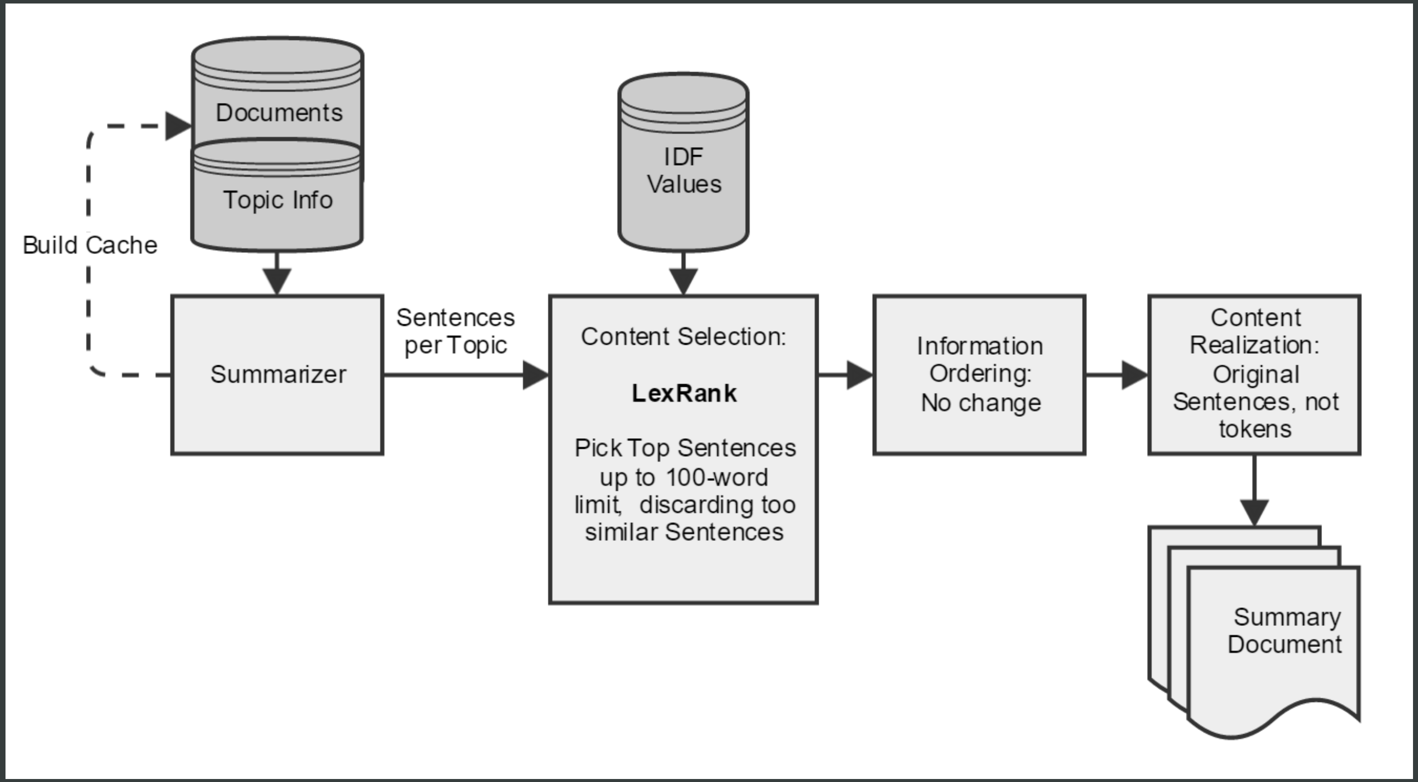
Multi-Document Summarization

DELIVERABLE 3: CONTENT SELECTION AND INFORMATION ORDERING

TARA CLARK, KATHLEEN PREDDY, KRISTA WATKINS



System Architecture

Our system is a collection of independent Python modules, linked together by the *Summarizer* module.

Content Selection: Overview

- Input: Documents in a Topic
- Algorithm: Query-focused LexRank
- Output: List of best sentences, ordered by rank

Query-Focused LexRank

- Nodes are sentences; edges are similarity scores
- Nodes: TF-IDF vector over each stem in the sentence

$$tf_t = \frac{number\ of\ times\ term\ t\ appears\ in\ doc}{total\ terms\ in\ doc}$$

$$idf_t = \log(\frac{total\ number\ of\ docs}{number\ of\ docs\ containing\ term\ t})$$

- Edges: Cosine similarity between sentences X and Y

$$\frac{\sum_{w \in x,y} tf_{w,x} tf_{w,y} (idf_w)^2}{\sqrt{\sum_{x_i \in x} (tf_{x_i,x}\, idf_{x_i})^2} * \sqrt{\sum_{y_i \in y} (tf_{y_i,y}\, idf_{y_i})^2}}$$

