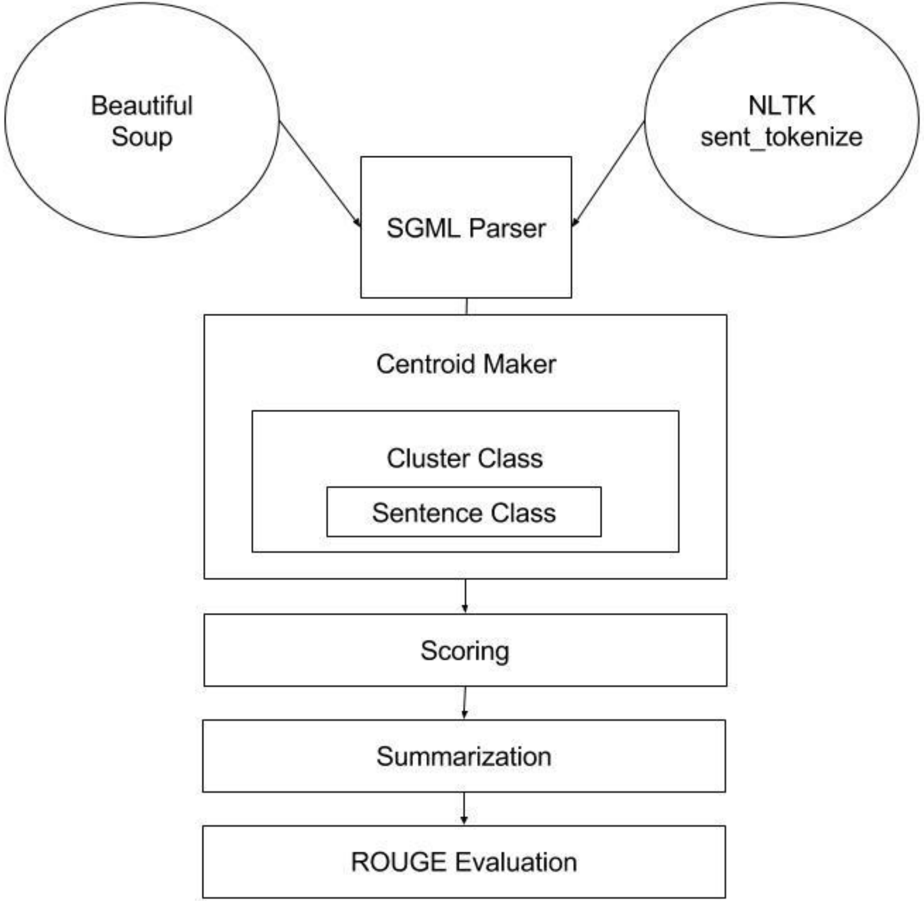
West Coast Python: Deliverable #2

Travis Nguyen, Karen Kincy, Tracy Rohlin


Beautiful Soup
NLTK sent_tokenize
SGML Parser
Centroid Maker
Cluster Class
Sentence Class
Scoring
Summarization
ROUGE Evaluation

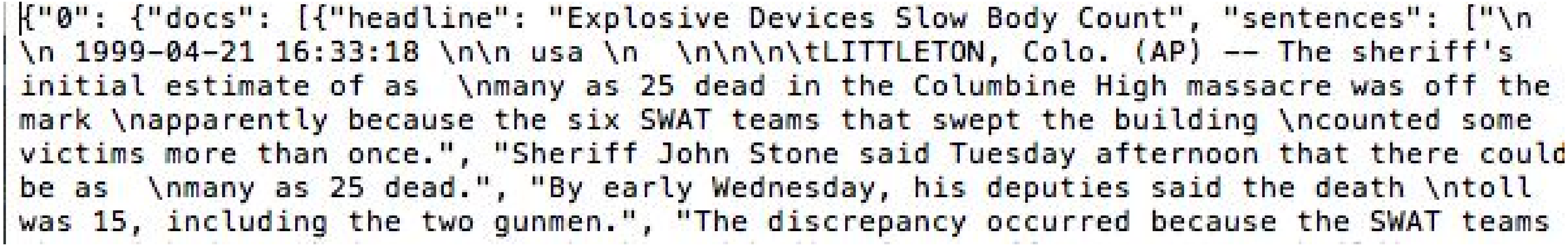
Approach

- Modification of Radev et. al’s centroid-based MEAD system
    - Standard Generalized Markup Language (SGML) Parser
    - Centroid-Based Algorithm
    - Knapsack Algorithm

SGML Parser

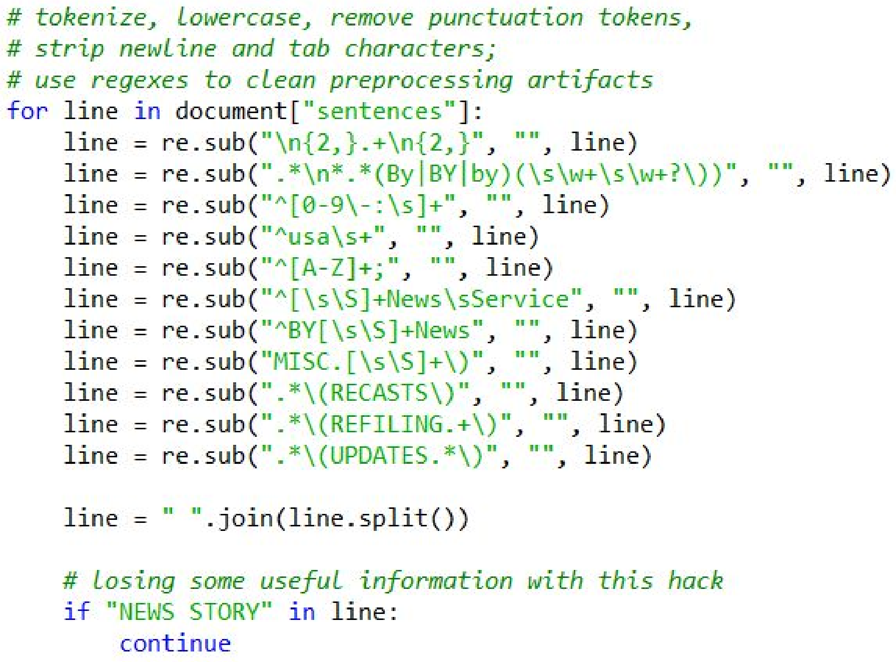
- Used subprocess library to GREP for each document ID in guided summary file
  - Open and parse only unvisited documents
    - Over 3,000 documents in the AQUAINT corpora folders
    - Only 186 ever used (~600 documents used)
    - Too costly to load all files into memory and parse them
- Parsed document using BeautifulSoup
- Removed duplicate headlines as well as extra tags and doc-ids from text
- Tokenized text using NLTK’s sentence tokenizer
- Saved tokenized text into a JSON object containing clusters

Example



```
{"0": {"docs": [{"headline": "Explosive Devices Slow Body Count", "sentences": ["\n
\n 1999-04-21 16:33:18 \n\n usa \n  \n\n\n\tLITTLETON, Colo. (AP) -- The sheriff's
initial estimate of as  \nmany as 25 dead in the Columbine High massacre was off the
mark \napparently because the six SWAT teams that swept the building \ncounted some
victims more than once.", "Sheriff John Stone said Tuesday afternoon that there could
be as  \nmany as 25 dead.", "By early Wednesday, his deputies said the death \ntoll
was 15, including the two gunmen.", "The discrepancy occurred because the SWAT teams
```

Regexes to Clean Text



```
# tokenize, lowercase, remove punctuation tokens,
# strip newline and tab characters;
# use regexes to clean preprocessing artifacts
for line in document["sentences"]:
    line = re.sub("\n{2,}.+\n{2,}", "", line)
    line = re.sub(".*\n*.*(By|BY|by)(\s\w+\s\w+?\))", "", line)
    line = re.sub("^[0-9\-:\s]+", "", line)
    line = re.sub("^usa\s+", "", line)
    line = re.sub("^[A-Z]+;", "", line)
    line = re.sub("^[\s\S]+News\sService", "", line)
    line = re.sub("^BY[\s\S]+News", "", line)
    line = re.sub("MISC.[\s\S]+\)", "", line)
    line = re.sub(".*\(RECASTS\)", "", line)
    line = re.sub(".*\(REFILING.+\)", "", line)
    line = re.sub(".*\(UPDATES.*\)", "", line)

    line = " ".join(line.split())

    # losing some useful information with this hack
    if "NEWS STORY" in line:
        continue
```

Centroid-Based Algorithm

- Based on MEAD system [Radev et al. (2004)]
  - No actual clustering done, unlike MEAD
  - Otherwise might have more than one summary per topic ID
- Cluster corresponds to document set with shared topic ID
- For each cluster, build a centroid of most relevant words
  - Relevancy determined by TF*IDF

TF*IDF

- Term Frequency * Inverse Document Frequency
- TF calculated as average term frequency across documents in cluster
- IDF calculated from background corpus
  - Used “news” subset of Brown corpus from NLTK
  - Will try more background corpora

Centroid #1 [Topic: Giant Panda]

```
giant    19.67778609637496
china    5.37115469050303
provincial    5.2978654874855655
pandas   5.2
province 3.4094961844768505
nature   3.308903072753306
protection    3.222692814300182
reserves 3.091042453358316
forestry 3.027351707134609
panda    3.0
protect  2.47283396268665З
wildlife 2.472833962686653
arrow     2.2705137803509565
forest   2.205935381835537
reserve  2.174751721484161
natural  2.174751721484161
spotted  1.918316218238696б
survey   1.8546254720149895
animal   1.8546254720149895
southwestern 1.8546254720149895
```

