summarization is another application example of automatic summarization. It consists in selecting a representative set of images from a larger set of images.<sup>[1]</sup> A summary in this context is useful to show the most representative images of results in an image collection exploration system. Video summarization is a related domain, where the system automatically creates a trailer of a long video. This also has applications in consumer or personal videos, where one might want to skip the boring or repetitive actions. Similarly, in surveillance videos, one would want to extract important and suspicious activity, while ignoring all the boring and redundant frames captured.



# Information Ordering

## First Sentence Summarization



## Pivoted QR Matrix Decomposition



There are broadly two types of extractive summarization tasks depending on what the summarization program focuses on. The first is generic summarization, which focuses on obtaining a generic summary or abstract of the collection (whether documents, or sets of images, or videos, news stories etc.). The second is query relevant summarization, sometimes called query-based summarization, which summarizes objects specific to a query. **Summarization systems are able to create both query relevant text summaries and generic machine-generated summaries depending on what the user needs.**

An example of a summarization problem is document summarization, which attempts to automatically produce an abstract from a given document. Sometimes one might be interested in generating a summary from a single source document, while others can use multiple source documents (for example, a cluster of articles on the same topic). This problem is called multi-document summarization. A related application is summarizing news articles. Imagine a system, which automatically pulls together news articles on a given topic (from the web), and concisely represents the latest news as a summary.

Image collection summarization is another application example of automatic summarization. It consists in selecting a representative set of images from a larger set of images.<sup>[10]</sup> A summary in this context is useful to show the most representative images of results in an image collection exploration system. Video summarization is a related domain, where the system automatically creates a trailer of a long video. This also has applications in consumer or personal videos, where one might want to skip the boring or repetitive actions. Similarly, in surveillance videos, one would want to extract important and suspicious activity, while ignoring all the boring and redundant frames captured.

There are broadly two types of extractive summarization tasks depending on what the summarization program focuses on. The first is generic summarization, which focuses on obtaining a generic summary or abstract of the collection (whether documents, or sets of images, or videos, news stories etc.). The second is query relevant summarization, sometimes called query-based summarization, which summarizes objects specific to a query. Summarization systems are able to create both query relevant text summaries and generic machine-generated summaries depending on what the user needs.

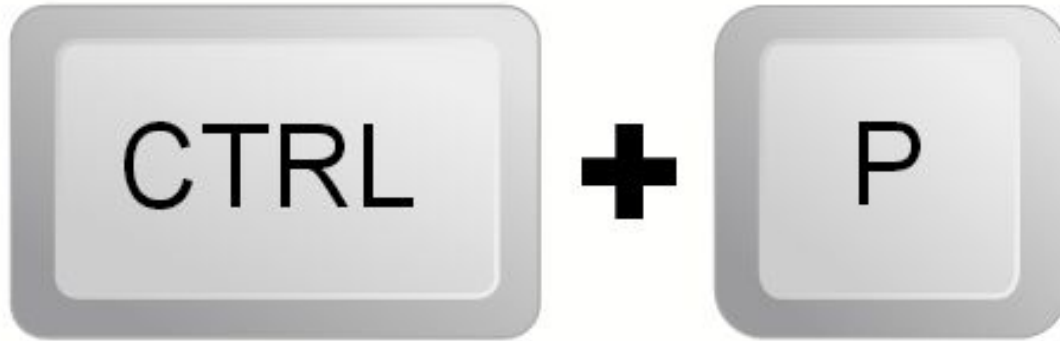
An example of a summarization problem is document summarization, which attempts to automatically produce an abstract from a given document. Sometimes one might be interested in generating a summary from a single source document, while others can use multiple source documents (for example, a cluster of articles on the same topic). This problem is called multi-document summarization. A related application is summarizing news articles. Imagine a system, which automatically pulls together news articles on a given topic (from the web), and concisely represents the latest news as a summary.

Image collection summarization is another application example of automatic summarization. It consists in selecting a representative set of images from a larger set of images.<sup>[10]</sup> **A summary in this context is useful to show the most representative images of results in an image collection exploration system. Video summarization is a related domain, where the system automatically creates a trailer of a long video.** This also has applications in consumer or personal videos, where one might want to skip the boring or repetitive actions. Similarly, in surveillance videos, one would want to extract important and suspicious activity, while ignoring all the boring and redundant frames captured.



# Content Realization

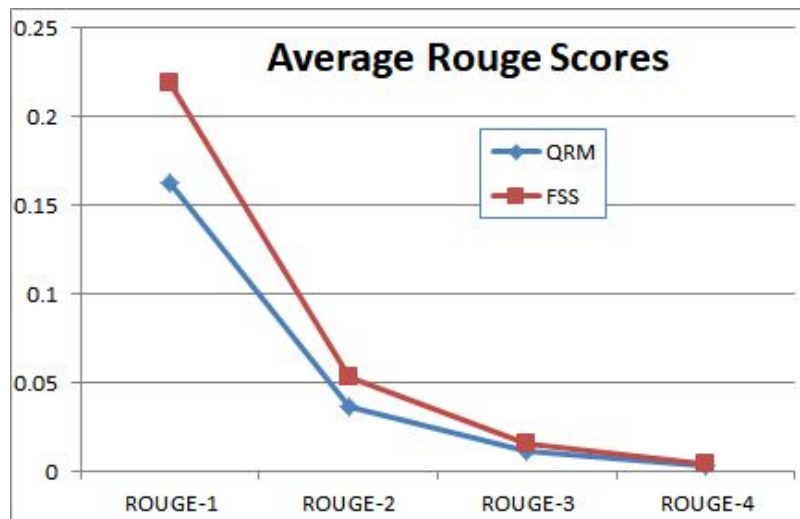
Sentences are written as-is to file with no modification.





# Issues and Successes

- **Know Your Data!**  
AQUAINT/AQUAINT2 formatting issues up to the last minute
- **Unit Tests and Functional Tests On Each Class**
- **Data Results**



# Related Reading

Left Brain Right Brain Multi-Document Summarization

John M. Conroy, Judith D. Schlesinger [2004]

Text Summarization via Hidden Markov Models and Pivoted QR Matrix Decomposition

John Conroy and Dian P. O'Leary [2001]

Beautiful Soup Documentation <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>