Prune edges below 0.1 threshold

Query-Focused LexRank: Relevance

- Compute the similarity between the sentence node and the topic query
- Uses tf-isf over the topic cluster sentences

$$rel(s|q) = \sum_{w \in q} \log(tf_{w,s} + 1) * \log(tf_{w,q} + 1) * \ isf_w$$

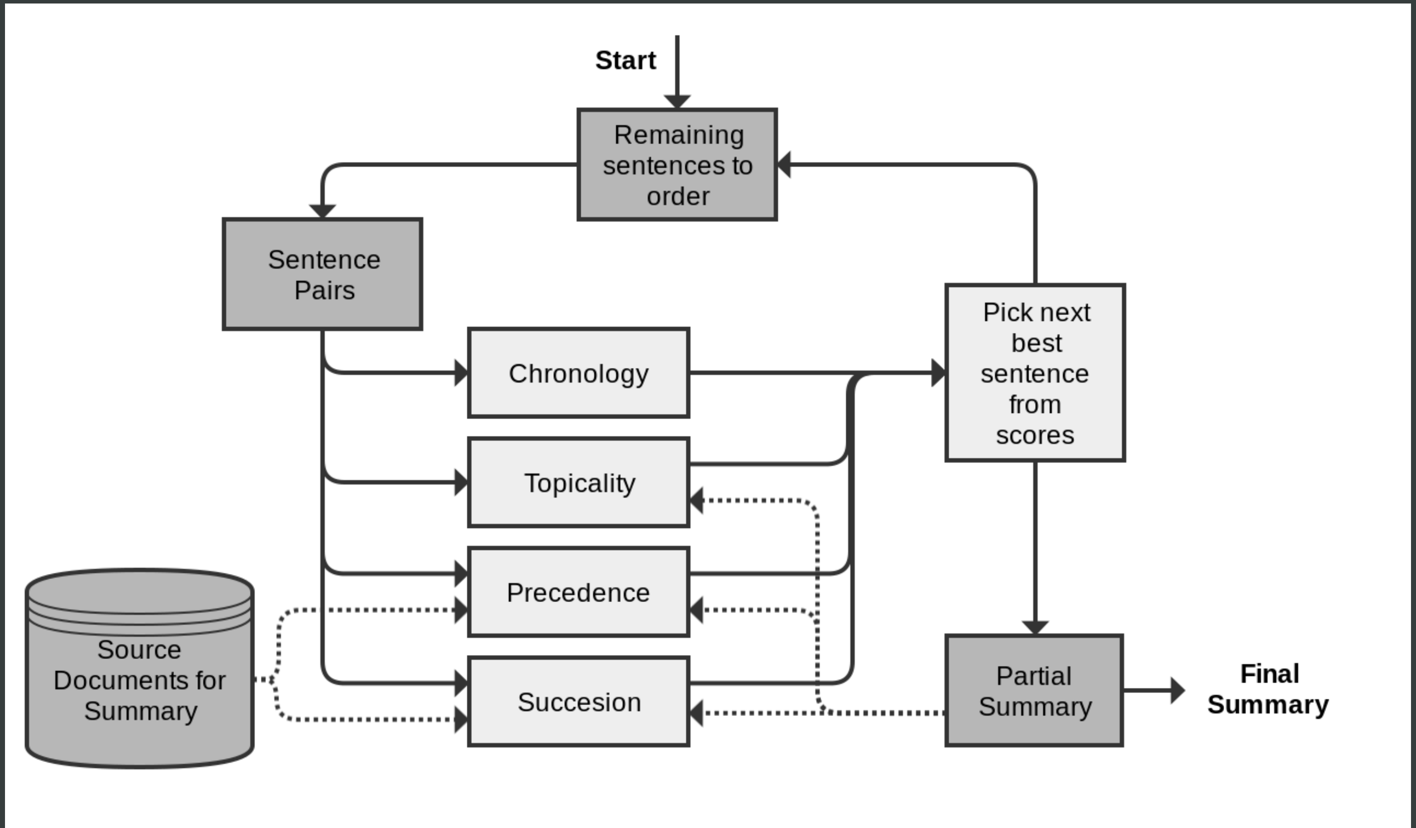
- This updates the whole LexRank similarity score:
- $p(s|q) = d \ * \frac{rel(s|q)}{\sum_{z \in C} rel(z|q)} + (1 - d) * \ \sum_{v \in C} \frac{sim(s,v)}{\sum_{z \in C} sim(z,v)} p(v|q)$
  - d is set to 0.95

Power Method

- Set normalized vector p
- Update p → dot product of transposed graph and current p
- Apply until convergence
- Apply scores from p vector to the original Sentence objects
- Return the best sentences, without going over 100 words or repeating yourself (cosine similarity < 0.95)

Information Ordering

- Input: List of sentences from content selection
- Algorithm: Expert voting (Bollegata et al.)
- Output: List of ordered sentences



Information Ordering

Architecture

Experts

- Chronology
- Topicality
- Precedence
- Succession

Chronology

- Inputs a pair of sentences
- Provides a score based on:
  - The date and time of each sentence's document
  - The position of each sentence within its document
- Votes for one of the sentences
- Ties return a 0.5 instead of a 1 or 0

Topicality

- Inputs a pair of sentences and the current summary
- Calculates the cosine similarity between each sentence and the sentences in the summary
- Votes for the sentence more similar to the summary
- Ties return 0.5

Precedence

- Inputs a pair of sentences
- Gathers all the sentences preceding each of these candidate sentences in their original documents
- The preceding sentence most similar to each candidate is extracted
- Whichever sentence has the higher similarity score gets the vote
- Ties receive 0.5

Succession

- Inputs a pair of sentences
- Gathers all the sentences succeeding each of these candidate sentences in their original documents
- The succeeding sentence most similar to each candidate is extracted
- Whichever sentence has the higher similarity score gets the vote
- Ties receive 0.5

Architecture

- Information Ordering module sends each possible pair of sentences to experts
- Uses the weights in Bollegata et al. to weight the votes from the experts
  - Chronology: 0.3335
  - Topicality: 0.0195
  - Precedence: 0.2035
  - Succession: 0.4435
- Scores >0.5 are added to Sent2; <0.5 to Sent1 for all sentence pairs
- Sentences are ordered by their final scores, from highest (most votes) to lowest