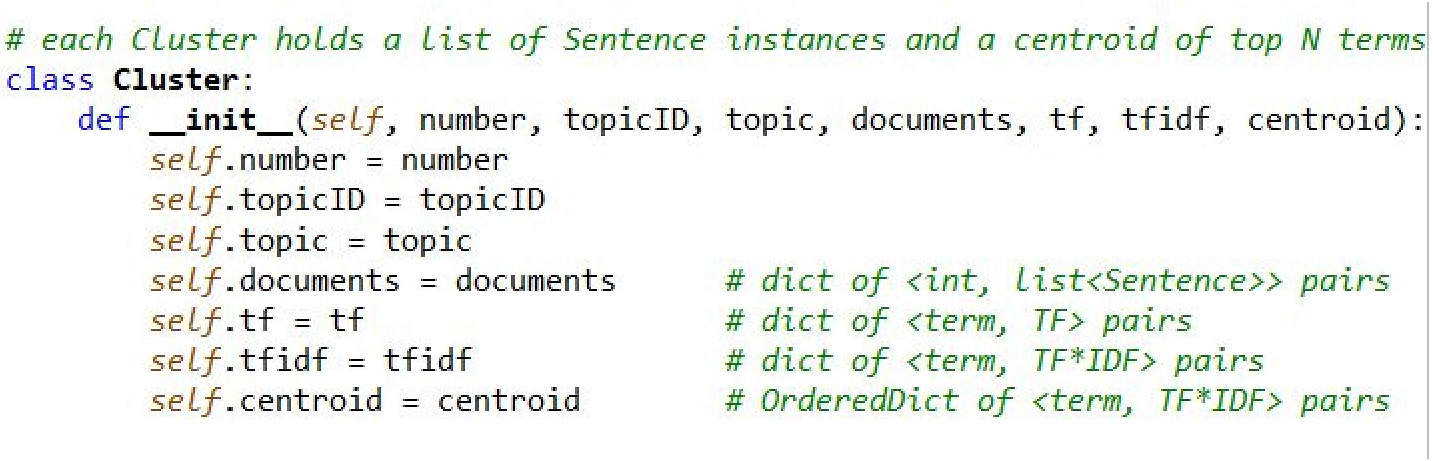
Centroid #2 [Topic: Mt St Helens]

```
steam    7.912396507283637
eruption 7.454545454545454
mount    6.591871665614009
volcano  4.7272727272727275
activity 4.150437715386597
imminent 3.784189633918261
rock     3.6621509253411153
mountain 3.5454545454545454
scientists   3.0
helens   2.909090909090909
ash      2.8181818181818183
survey   2.8100385939621053
erupted  2.8100385939621053
miles    2.6601606031783076
1980     2.6363636363636362
explosion    2.535820209605717
57       2.529034734565895
cloud    2.4081206761298026
underground  2.2480308751696847
alert    2.0641034366826876
```

Centroid #3 [Topic: Ephedra]

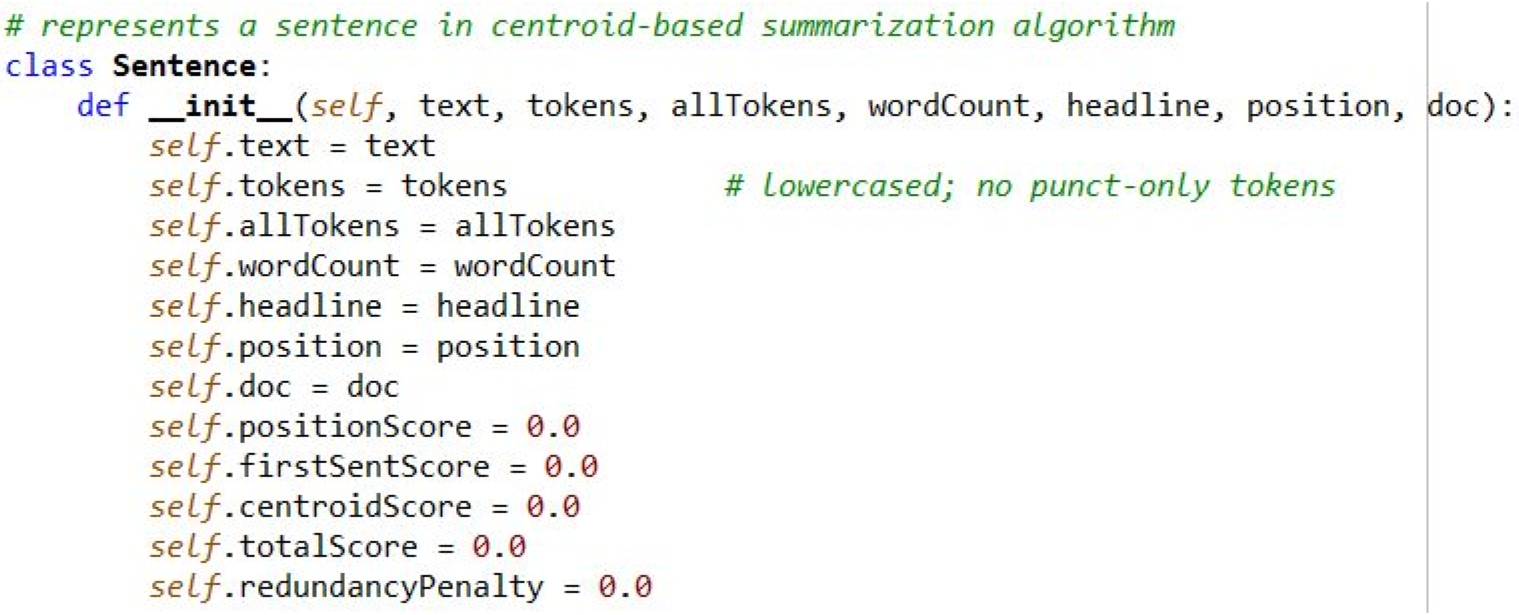
```
consumers     8.600430986177866    dangerous     3.3720463127545264
products      8.00084357418434     fda           3.2727272727273
supplement    7.868108063093895    heart         3.163275231249689
herbal        5.818181818181818    over-the-counter 3.096155155024032
supplements   5.636363636363637    chemicals     3.096155155024032
effects       5.620077187924211
herb          5.160258591706719
patients      5.160258591706719
stimulant     4.816241352259605
safe          4.816241352259605
drug          4.545454545454546
product       4.175430978968244
label         4.128206873365375
containing    3.5586846351559
physicians    3.4401723944711464
```

Cluster Class



```
# each Cluster holds a list of Sentence instances and a centroid of top N terms
class Cluster:
    def __init__(self, number, topicID, topic, documents, tf, tfidf, centroid):
        self.number = number
        self.topicID = topicID
        self.topic = topic
        self.documents = documents      # dict of <int, list<Sentence>> pairs
        self.tf = tf                    # dict of <term, TF> pairs
        self.tfidf = tfidf              # dict of <term, TF*IDF> pairs
        self.centroid = centroid        # OrderedDict of <term, TF*IDF> pairs
```

Sentence Class



```python
# represents a sentence in centroid-based summarization algorithm
class Sentence:
    def __init__(self, text, tokens, allTokens, wordCount, headline, position, doc):
        self.text = text
        self.tokens = tokens            # lowercased; no punct-only tokens
        self.allTokens = allTokens
        self.wordCount = wordCount
        self.headline = headline
        self.position = position
        self.doc = doc
        self.positionScore = 0.0
        self.firstSentScore = 0.0
        self.centroidScore = 0.0
        self.totalScore = 0.0
        self.redundancyPenalty = 0.0
```

Scoring

- Every sentence in a cluster has:
    1. Centroid score
    2. Position score
    3. First sentence overlap score

1. Centroid Score

- Centroid score for each sentence
  - Sum of centroid scores for each word in sentence
- Ignores words outside of centroid
  - Centroid size can be tweaked as parameter
  - Not TF*IDF threshold, unlike MEAD
  - Favors sentences with content relevant to topic of cluster

2. Position Score

- First sentence in every document given position score equal to highest centroid score in this document
    - Favors first sentences
- Position score formula :
    - *C(max)* is highest centroid score of all sentences in this document
    - *n* = number of sentences in this document
    - *i* = index of this sentence

$$P_i = ((n - i + 1)/n) * C_(max)$$

3. First Sentence Overlap Score

- Dot product of first sentence and current sentence
  - Don’t actually vectorize sentences
  - Store <word, count> pairs in dictionary for each sentence
  - Find intersection of keys
  - Calculate dot product
- Like position score, also favors first sentences
  - Need to implement weights for the three scores!

Redundancy Penalty

- Used redundancy penalty in Radev et al. (2004)
  - Strongly favors first sentences
    - Some summaries consist only of first sentences
    - Repetitive
- Used modified version of redundancy penalty
  - Calculate overlap of current sentence with that of every sentence with a higher total score
  - Performs better than original version
    - More concise
    - Less repetitive

Summarization

- Order sentences in descending order by total score
- Knapsack algorithm
    - Return sublist of sentences
        - Highest total score (summation of total scores for each sentence in sublist)
        - Word length equal to or less than a threshold
- Output summaries as model evaluation files with unique document IDs

Runtime

- About one hour for SGML parser
  - Caching the text much faster overall
  - JSON file loads in seconds
- Average of 38.222 seconds for main module
  - On Karen’s old local machine
  - Extensively uses hash tables
  - Only needs to calculate IDF scores once

Evaluation

- ROUGE evaluation toolkit
- N-gram size: 1-4

Good [Results for Topic: Debra La Fave]

A hitch has developed in a plea deal agreed to by a former teacher who pleaded guilty to having sex with a 14-year-old student, a spokesman for the prosecutor said. A female teacher pleaded guilty Tuesday to having sex with a 14-year-old student, avoiding prison as part of a plea agreement. Prosecutors released Thursday photographs and secretly recorded tapes to support their accusation that Debra LaFave, formerly a schoolteacher, had sex with a 14-year-old student. She pleaded guilty to two counts of lewd and lascivious battery. The agreement was meant to resolve charges against Debra Lafave, 25, in two counties.

Topic ID	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
D1021D	37.5%	10.909%	1.852%	0.0%

Bad [Results for Topic: Threat To Frogs]

Gerardo de la Cruz, a 42-year-old farmer, is one of thousands who spent two weeks blockading a U.S.-run gold mine in the Andes, the latest example of the uneasy relationship between Peru's rural poor and the mining industry. Almost 150 species of amphibians have apparently gone extinct and at least one-third of the rest are facing imminent threats that could soon wipe them out, according to a worldwide assessment by scientists published Thursday. Amphibians are experiencing a precipitous decline across the globe, according to the first comprehensive world survey of the creatures, which include frogs, toads and salamanders.

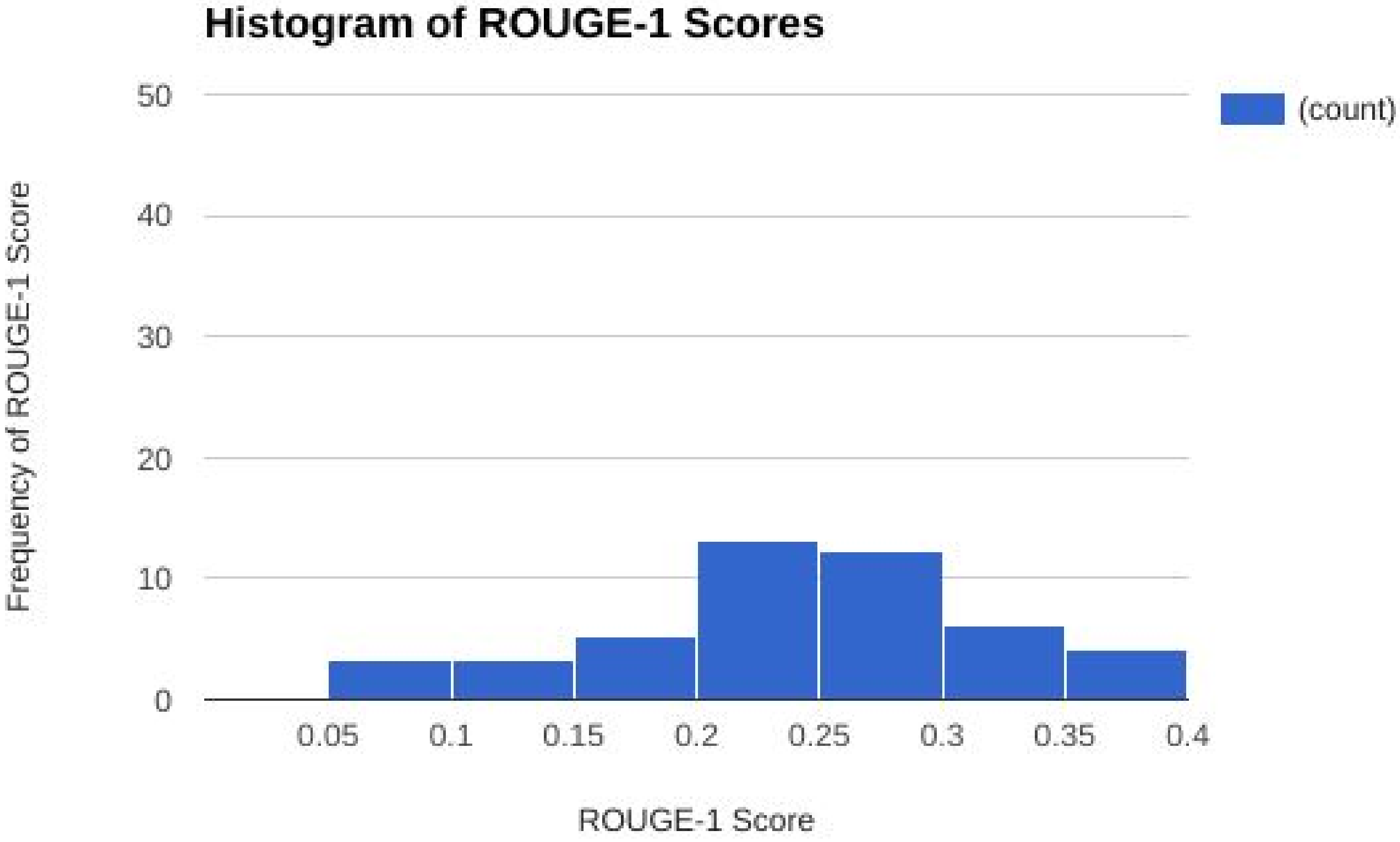
Topic ID	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
D1034F	5.714%	0.0%	0.0%	0.0%

Ugly [Results for Topic: Head Injuries]

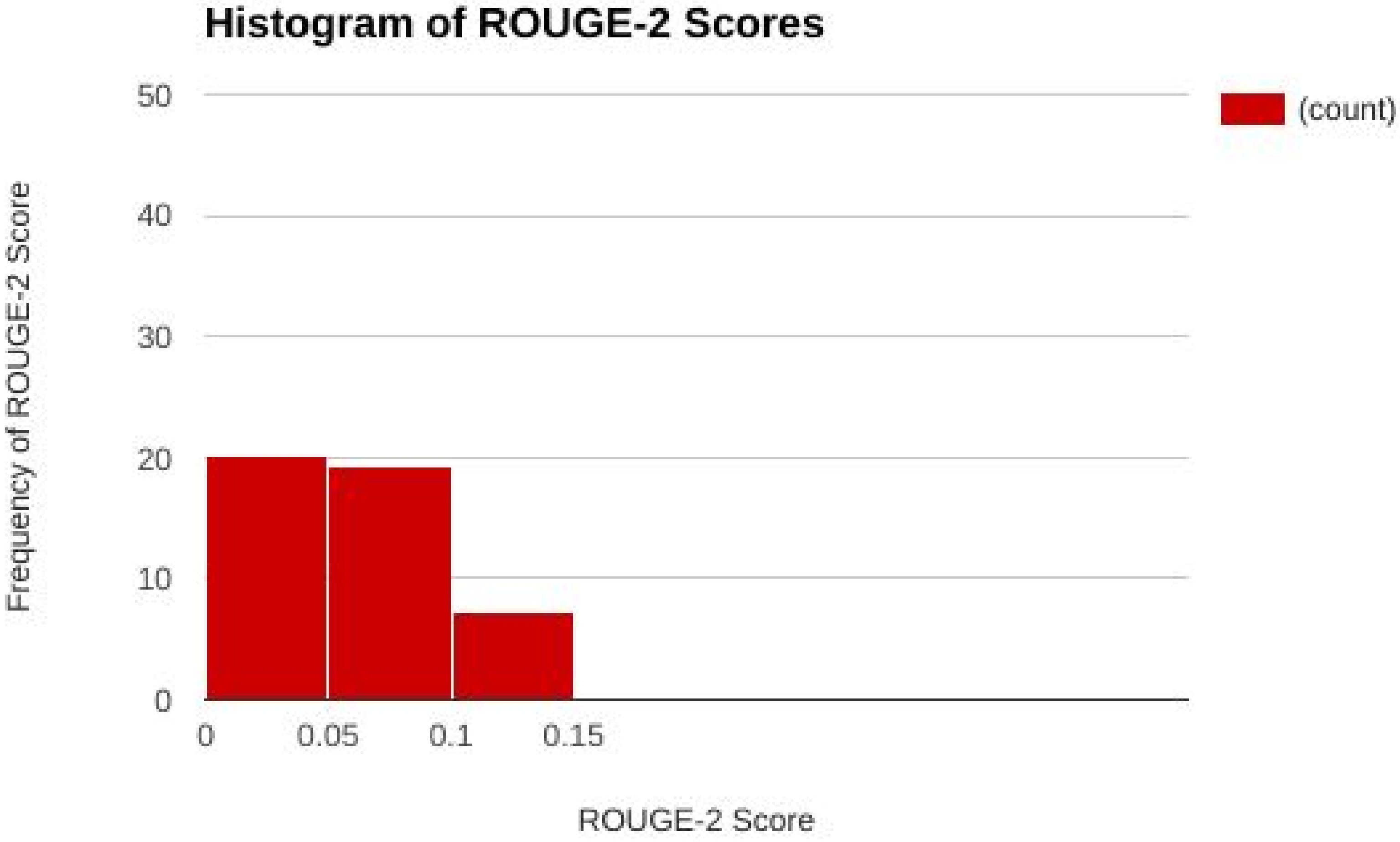
WHO calls on China to lower 680-a-day road accident death tollATTENTION - INSERTS details, ADDS quotes /// ss problems with the way transportation is organized, factors contributing to accidents, the need to create better safety devices for vehicles and passengers and to build a better mechanism to respond to accidents, Kurg said. ASEAN Transport Ministers issued a ministerial declaration on ASEAN road safety Tuesday to enhance the road safety and reduce the traffic casualties in member countries. Moreover, the ministers agreed to foster the development of a new culture of road safety among citizens especially the young, school-age or child population.

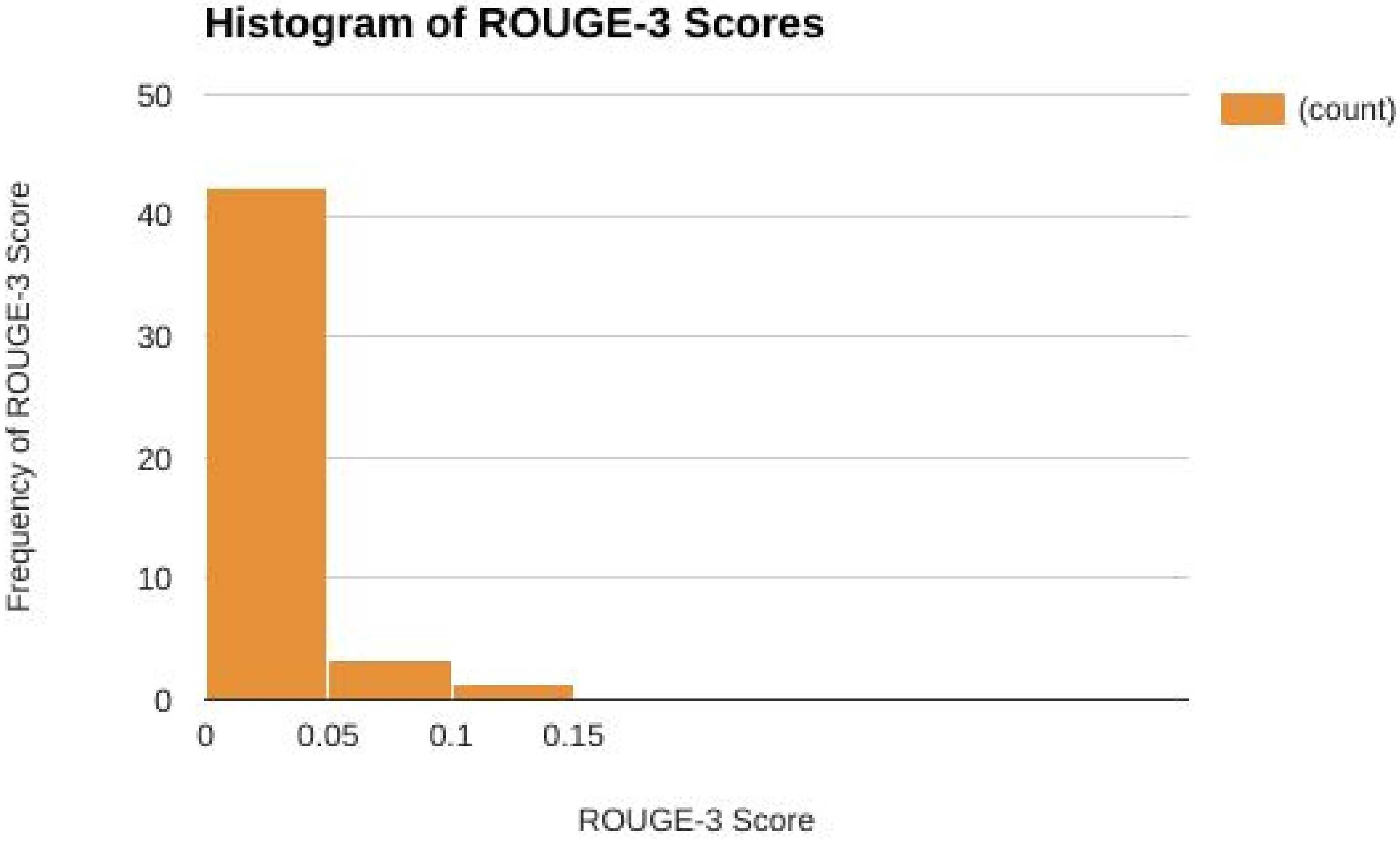
Topic ID	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
D1026E	7.273%	0.0%	0.0%	0.0%