Content Realization

- Input: List of sentences from Information Ordering
- Trim the length of the summary to be 100 words, max
- Output: Write each sentence on a new line to the output file

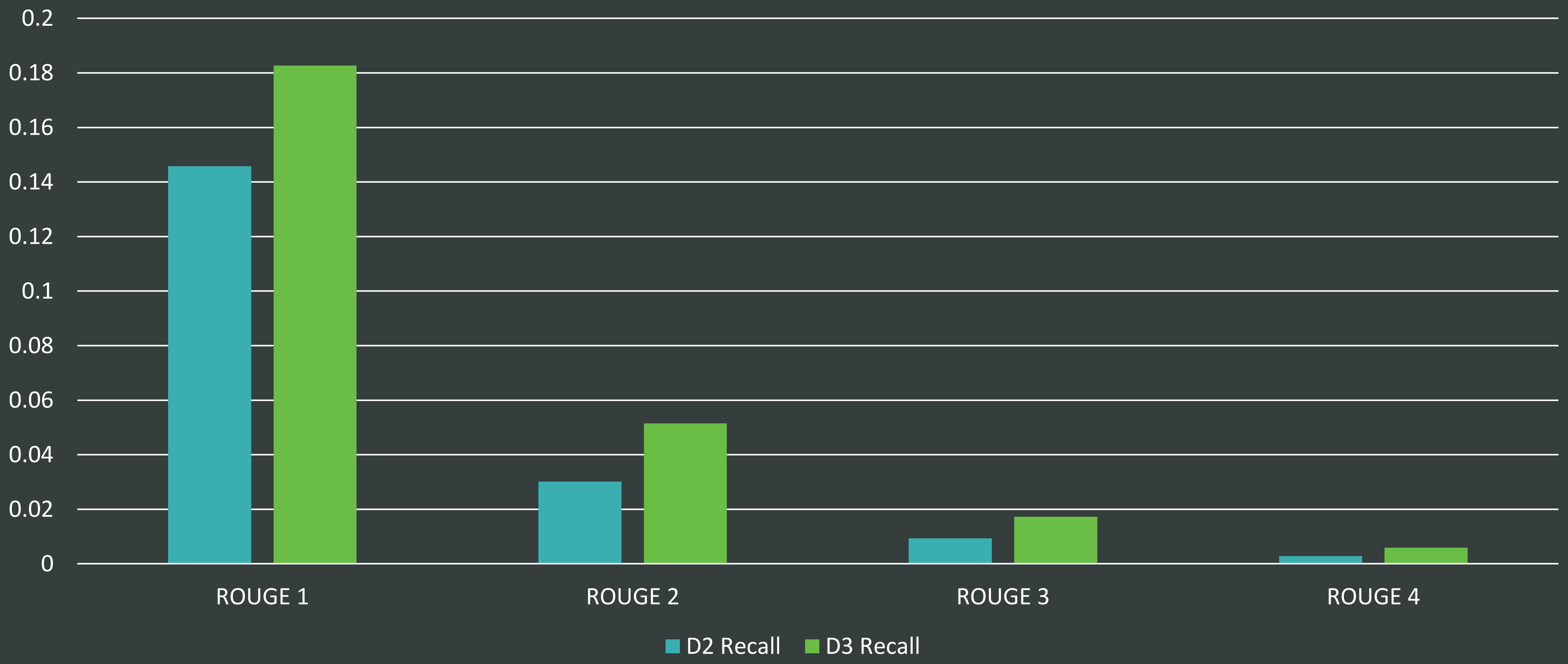
Issues and Successes

- Returning longer summaries
  - D2:
    - 26% of summaries were 1 sentence long
    - Average summary length: 2.087 sentences
    - Average word count: 77.370 words/summary
  - D3:
    - 0% of summaries are 1 sentence long
    - Average summary length: 3.565 sentences
    - Average word count: 85.217 words/summary
- Calculating IDF over a larger corpus

Issues and Successes

- Query focused LexRank
  - Large impact on training ROUGE scores
  - Smaller impact on devtest ROUGE scores
- Information ordering
  - Lost some good information due to moving 100-word cap to content realization
- Logistics:
  - Easily converted outputs, etc., by changing some parameters from “D2” to “D3”
  - Good team communication
  - Sickness 🤒

Results


0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
ROUGE 1
ROUGE 2
ROUGE 3
ROUGE 4
D2 Recall
D3 Recall

Results

	D2 Recall	D3 Recall
ROUGE-1	0.14579	0.18275
ROUGE-2	0.03019	0.05149
ROUGE-3	0.00935	0.01728
ROUGE-4	0.00285	0.00591

Related Reading

Regina Barzilay, Noemie Elhadad, and Kathleen R. McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. J. Artif. Int. Res., 17(1):35–55, August.

Danushka Bollegala, Naoaki Okazaki, and Mitsuru Ishizuka. 2012. A preference learning approach to sentence ordering for multi-document summarization.

Inf. Sci., 217:78–95, December.

Gunes Erkan and Dragomir R Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research, 22:457–479.

Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. 2007. The pyramid method: Incorporating human content selection variation in summarization evaluation. ACM Trans. Speech Lang. Process., 4(2), May.

Jahna Otterbacher, G¨unes¸ Erkan, and Dragomir R. Radev. 2005a. Using random walks for question focused sentence retrieval. In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT ’05, pages 915–922, Stroudsburg, PA, USA. Association for Computational Linguistics.

Karen Sparck Jones. 2007. Automatic summarising: The state of the art. Inf. Process. Manage., 43(6):1449–1481, November.

Questions?

West Coast Python Deliverable 3

Tracy Rohlin, Karen Kincy, Travis Nguyen

D3 Tasks

Tracy: information ordering, topic focus score with CBOW

Karen: pre-processing, lemmatization, background corpora

Travis: improvement and automation of ROUGE scoring

Summary of Improvements

- Changed SGML parser
  - Includes date info
  - Searches for specific document ID
- Improved post-processing with additional regular expressions
- Added several different background corpora choices for TF*IDF
- Added topic focus score and weight
- Implemented sentence ordering
- Fixed ROUGE bug

Pre-Processing

- Added more regular expressions for pre-processing
  - Still too much noise in input text
    - Issue with 100-word limit in summaries
    - More noise = less relevant content
- Output all pre-processed sentences to text file for debugging
  - Allowed us to verify quality of pre-processing
  - Checked for overzealous regexes
  - Results still not perfect

Additional Regexes

```
line = re.sub("^\&[A-Z]+;", "", line)
line = re.sub("^[A-Z]+.*_", "", line)
line = re.sub("^[_]+.*", "", line)
line = re.sub("^[A-Z]+.*_", "", line)
line = re.sub("^.*OPTIONAL.*\)", "", line)
line = re.sub("^.*optional.*\)", "", line)
line = re.sub("^.*\(AP\)\s+--", "", line)
line = re.sub("^.*\(AP\)\s+_", "", line)
line = re.sub("^.*[A-Z]+s+_", "", line)
line = re.sub("^.*\(Xinhua\)", "", line)
line = re.sub("^\s+--", "", line)
```

- Tried to remove:
  - Headers
  - Bylines
  - Edits
  - Miscellaneous junk

Lemmatization

- Experimented with lemmatization
  - WordNetLemmatizer from NLTK
- Goal: collapsing related terms into lemmas
  - Should allow more information in each centroid
- Results: lemmatizer introduced more errors
  - “species” -> “specie”; “was” -> “wa”
  - WordNetLemmatizer takes “N” or “V” as optional argument
  - Tried POS tagging to disambiguate nouns and verbs

Background Corpus

- Need background corpus for IDF calculation of TF*IDF
- Initially used “news” subset of Brown corpus
  - Too small (~40 documents)
- Added two alternative background corpora from NLTK
  - Entire Brown corpus
  - Reuters corpus
- Reuters resulted in best ROUGE scores
  - Likely due to news domain of Reuters

Topic Score

- Added topic score using Gensim’s Continuous Bag of Words (CBOW) model
- Total summed score multiplied by weight given to topic words
  - Grid search found that any weight other than 1 caused a decrease in ROUGE scores
  - Might be worth examining more in D4

Information Ordering

Based on Bollelaga, et al.'s 2011 paper about chronological ordering

Original formula

$$\mathrm{PREF}_{chro}(u, v, Q) = \begin{cases} 1 & T(u) < T(v) \\ 1 & [D(u) = D(v)] \wedge [N(u) < N(v)] \\ 0.5 & [T(u) = T(v)] \wedge [D(u) \neq D(v)] \\ 0 & \mathit{otherwise} \end{cases}.$$

Orders by date and then by location in document

Ordering in Our System

- System refers ordering based on whether sentence is first in a document
  - No tie breaking between two first sentences, i.e., original order kept
- If not first sentence, order based on publication date
  - Tie breaking based on sentence position
- Results in more readable summaries than ordering based on date alone

First Sentence + Date Ordering:

1. Seven weeks before Merck & Co. pulled the arthritis drug Vioxx off the market because of safety concerns, federal drug regulators downplayed the significance of scientific findings citing the increased risks, documents released Thursday show.

2. The FDA said such discussions are typical before scientific findings are published.

3. FitzGerald also challenged Pfizer's contention that no science shows increased risk from Celebrex.

4. But the study was halted when it indicated a heightened risk of cardiovascular complications.

5. For patients on blood thinners such as Coumadin, the combination could be highly risky without proper supervision.

Date-Only Ordering:

2. The FDA said such discussions are typical before scientific findings are published.
1. Seven weeks before Merck & Co. pulled the arthritis drug Vioxx off the market because of safety concerns, federal drug regulators downplayed the significance of scientific findings citing the increased risks, documents released Thursday show.
3. FitzGerald also challenged Pfizer's contention that no science shows increased risk from Celebrex.
5. For patients on blood thinners such as Coumadin, the combination could be highly risky without proper supervision.
4. But the study was halted when it indicated a heightened risk of cardiovascular complications.

D2 Bug: ROUGE Script

- Bug
  - Each system summary treated as its own test set
  - Each system summary had its own alphanumeric code
  - Should have set one alphanumeric code per test run
- Fix
  - System summaries corresponding to one test run share same alphanumeric code

D2 Bug: Randomized Summaries

- Scores and summaries randomized
  - Only on Patas, not when run locally
  - Issue discovered during parameter optimization
  - Had to output all sentences and scores to debug
- Bug: input ordering not preserved
  - JSON file loaded into dictionary
  - Switched to OrderedDict

Results...