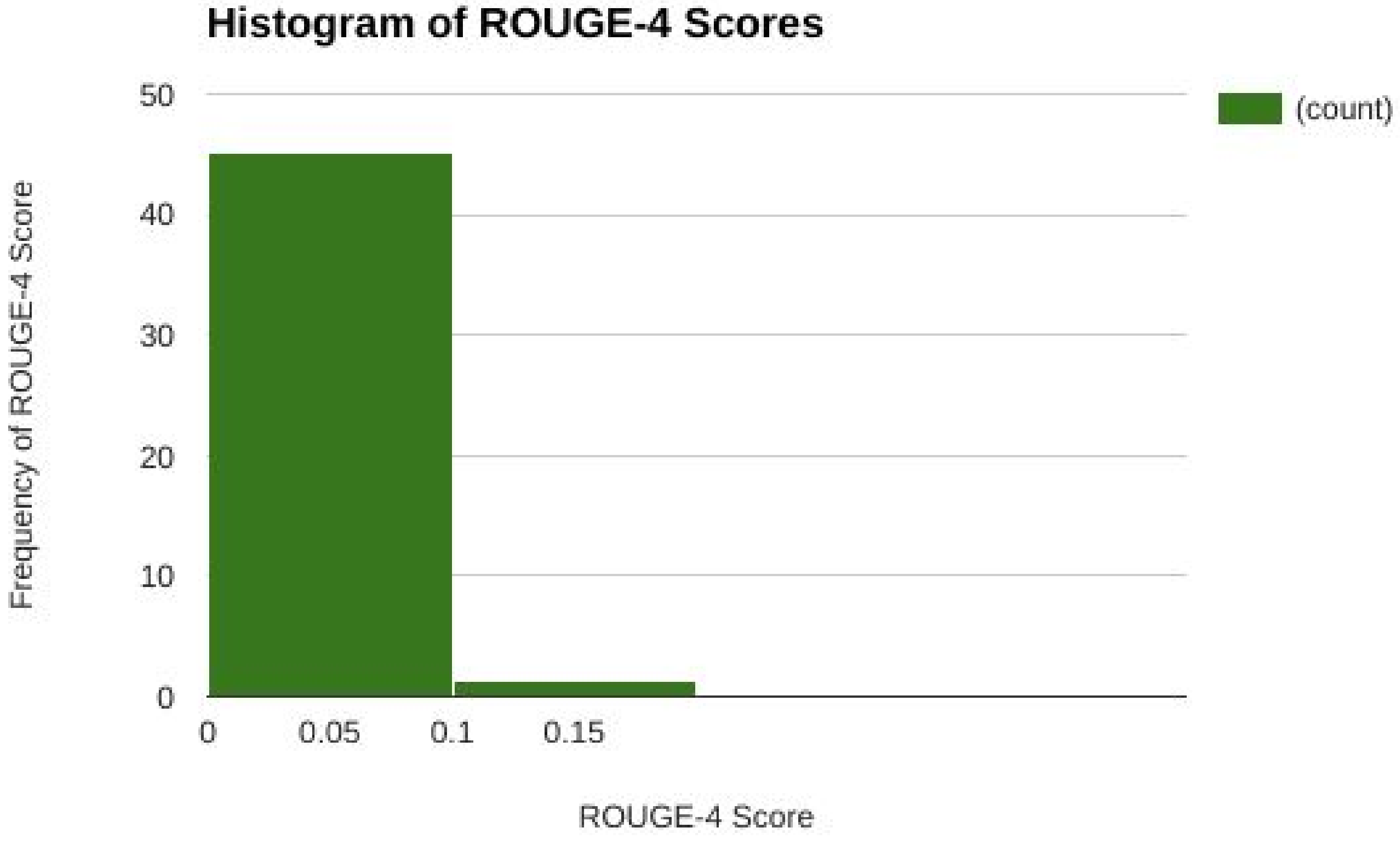
Comparing ROUGE Scores

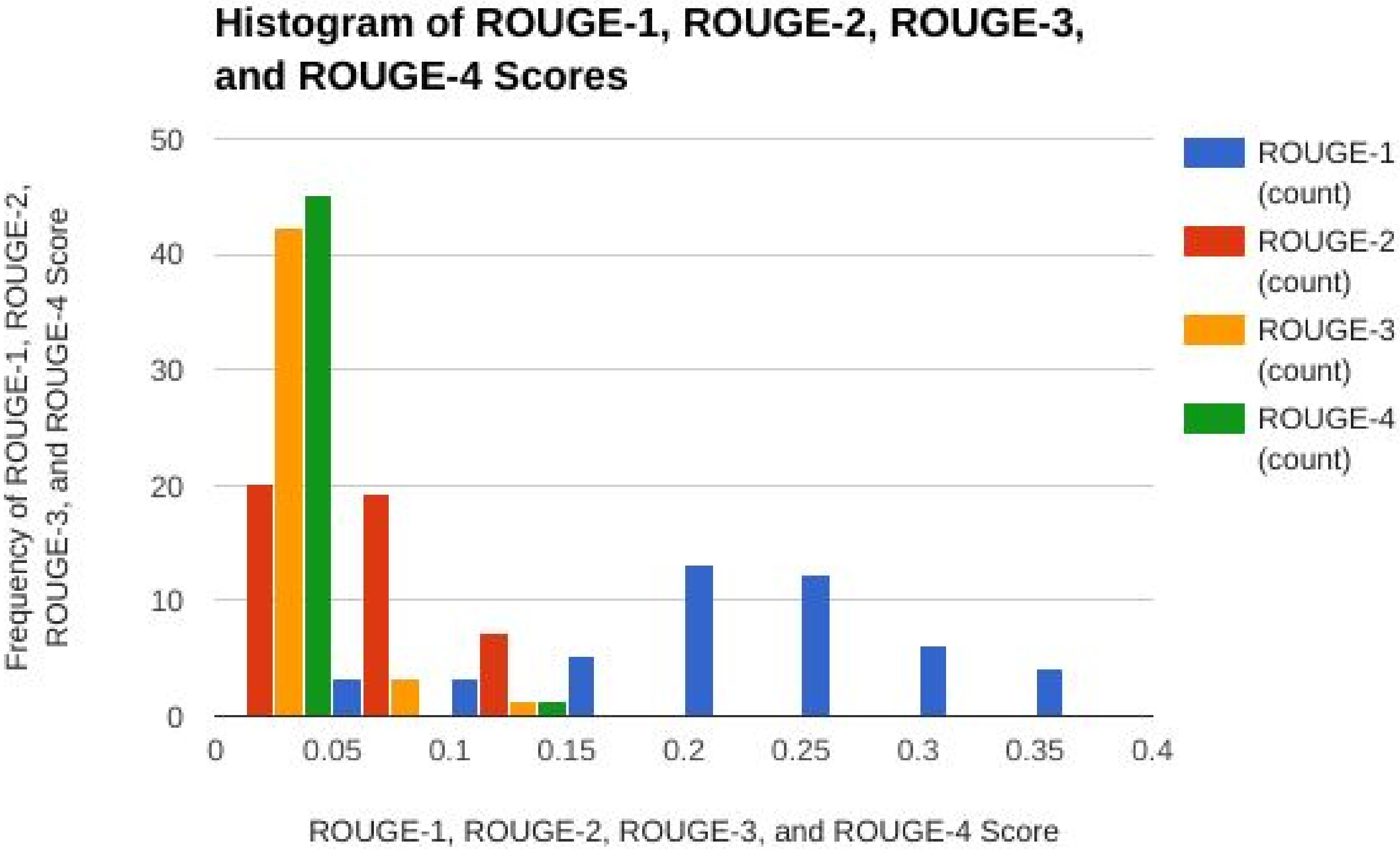


Histogram of ROUGE-1 Scores
50
40
30
20
10
0
Frequency of ROUGE-1 Score
🟦 (count)
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
ROUGE-1 Score


Histogram of ROUGE-2 Scores
🟥 (count)
Frequency of ROUGE-2 Score
50
40
30
20
10
0
0
0.05
0.1
0.15
ROUGE-2 Score


Histogram of ROUGE-3 Scores
(count)
Frequency of ROUGE-3 Score
50
40
30
20
10
0
0
0.05
0.1
0.15
ROUGE-3 Score


Histogram of ROUGE-4 Scores
🟩 (count)
Frequency of ROUGE-4 Score
50
40
30
20
10
0
0
0.05
0.1
0.15
ROUGE-4 Score


Histogram of ROUGE-1, ROUGE-2, ROUGE-3, and ROUGE-4 Scores
Frequency of ROUGE-1, ROUGE-2, ROUGE-3, and ROUGE-4 Score
50
40
30
20
10
0
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
ROUGE-1, ROUGE-2, ROUGE-3, and ROUGE-4 Score
ROUGE-1 (count)
ROUGE-2 (count)
ROUGE-3 (count)
ROUGE-4 (count)

Average and Standard Deviation

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Average	24.28863043%	5.871891304%	1.670565217%	0.5362608696%
Standard Deviation	7.825564137%	3.582682832%	2.329799339%	1.712149597%

Discussion

- ROUGE-1 performs the best
- ROUGE-1 scores typically in range of 20-30%
  - This makes sense...
  - Centroid-based algorithm favors terms with high TF*IDF scores
  - High TF*IDF scores = unigrams relevant to topic cluster
- ROUGE-2 and above scores need improvement
  - ROUGE-4 especially terrible
  - Possibly information ordering issue
  - Content realization might further reduce redundancy

Error Analysis

- Some documents still contain header information
    - Source, date, location, etc.
    - Takes up space in summaries
- Some evals did not have corresponding peer/model summaries
    - Bug in creating the model summaries?

Future Improvements

- Add corpus selection capability
  - Reuters, New York Times, etc.
- Order input documents chronologically
  - Mentioned in MEAD paper [Radev et al. (2004)]
- Tweak redundancy penalty to further reduce repetition in summaries
- Add weights to optimize scores for sentences
  - Implement machine learning to find best weights
- Try lemmatization and stop-wording
  - Might improve centroids even more
- Fix the regexes
  - Text still noisy!

Bibliography (1 of 2)

Dragomir R. Radev, Hongyan Jing, & Malgorzata Budzikowska. 2000. *Centroid-based Summarization of Multiple Documents: Sentence Extraction, Utility-based Evaluation, and User Studies*. NAACL-ANLP-AutoSum '00 Proceedings of the 2000 NAACL-ANLP Workshop on Automatic Summarization. Pages 21-30.

Dragomir R. Radev, Hongyan Jing, Malgorzata Stys, & Daniel Tam. 2004. *Centroid-based Summarization of Multiple Documents*. Information Processing and Management 40. Pages 919-938.

Bibliography (2 of 2)

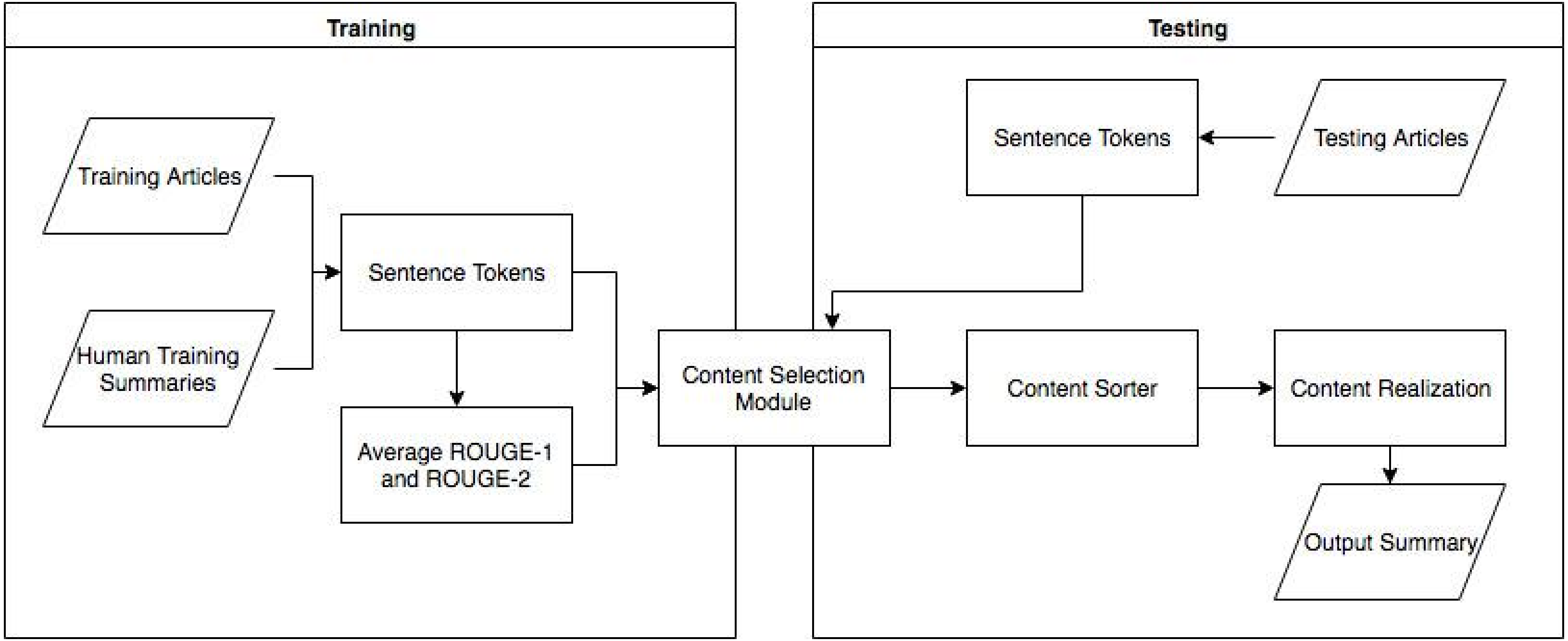
Kai Hong, Mitchell Marcus, & Ani Nenkova. 2015. *System Combination for Multi-document Summarization*. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal. Pages 107-117.

Natalia Vanetik, & Marina Litvak. 2015. *Multilingual Summarization with Polytope Model*. Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue. Pages 227-231.

573 Project Report

Mackie Blackburn, Xi Chen, Yuan Zhang

System Overview



Data

Training Data:

DUC 2009: 44 topic clusters x 10 docs + 44 x 8 abstractive summaries

SummBank 1.0: 40 topic clusters x 10 docs + 40 x 3 groups (50, 100, 200) x 3 abstractive summaries

Devtest Data:

DUC 2010: 46 topic clusters x 10 docs + 46 x 8 abstractive summaries

SummBank summaries:

50-word:
"Customs Commissioner, Mr.John Tsang Chun-wah assured that Customs is committed to intellectual property rights (IPR) protection and will continue to maintain Hong Kong as a pirate-free zone.\n\nFor 1997 and 1998 Hong Kong has remained on the \"Watch List\" of the United States Trade Representative's _Special 301 Report_ on the protection of IPR.\n

100-word:
"Customs Commissioner, Mr.John Tsang Chun-wah assured that Customs is fully committed to intellectual property rights (IPR) protection and will continue to maintain Hong Kong as a pirate-free zone. \n\nHong Kong had remained on the \"Watch List\" of the United States Trade Representative's _Special 301 Report_ on the protection of IPR for 1997 and 1998.\n\nApproximately 150 delegates attented the World Intellectual Property Organization Asian Regional Symposium . In

200-word:
"Customs Commissioner, Mr.John Tsang Chun-wah assured that Customs is fully committed to intellectual property rights (IPR) protection and will continue to maintain Hong Kong as a pirate-free zone. There are approximately 300 officers in the Intellectual Property Investigation Bureau and a 185-member Special Task Force that helps fight the street war.\n\nIn December 1997, a license requirement for the import and export of optical disc manufacturing equipment was issued. In spite of this, Hong Kong remained on the \"Watch List\" of the United States

Content Selection

Multi-Layer Perceptron Regression

- Better performance than linear regression

Vectors created from sentences

Initially used binary target

Better performance with averaged ROUGE-1 and ROUGE-2 as target

Content Selection Features

Counts of POS tags

Probability of quotes, commas, numbers, and capital words

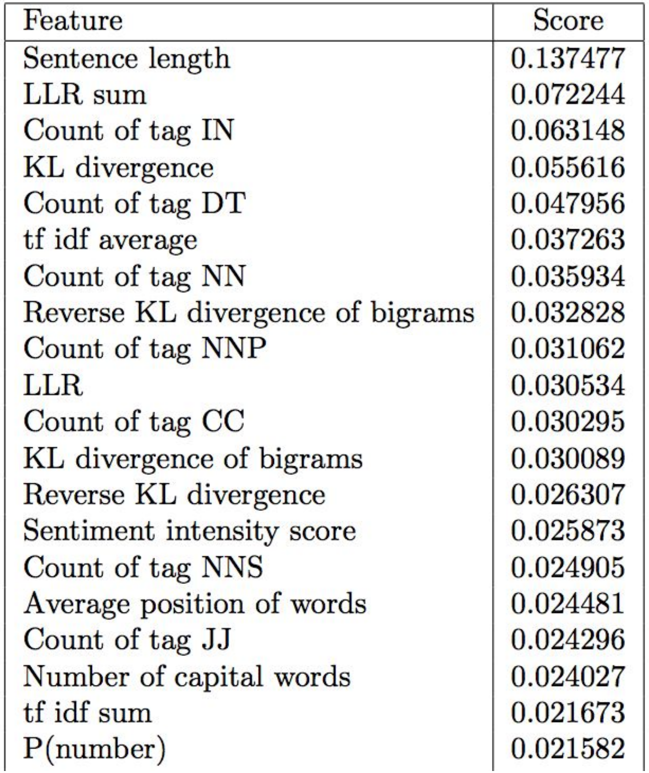
Sentiment analysis

TF*IDF sum and average

LLR

KL Divergence of unigram and bigram probabilities

Averaged positions of words in documents



Feature	Score
Sentence length	0.137477
LLR sum	0.072244
Count of tag IN	0.063148
KL divergence	0.055616
Count of tag DT	0.047956
tf idf average	0.037263
Count of tag NN	0.035934
Reverse KL divergence of bigrams	0.032828
Count of tag NNP	0.031062
LLR	0.030534
Count of tag CC	0.030295
KL divergence of bigrams	0.030089
Reverse KL divergence	0.026307
Sentiment intensity score	0.025873
Count of tag NNS	0.024905
Average position of words	0.024481
Count of tag JJ	0.024296
Number of capital words	0.024027
tf idf sum	0.021673
P(number)	0.021582

Information Ordering and Content Realization

Currently, information is ordered in the output of content selector by sentence scores.