The bad news:

- Highest-scoring summaries decreased from 0.375 to 0.35841 for ROUGE-1
- Still some zero scores for ROUGE-3 and ROUGE-4

The good news:

- Improvement across all scores
- Standard deviation slightly decreased for ROUGE-1 & 4, by less than 1%

Average ROUGE Scores: D2 vs. D3

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
D2	0.23654	0.06117	0.01829	0.00618
D3	0.25363	0.07330	0.02577	0.01001
Difference	+1.709%	+1.213%	+0.748%	+0.383%

Standard Deviation of ROUGE Scores

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
D2	0.07825564137	0.03582682832	0.02329799339	0.01712149597
D3	0.07370586712	0.03780649756	0.02443678615	0.01703135117
Difference	-0.454977425%	+0.197966924 %	+0.113879276 %	-0.00901448%

Summary: “Giant Panda”

Forest coverage in southwestern Sichuan Province has increased to 27.94 percent from 24.3 percent in 2003, making the region, a major habitat of giant pandas, a greener home, according to the local government.

China has applied to the United Nations to make giant pandas' natural habitat in southwestern Sichuan province a world heritage area to help protect the endangered species, state press reported Tuesday.

Nature preserve workers in northwest China's Gansu Province have formulated a rescue plan to save giant pandas from food shortage caused by arrow bamboo flowering.

Future Ideas

Further improve pre-processing

Use tree parsing [Zajic et al. (2006)] to do sentence compression, maybe include entity grid [Barzilay et al. (2005)]

Incorporate machine learning techniques to learn best content to pick for each cluster, perhaps Word2Vec

Multi-document Summarization

Ling 573 group project by
Joanna Church, Anna Gale, Ryan Martin

Updated for D3
May 2017

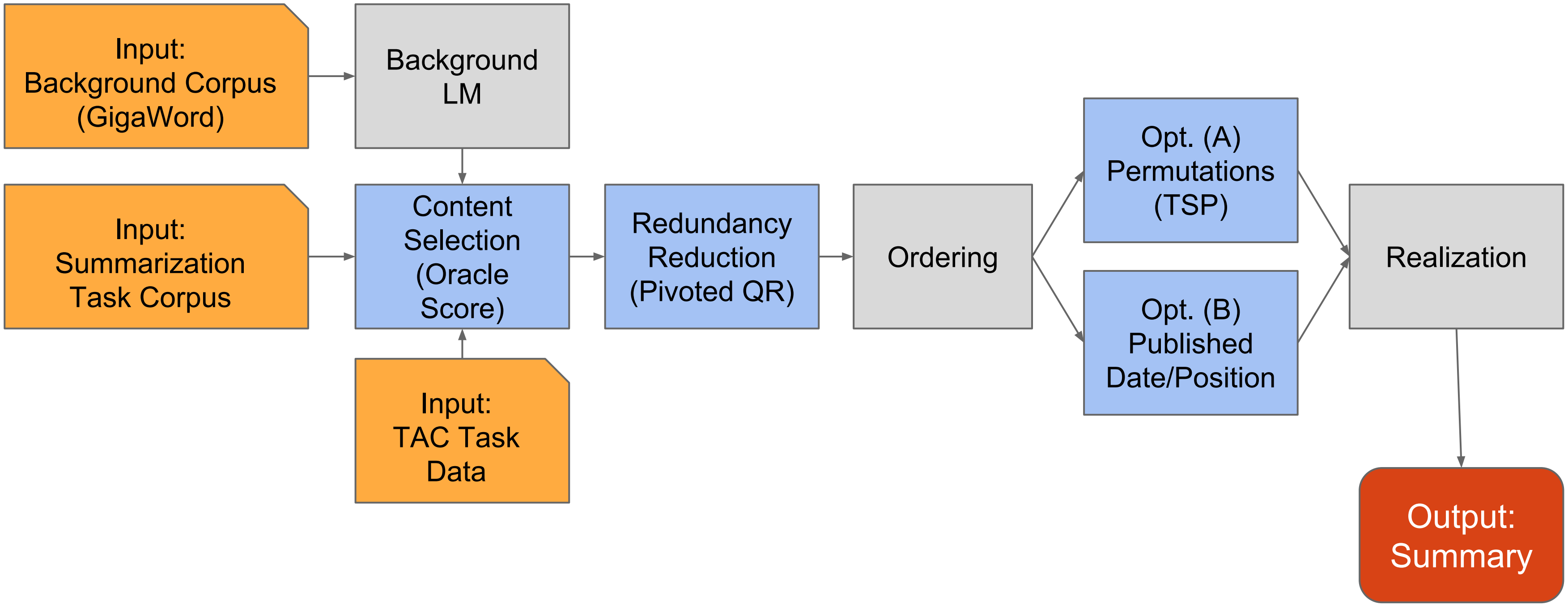
Overview

Our Inspiration

John M. Conroy, Judith D. Schlesinger, and Dianne P. O'Leary. 2006. **Topic-focused multi-document summarization using approximate oracle score**. In *Proceedings of the COLING-ACL on Main Conference Poster Sessions*, COLING-ACL '06, pages 152-159, Stroudsburg, PA, USA. Association for Computational Linguistics.

John M. Conroy, Judith D. Schlesinger, Dianne P. Oleary, and Jade Goldstein. 2006b. **Back to Basics: Classy 2006**. In Proceedings of DUC, volume 6, page 150.