In content realization, a greedy algorithm is applied:

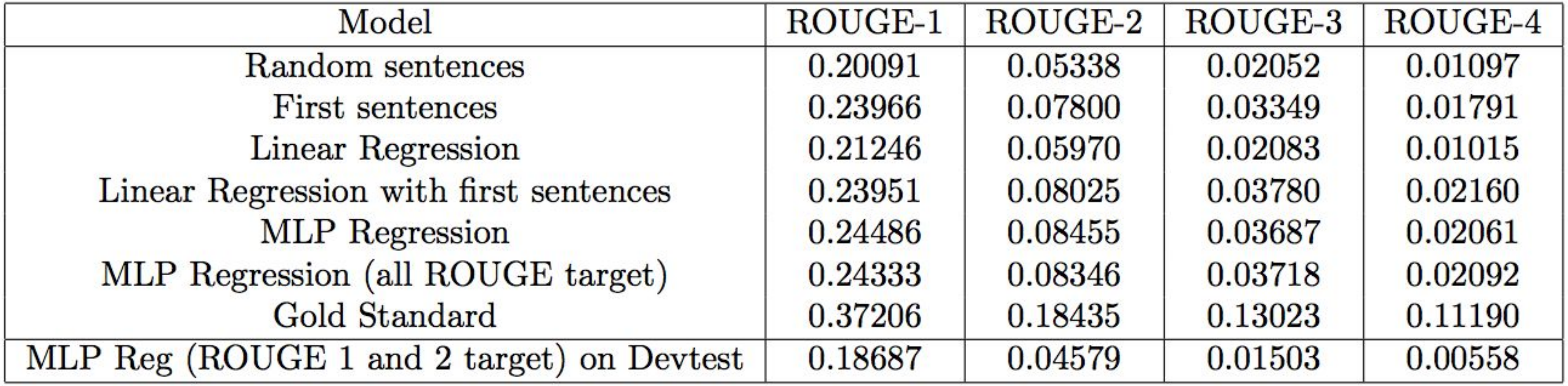
1, while not exceed word limit:

2, pick the sentence with the highest score among candidates

3, unless the sentence's tf-idf similarity with candidates exceed threshold

(t <0.8)

Results



Model	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Random sentences	0.20091	0.05338	0.02052	0.01097
First sentences	0.23966	0.07800	0.03349	0.01791
Linear Regression	0.21246	0.05970	0.02083	0.01015
Linear Regression with first sentences	0.23951	0.08025	0.03780	0.02160
MLP Regression	0.24486	0.08455	0.03687	0.02061
MLP Regression (all ROUGE target)	0.24333	0.08346	0.03718	0.02092
Gold Standard	0.37206	0.18435	0.13023	0.11190
MLP Reg (ROUGE 1 and 2 target) on Devtest	0.18687	0.04579	0.01503	0.00558

Example Summary

After a massive draft environmental analysis the controversial proposal to construct a wind farm in Nantucket Sound is coming down to aesthetics The report on Cape Wind s 130 turbine proposal finds few costs to marine or bird life that might outweigh the project s benefits to both the environment and the economy But the ships involved today are only small fishing boats wanting to protect their livelihood He s been there and done that Now a private company is proposing to build the world s largest offshore wind power plant right in the middle of it But a wind farm with hundreds or even thousands of large turbines removes an enormous amount of energy from the air

Issues and Future Improvements

Explore the possibility of machine learning on information ordering

Sentence segmentation lowers ROUGE scores. Why? Intersentential relationship?

Data: more data is good data? SummBank

Features, more features! LexRank

- Identify good features

Related Readings

Hong, K. and Nenkova, A. (2014) Improving the estimation of word importance for news multi-document summarization, in Proceedings of EACL.

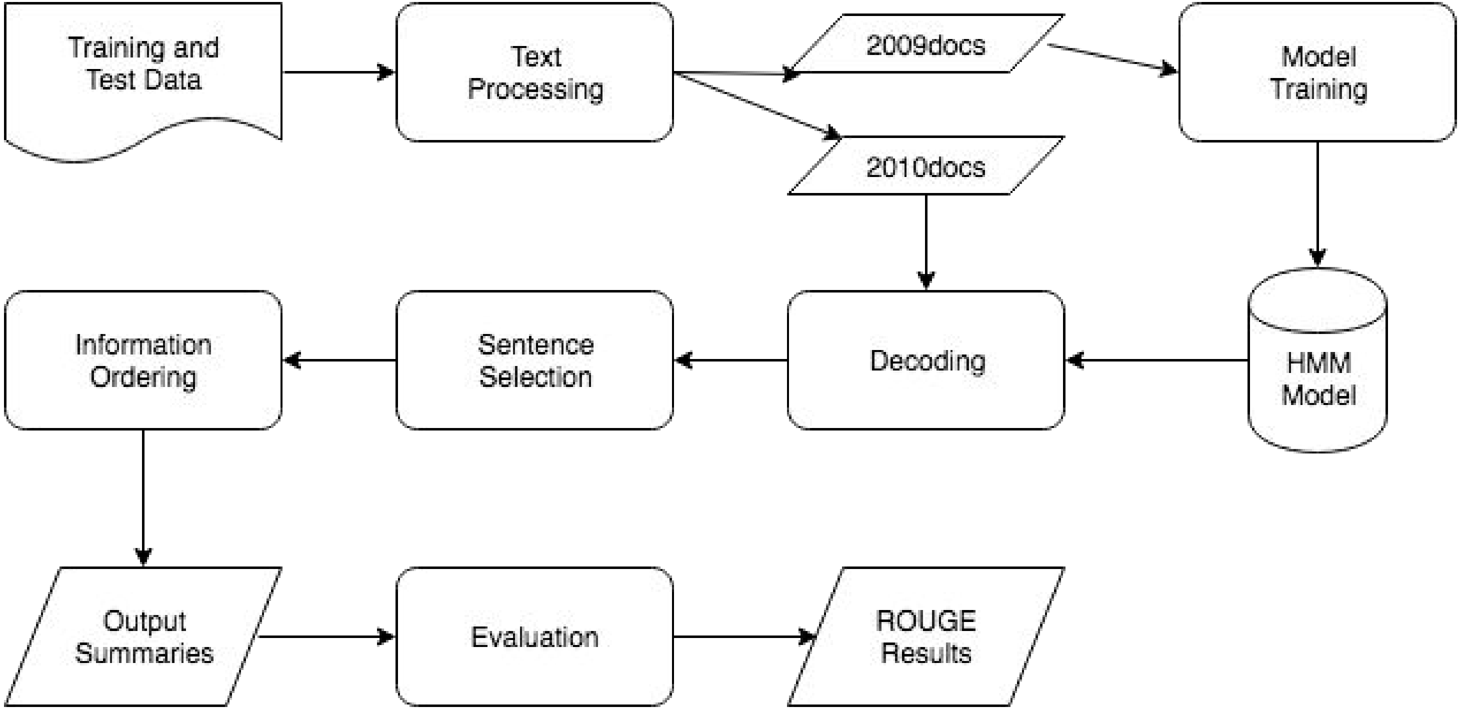
Erkan, G. and Radev, D. (2004).Lexrank: graph-based lexical centrality as salience in text summarization. J. Artificial Intelligence Research, 22(1):457-479.

Cao, Z., Wei, F., Dong, L., Li, S., & Zhou, M. (2015). Ranking with Recursive Neural Networks and Its Application to Multi-Document Summarization. In *AAAI* (pp. 2153-2159).

D2: A System in the Making

Angie McMillan-Major, Alfonso Bonilla, Marina Shah, Lauren Fox

System architecture



Preprocessing

- Processing of XML files
  - Grab topic ID, title, narrative (if there is one), doc set ID, and individual document IDs
  - Print as an array of JSON objects to a file
- Inserting Data into JSON File
  - Extract out headline and text
  - Parsed Using NLTK

```
    {
    "topicID":"",
    "title":"",
    "narrative":"",
    "doc-setID":"",
    "docIDs":[list of doc ids]
    "doc-paths":[list of doc paths]
    "text":[[headline,[list of tokenized
sentences]]]
    "summaries":[list of summaries]
    }
```

Content selection

- Feature Extraction
  - From JSON files, use gold standards to produce I/O tags for the docset text
  - Extract features we think are relevant for each sentence
- Model Building
  - HMM
- Decoding
  - Viterbi

Feature Extraction

- Input: JSON file from the last step
- Output: CSV with I/O tagged data, topicID field, narrative field
  - For each model summary set, take first sentences together and find most similar sentence in docset - repeat for all model sentences
  - We label I/O on the sentence level and will use sub-sentence-level features
- CSV is input to the model-building module, which performs feature extraction
  - Number of keywords: x<=5, 5<x<=10, x>10
  - Contains [NER]: Binary feature for each NER type
  - Sentence length: 0<x<=15, 16<x<=30, 31<x<=45, etc. until x>90
  - Also: Get term frequency counts for LLR weights

Model Building

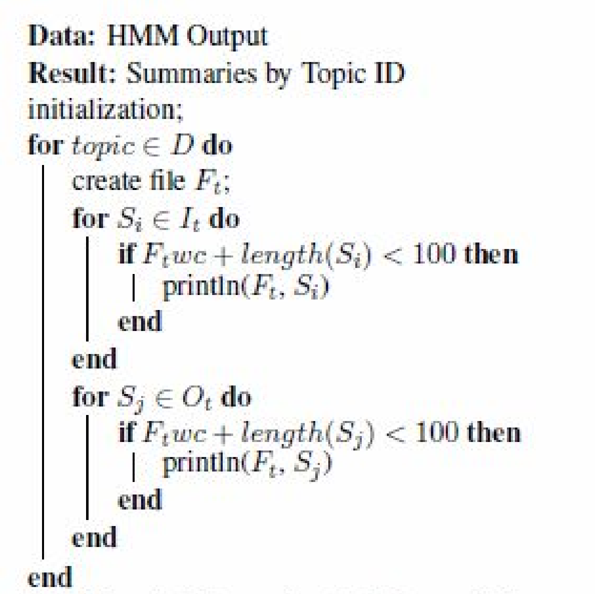
- HMM: Need initial state probabilities, transition probabilities, and emission probabilities
- Initial state probabilities
  - P(I|first_sent_in_docset) and P(O|first_sent_in_docset)
  - Right now, “lazy” method of just taking all sentences in docset together
  - Should separate by article somehow
- Transition probabilities
  - P(I|O), P(I|I), etc. for label sequences
- Emission probabilities
  - P(sentence|O) = P(feature$_1$|O)*P(feature$_2$|O)*…*P(feature$_N$|O)
  - Same for I

Decoding

- Viterbi Algorithm
- Input: Model
    - Initial, transition, and emission probabilities from training
    - Term counts for background corpus for LLR computing
- Calculate P(sentence|label) by treating each sentence’s score as a product of features
- Output: For each docset
    - Docset ID
    - Text with I/O labels and LLR weights for postprocessing
        - E.g. sentence_1/I/0.35 sentence_2/O/0.27 … sentence_N/O/0.11

Information Ordering

- Currently relevance-based ordering
  - For each topic, I-tagged sentences and O-tagged sentences are put into separate priority queues, based on highest LLR
  - Pull from I-tagged sentences until either the word count reaches 100 or I-tagged priority queue is empty
  - Do the same for the O-tagged sentences
- Future improvements:
  - Incorporating concept salience
  - Incorporating temporal structure
  - Checking for redundancy



```
**Data:** HMM Output
**Result:** Summaries by Topic ID
initialization;
for $topic \in D$ do
    create file $F_t$;
    for $S_i \in I_t$ do
        if $F_t wc + length(S_i) < 100$ then
            println($F_t$, $S_i$)
        end
    end
    for $S_j \in O_t$ do
        if $F_t wc + length(S_j) < 100$ then
            println($F_t$, $S_j$)
        end
    end
end
```

Algorithm 1: Information Ordering and Content Realization

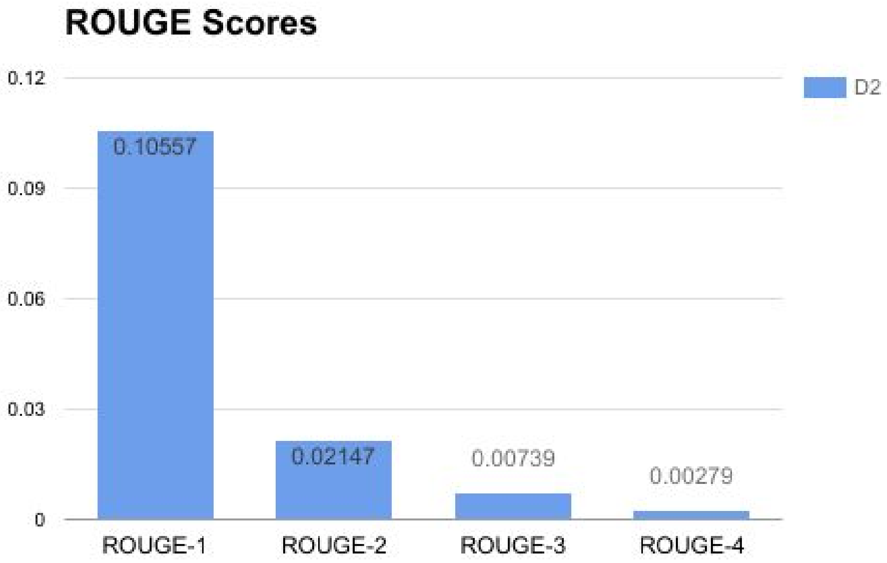
Content Realization

- Sentences are currently printed without changing the string as it appears in the text
- Future improvements to explore:
    - Constraining minimum and maximum sentence length (see example summaries)
    - Running a POS tagger on selected sentences to remove ADJ, ADV, PP, etc.
    - Coreference resolution

Results

ROUGE Evaluation Metric

- Compare automatically generated summary against human-created gold standard summaries
- N-Gram overlap:
  - Uni-, bi-, tri-, and 4-grams
- Reports 3 statistics:
  - Recall
  - Precision
  - F-Measure
- We are interested in **recall** - the fraction of relevant n-grams (n-grams in human summaries) that our system generates



Results: Example Summaries

A good(ish) example…

```
Sara Lee Corp. announced in late December a recall of hot dogs and other packaged meats as a precaution after an outbreak of food poisoning sickened more than 35 people in nine states , killing four .
Last week , Oscar Mayer Foods Corp. recalled more than 28,000 pounds of deli meat because of concerns of possible contamination with the Listeria bacteria .
`` The new information based on this outbreak brings into question the adequacy of control procedures we 've been relying on , '' Billy said .
So , why is listeria suddenly popping up ?
```

A less good example…

```
Pietersz said the trio could not have dropped Holloway off at the Holiday Inn because their vehicle was not captured by hotel surveillance cameras .
( mn-lja )
( pp-lja )
( pp-lja/maf )
( pp-maf )
`` Basically we interrogated him , '' Bearman said . `` He never denied being with her . ''
`` They did not know what was going on , '' she said . `` We were there on a mission . ''
Aruban authorities have said they are pursuing all leads and protecting no one .
```

Issues and Successes

Issues/Future Work:

- Inconsistencies in the Documents
- Gold summaries are Abstractive -> cosine similarity to attempt handling
- HMM heavily favors ‘O’ label -> Modify preprocessing steps
- Holes in data? (Missing text for some docsets and missing pieces of sentences) -> Bug tracking
- Inclusion of word salad sentences that should be ignored in preprocessing
- More complex content ordering and realization

Successes:

- It runs end to end :D
- Some of the summaries seem reasonable

Acknowledgements

We would like to thank Markov, for remaining hidden.

References

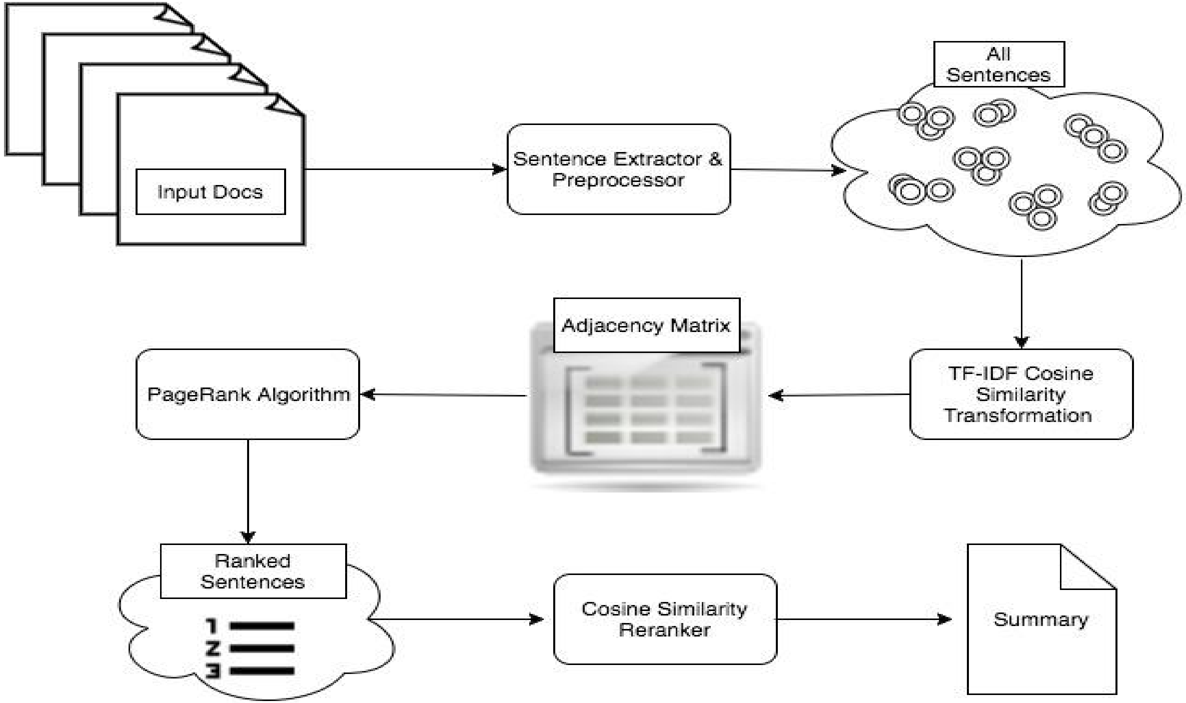
John M. Conroy and Dianne P. O’Leary. 2001. Text summarization via hidden markov models. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, New York, NY, USA, SIGIR ’01, pages 406–407. https://doi.org/10.1145/383952.384042.

John M. Conroy, Judith D. Schlesinger, Jade Goldstein, and Dianne P. O’Leary. 2004. Left-brain/right-brain multi-document summarization. In Proceedings of the Document Understanding Conference (DUC 2004).

Multi-Document Summarization

Eslam Elsawy, Audrey Holmes, Masha Ivenskaya

System Architecture


Input Docs
Sentence Extractor & Preprocessor
All Sentences
TF-IDF Cosine Similarity Transformation
Adjacency Matrix
PageRank Algorithm
Ranked Sentences
1
2
3
Cosine Similarity Reranker
Summary

Dataset Pre-Processing

Issues:

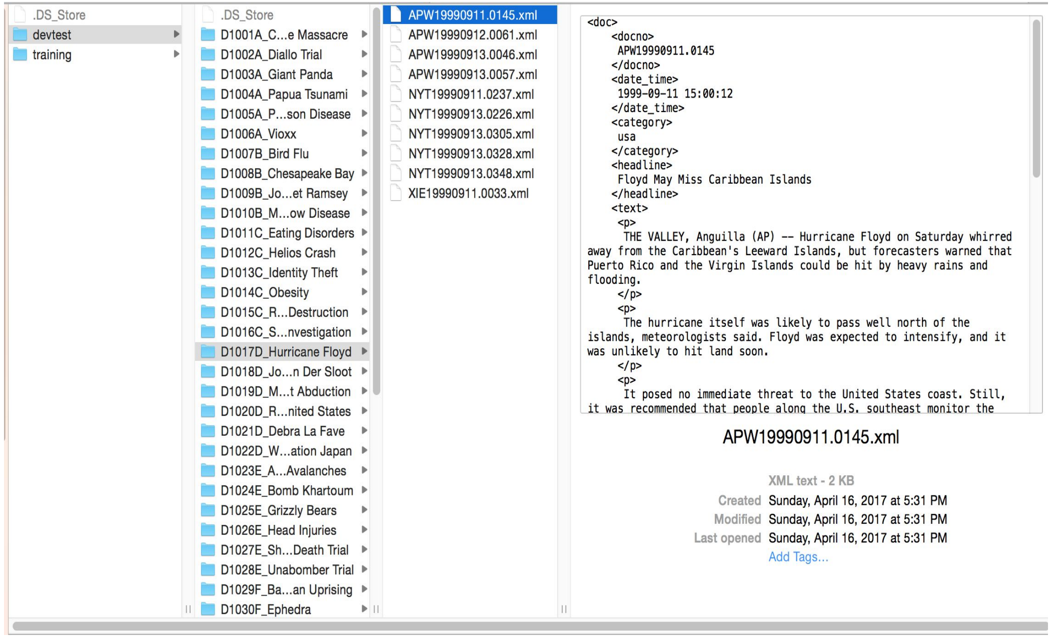
- Non-uniform file naming scheme
- Encoded characters (e.g: &amp)
- Not rooted xml files