System Architecture



Updated Architecture

Updates

Content Selection

- Query terms: Stemming (Porter) was added to query term selection. (“*avalanche*” and “*avalanches*” are now comparable).
- Resolved issues with underflow in background language model LLR calculations.
- Added smoothing in background language model for OOV terms found in signature term selection.
- Better parsing of corpora:
  - Remove datelines
  - Remove non-content meta-data
  - Remove leading non-word characters

Redundancy Reduction

- Pivoted QR decomposition of the term-sentence matrix
    - Doesn’t work particularly well with identical or nearly-identical sentences.
    - The “importance” of the second sentence in an identical pair is discounted, but the sentence is not necessarily removed. It may be selected in the next iteration.
- Added a high-threshold cosine similarity test before Pivoted QR to remove identical pairs.

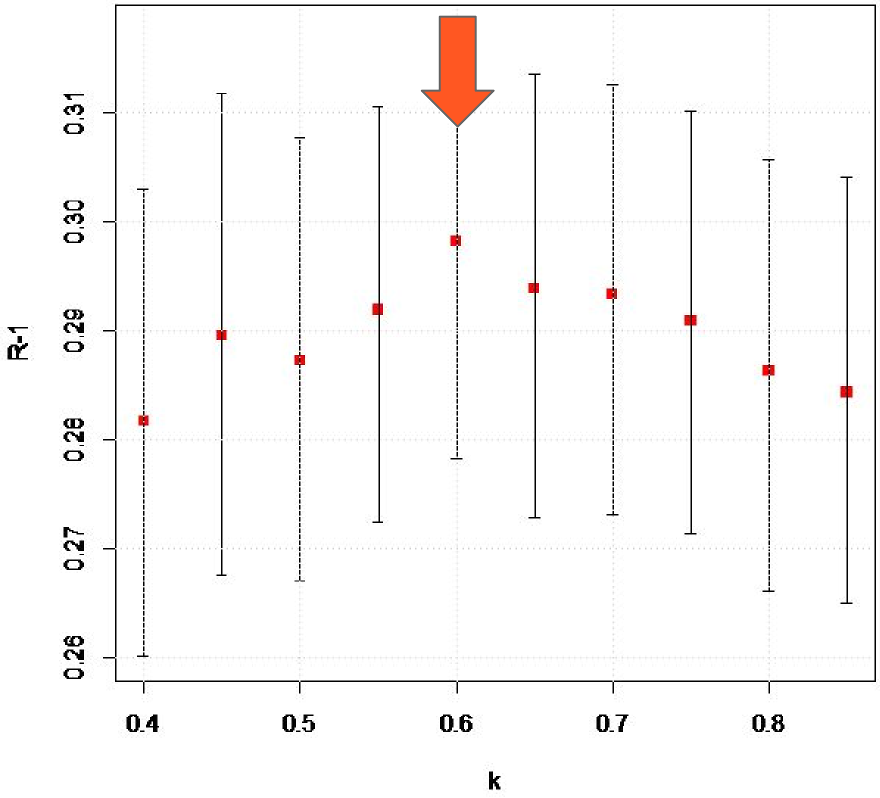
Parameter Optimization

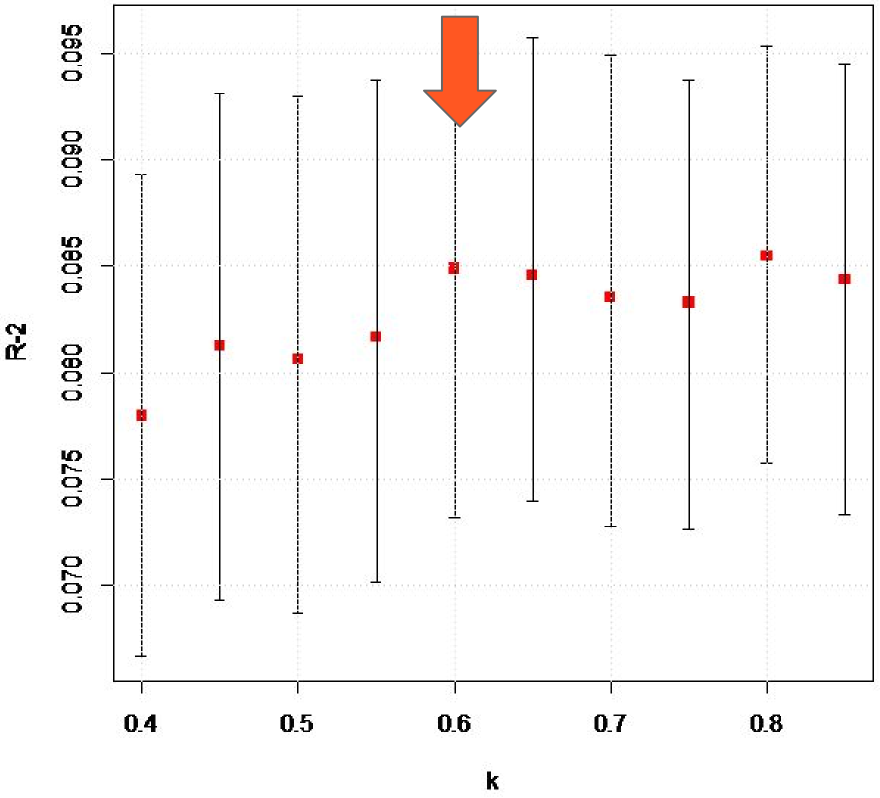
- Previous version used default 0.50 for the value of k.
- Optimize k based on training set.

$$P(t|\tau) = kq_t(\tau) + (1-k)s_t(\tau)$$

$$k \in [0,1]$$

Optimization (Best k ~ 0.60)





Information Ordering

Information Ordering Strategy

We compared two ordering implementations:

1. Analysis of permutations (TSP) -- Calculate a distance function based on coherence and salience between sentences. We calculate this distance measure for every permutation of sentences, and choose the lowest scoring grouping of sentences as the best summary ordering.
2. Sorting sentences based on published date/time and sentence position

Ordering Analysis

- Permutation Method (TSP):
  - Advantage: Good cohesion between adjacent sentences.
  - Disadvantage: First and last sentences often feel “out of place”. Performance issues when the number of sentences is too large.
- Date/Position:
  - Advantage: First sentence is usually a good selection (feels natural).
  - Disadvantage: Subsequent sentences may lack cohesion.
- Next Steps:
  - Select a fixed lead sentence (Salience and/or Document Position), then use permutation method to order the remaining sentences in the summary.

Content Realization

Content Realization

- Select top candidate sentences from ordering step (not to exceed 100 tokens).

More to come...

Results

ROUGE

System	R-1	R-2	R-3	R-4
D2 (devtest)	0.1576	0.0218	0.0048	0.0018
D3 (devtest)	0.2744	0.0788	0.0316	0.0136
D3 (training)	0.2933	0.0835	0.0316	0.0136

Examples

Examples

The trial for one of two men accused in the beating death of University of Wyoming student Matthew Shepard will begin with jury selection March 24. Authorities said Henderson and McKinney posed as homosexuals and lured the 5-foot-2, 105-pound Shepard out of a bar, kidnapped him, pistol-whipped him and stole $20. Seven people were dismissed Thursday as jury selection continued in the trial of a man accused in the beating death of gay college student Matthew Shepard. Russell Henderson was a witness to the beating of Matthew Shepard,'' attorney Wyatt Skaggs told prospective jurors as Henderson's trial opened Wednesday.

(D1045-A.M.100.H.1, R-1 0.38583, R-2 0.088)

They include former soldiers who fought in areas sprayed with Agent Orange by U.S. aircraft and children who were later born with deformities. In *the institute's* last review of scientific research in 1996, six other diseases were listed with ``limited or suggestive evidence of association'' to Agent Orange, other herbicides or the contaminant dioxin. In *the institute's* last review of scientific research in 1996, six other diseases were listed as those with ``limited or suggestive evidence of association'' to Agent Orange, other herbicides or the contaminant dioxin.

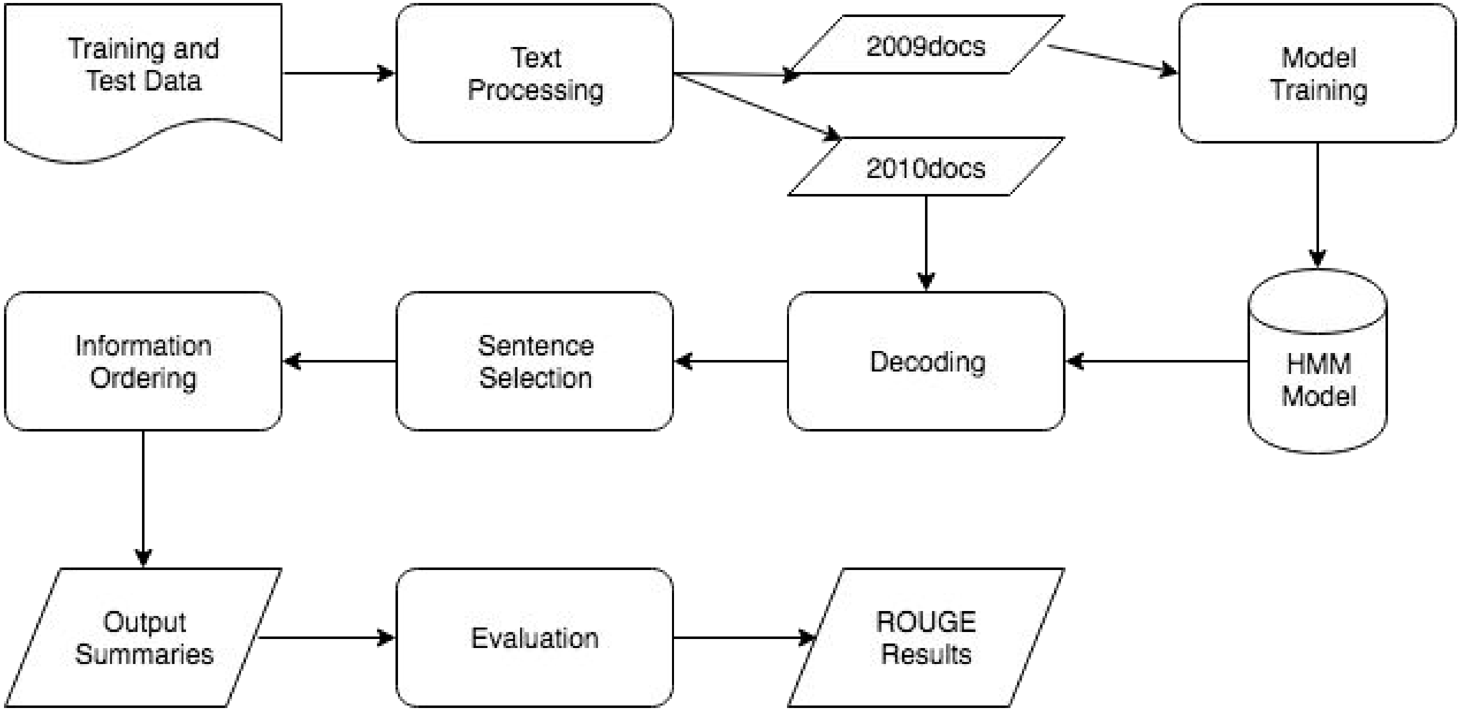
(D1035-A.M.100.G.1, R-1 0.16667, R-2 0.04545)

Thanks for listening!