What we did:

- Separate cleaning module

Gain:

- Easier to explore the dataset
- Better running times



.DS_Store
devtest
training

.DS_Store
D1001A_C...e Massacre
D1002A_Diallo Trial
D1003A_Giant Panda
D1004A_Papua Tsunami
D1005A_P...son Disease
D1006A_Vioxx
D1007B_Bird Flu
D1008B_Chesapeake Bay
D1009B_Jo...et Ramsey
D1010B_M...ow Disease
D1011C_Eating Disorders
D1012C_Helios Crash
D1013C_Identity Theft
D1014C_Obesity
D1015C_R...Destruction
D1016C_S...nvestigation
D1017D_Hurricane Floyd
D1018D_Jo...n Der Sloot
D1019D_M...t Abduction
D1020D_R...nited States
D1021D_Debra La Fave
D1022D_W...ation Japan
D1023E_A...Avalanches
D1024E_Bomb Khartoum
D1025E_Grizzly Bears
D1026E_Head Injuries
D1027E_Sh...Death Trial
D1028E_Unabomber Trial
D1029F_Ba...an Uprising
D1030F_Ephedra

APW19990911.0145.xml
APW19990912.0061.xml
APW19990913.0046.xml
APW19990913.0057.xml
NYT19990911.0237.xml
NYT19990913.0226.xml
NYT19990913.0305.xml
NYT19990913.0328.xml
NYT19990913.0348.xml
XIE19990911.0033.xml

```
<doc>
    <docno>
     APW19990911.0145
    </docno>
    <date_time>
     1999-09-11 15:00:12
    </date_time>
    <category>
     usa
    </category>
    <headline>
     Floyd May Miss Caribbean Islands
    </headline>
    <text>
     <p>
      THE VALLEY, Anguilla (AP) -- Hurricane Floyd on Saturday whirred
away from the Caribbean's Leeward Islands, but forecasters warned that
Puerto Rico and the Virgin Islands could be hit by heavy rains and
flooding.
     </p>
     <p>
      The hurricane itself was likely to pass well north of the
islands, meteorologists said. Floyd was expected to intensify, and it
was unlikely to hit land soon.
     </p>
     <p>
      It posed no immediate threat to the United States coast. Still,
it was recommended that people along the U.S. southeast monitor the
```

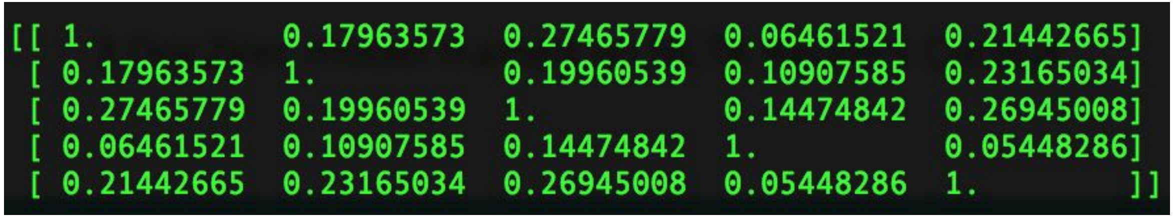
APW19990911.0145.xml

XML text - 2 KB

Created Sunday, April 16, 2017 at 5:31 PM
Modified Sunday, April 16, 2017 at 5:31 PM
Last opened Sunday, April 16, 2017 at 5:31 PM
Add Tags...

Content Selection - LexRank - Sentence Similarity

- All sentences → adjacency matrix by similarity score



```
[[ 1.          0.17963573  0.27465779  0.06461521  0.21442665]
 [ 0.17963573  1.          0.19960539  0.10907585  0.23165034]
 [ 0.27465779  0.19960539  1.          0.14474842  0.26945008]
 [ 0.06461521  0.10907585  0.14474842  1.          0.05448286]
 [ 0.21442665  0.23165034  0.26945008  0.05448286  1.        ]]
```

- Approaches to measure pairwise similarity:

  1.Cosine Similarity

  2.Cosine Similarity with TF-IDF (highest Rouge scores)

  3.Doc2Vec model trained on NLTK Reuters Corpus

Content Selection - LexRank - Algorithm

- Binarize similarity matrix $\boldsymbol{M}$ with 0.15 threshold.
- Implement LexRank using Power Method:

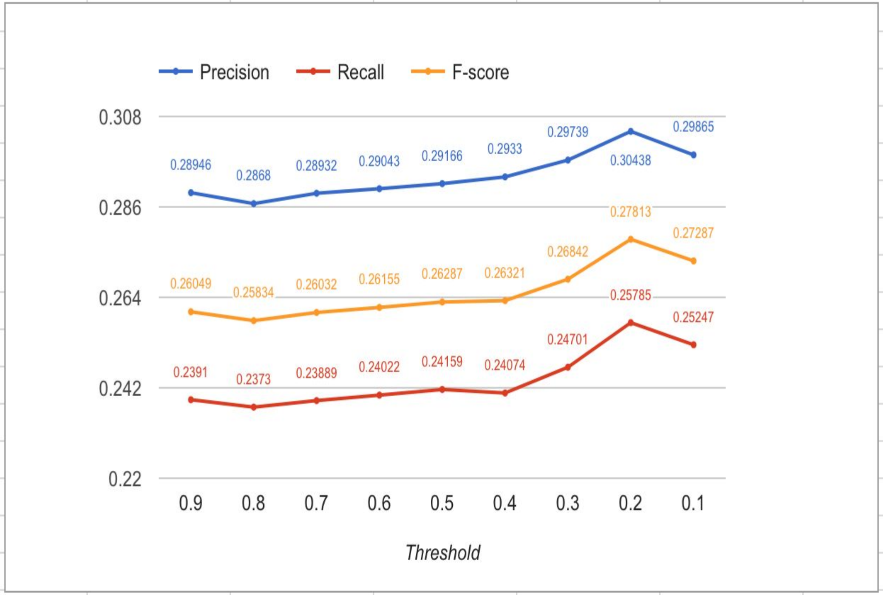
1. Initialize $\boldsymbol{p}_0$ vector with uniform distribution.
2. Iteratively update $\boldsymbol{p}_t$ such that $\boldsymbol{p}_t = \boldsymbol{M}^T\boldsymbol{p}_{t-1}$ until $||\boldsymbol{p}_t - \boldsymbol{p}_{t-1}||$ is sufficiently small.
3. Return $\boldsymbol{p}_t$.

Information Ordering & Content Realization

- Input: sentences sorted by score
- TF-IDF cosine similarity ordering [1]
- One Sentence per line
- Max 100 words

Threshold?

- At 1.0 => R-1 R = 0.239
- At 0.2 => R-1 R = 0.258 (+0.019)

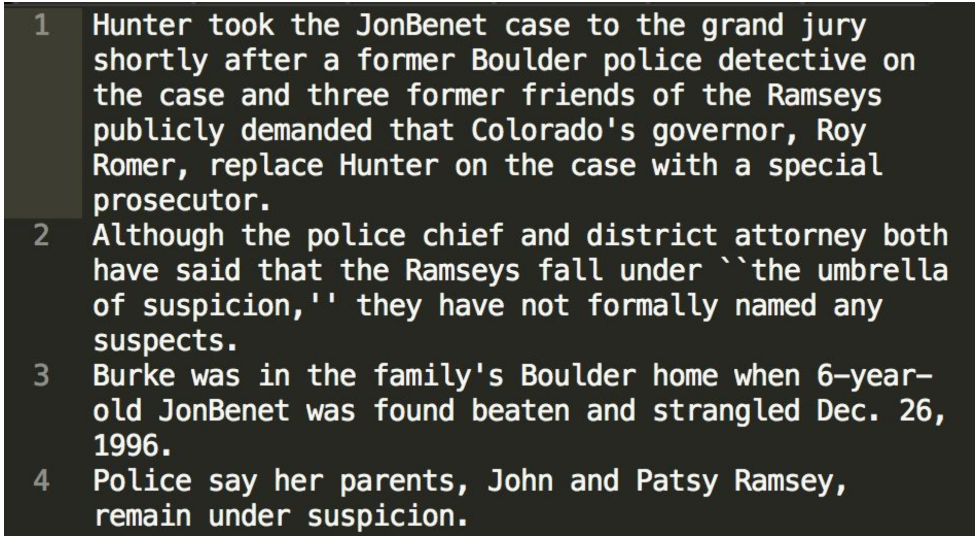


Results

- Runtime ~= 5 - 10 seconds
- Best results:
    - Content Selection: TF-IDF cos sim
    - Ordering: threshold = 0.2

ROUGE-L	Recall
ROUGE-1	0.25785
ROUGE-2	0.07108
ROUGE-3	0.02438
ROUGE-4	0.00847

Sample output
Topic: JonBenet Ramsey Murder



```
Hunter took the JonBenet case to the grand jury shortly after a former Boulder police detective on the case and three former friends of the Ramseys publicly demanded that Colorado's governor, Roy Romer, replace Hunter on the case with a special prosecutor.
Although the police chief and district attorney both have said that the Ramseys fall under ``the umbrella of suspicion,'' they have not formally named any suspects.
Burke was in the family's Boulder home when 6-year-old JonBenet was found beaten and strangled Dec. 26, 1996.
Police say her parents, John and Patsy Ramsey, remain under suspicion.
```

Issues and Successes

Success: Implementing tf-idf raised ROUGE-1 recall from 0.13 to 0.26.

Problems that need fixing:

- Meta-info like:
  - LITTLETON, Colo. (AP) --
  - NEW YORK _ The parents of Amadou Diallo plan to meet with the Bronx district
- Quotes without attributions
- Pronouns without reference
- Ordering of the sentences within summary
- Long sentences

References

[1] Radev, Dragomir R., et al. "MEAD-A Platform for Multidocument Multilingual Text Summarization." *LREC*. 2004.

[2] Erkan, Günes, and Dragomir R. Radev. "Lexrank: Graph-based lexical centrality as salience in text summarization." Journal of Artificial Intelligence Research 22 (2004): 457-479.

[3] Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." Text summarization branches out: Proceedings of the ACL-04 workshop. Vol. 8. 2004.

Questions ?