D3: 2 Hidden 2 Ordered

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

System architecture



Preprocessing

- Processing 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 headline and text
  - Parsed Using NLTK
  - Sentences are lowercased, stopworded, & lemmatized*

```
{
"topicID":"",
"title":"",
"narrative":"",
"doc-setID":"",
"docIDs":[list of doc ids]
"doc-paths":[list of doc paths]
"Text":[{dict of par#:{sentences}}]
"summaries":[list of summaries]
}
```

* Or will be, anyway…

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
  - $P(\text{sentence}|O) = P(\text{feature}_1|O)*P(\text{feature}_2|O)*\ldots*P(\text{feature}_N|O)$
  - 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, article dates, and probability for postprocessing
    - E.g. $sentence_1$/date/I/0.35 $sentence_2$/date/O/0.27 … $sentence_N$/date/O/0.11

Information Ordering

- Initially relevance-based ordering
- (Semi-)exhaustive search of possible combinations of I-tagged sentences
- Possible outputs ranked based on:
  - Precedence: how much does each sentence look like the following sentence’s original previous context (stopped and lemmatized, using cosine similarity)
  - Succession: how much does each sentence look like the preceding sentence’s original following context (stopped and lemmatized, using cosine similarity)
  - Chronology: do the sentences appear in chronological order based on publishing date
  - LLR (for cases where not all sentences may appear in the final summary due to the word count constraint)

Information Ordering

- Exhaustive search works as long as the number of included sentences < 10, otherwise search space is too great (varies from 3-40+!)
  - Currently, reducing search space by picking sentences with highest LLR
  - Future: reduce search space by topic-clustering and picking 1-2 sentences from each cluster
- More experimentation with weighting of each score category
- Size of previous/following contexts
  - Currently includes (stopped, lemmatized) 2 sentences of context

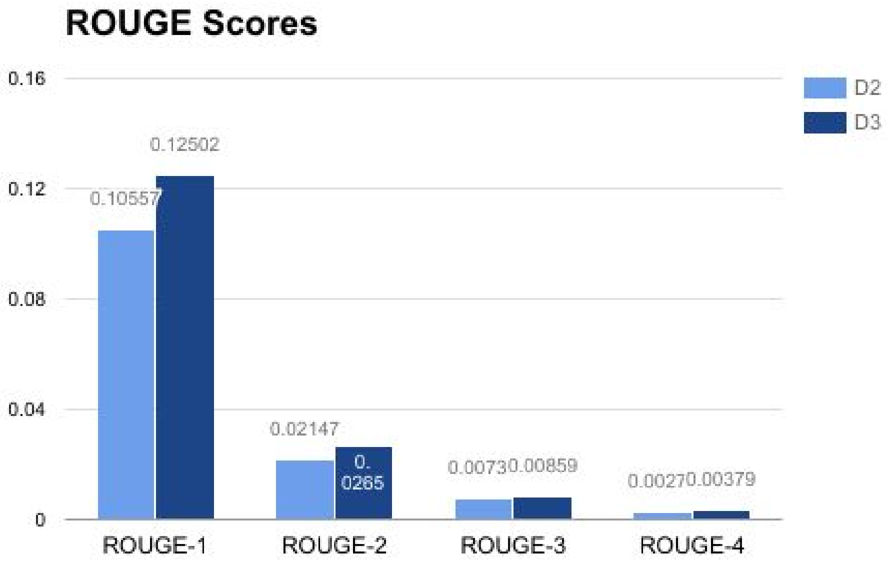
Content Realization

- Sentences are currently printed without changing the string as it appears in the text
- Future improvements to explore:
  - Incorporating pre-processed text in each module
  - Coreference resolution
  - Removing starting adverbials
  - Removing parenthetical text
  - Removing location information from first sentences

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

An old summary - Not good!

Mining is key to Peru 's economy , which has been growing at about 4 percent annually since President Alejandro Toledo took office in 2001 . Mining provides about half of Peru 's more than US $ 11 billion (euro8.9 billion) in exports this year , but directly employs only about 70,000 of Peru 's 27 million people , mostly in remote regions .
`` There may be an issue with frogs , that they are not warm and fuzzy , '' she said .
(Begin optional trim)
(End optional trim)

A new summary - Better!

Gascon , at Conservation International , said `` there are some actions we can take today to prevent the immediate extinction of many species as we work on a longer term solution . ''
These include creating parks and ecological reserves , working to reduce emissions that contribute to climate change and breeding animals in captivity in order to sustain vulnerable species .
The authors attributed some of the declines , which have occurred mainly in tropical areas , to habitat loss or to humans collecting animals for food , medicine , or pets .

Issues and Successes

Issues/Future Work:

- Inconsistencies in the Documents
- Gold summaries are Abstractive -> cosine similarity to attempt handling
  - Experiment with other gold creation methods: similarity threshold vs 1-best
- Inclusion of word salad sentences that should be ignored in preprocessing
  - Have done preprocessing
  - Now need to incorporate it into model
- More complex content realization
- Remove location information from beginning of articles
- Coreference issues (first mentions, multiple mentions)

Successes:

- It runs end to end :D
- No more blank summaries
- Previously bad summaries look much better now

Acknowledgements

We would like to thank Markov, model hide and seek champion.

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).