Final Summarization System
LING 573 Deliverable #04

Eric Garnick (egarnick@uw.edu)
John T. McCranie (jtm37@uw.edu)
Olga Whelan (olgaw@uw.edu)

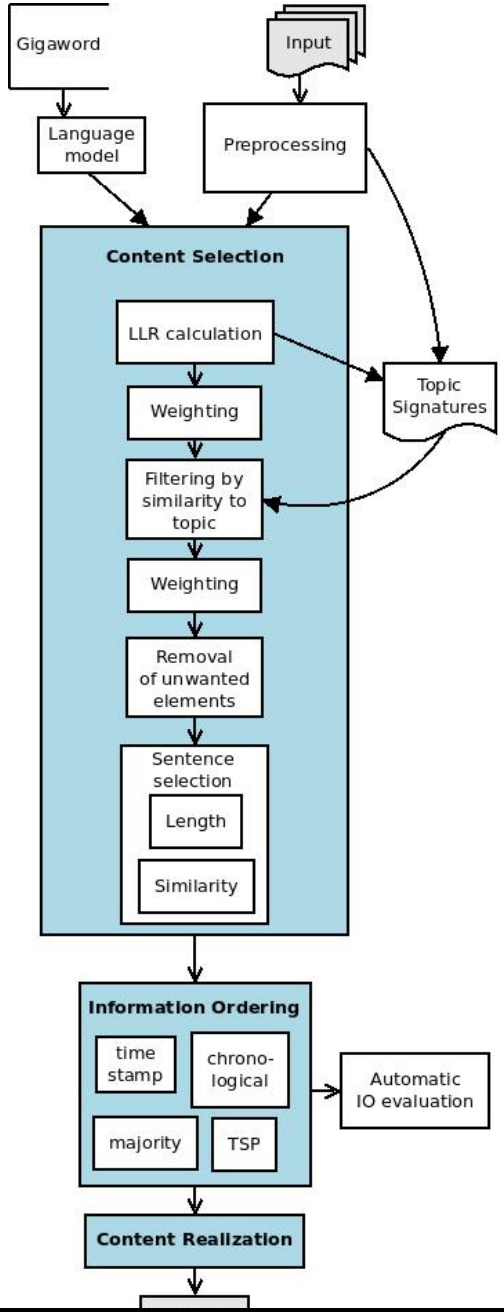
"To summarize the summary of the summary: people are a problem."
(Adams, 1980)

June 04, 2015

Outline

- System Overview
- Content Selection
- Information Ordering
- Content Realization
- Issues and Deadends
- ROUGE scores
- Influences
- Future Directions

System Overview



- Python 3, nltk
- Gigaword corpus
- 3 high-level, independent components: content selection, ordering, realization
- Barzilay, et. al. 2002
- final ROUGE F scores for evaltest data:

R1	R2	R3	R4
0.31068	0.09708	0.03683	0.01701

Content Selection

- log likelihood ratio (Gigaword corpus)
- tokenization, data cleanup
- topic signatures (cluster title, article headlines)
- tf-idf cosine similarity scores

Information Ordering

added three different methods to re-order summaries, all part of a discrete step:

- chronological (Barzilay, 2002; publication date and relative position in document)
- majority (Barzilay, 2002; heuristic topological search, relative sentence positions)
- traveling sales person (Conroy, 2006; consider all possible pairs, minimize distances)

Information Ordering Evaluation

Earlier we evaluated reorderings by random spot check, gauging an overall "feel", which ranked majority highest.
Now we have two different review methods:

- manual, semi-blinded evaluation, 3 levels of comprehensibility (Barzilay, 2002)
- automatic, correlation to gold-standard summaries

Information Ordering Evaluation: Manual

System	Comprehensible	Somewhat Comp.	Incomp.
Original	8	9	4
Majority	11	7	3
Chronological	6	11	4
Similarity	6	10	5

Majority the least bad?

Information Ordering Evaluation: Automatic

Re-ordering processes applied to the gold-standards, and then calculated Spearman's rank correlation.

Dataset	Algorithm	rho	p-value
evaltest	chrono	0.0604	0.3190
evaltest	majority	-0.0384	0.5270
evaltest	pass	0.8825	4.4456e-91
evaltest	tsp	0.4102	1.5206e-12
devtest	chrono	0.03787	0.5265
devtest	majority	0.0379	0.5259
devtest	pass	0.9816	1.3553e-203
devtest	tsp	0.2655	6.1884e-06

TSP the least bad?

Content Realization

Sentence filtering:

- pruning extraneous words (email addresses, phone numbers, paraentheticals, &c.) found by regexes
- remove attributions
- remove adverbials
- remove ALL CAPS items (but with an exception list)
- exclude sentences without any verbs (as determined by SpaCy POS tagging)

Issues and Deadends

Items that were tried and abandoned, or were deemed unhelpful:

- co-reference resolution
- sentence position weighting
- stemming / lemmatization
- augmented chronological re-ordering
- various parameter tunings

ROUGE Scores

ROUGE scores for final system, on devtest and evaltest datasets:

System	Precision	Recall	F
R1	0.29670	0.28859	0.29215
R2	0.08910	0.08678	0.08780
R3	0.03275	0.03209	0.03237
R4	0.01477	0.01452	0.01463

System	Precision	Recall	F
R1	0.31154	0.31115	0.31068
R2	0.09698	0.09761	0.09708
R3	0.03644	0.03742	0.03683
R4	0.01663	0.01752	0.01701

Influences

Adams, Douglas, 1980
The Restaurant at the End of the Universe

Barzilay, et. al. 2002
Inferring strategies for sentence ordering in multidocument news summarization,

Conroy et. al., 2006
Left-Brain/Right-Brain Multi-Document Summarization.

Future Directions

If we had more time, we would:

- add a workable co-reference system
- implement a machine-learning algorithm to use sentence features we currently weight by tuning

Document Summarization

LING 573, Spring 2015

Jeff Heath
Michael Lockwood
Amy Marsh

Documents with a common topic are extracted from the XML corpora by their Document IDs

Document
Sentence1
Sales of...
dt=yyyymmdd
psn=1
Sentence2
But its...
dt=yyyymmdd
psn=2
...

A document's text is broken into sentences and is added (a paragraph at a time) to a Document object. Sentences are stored as Sentence objects, which store the sentence string, a tokenized list of words in the sentence, the document date, the position in the paragraph – everything that might help us in our summarization

DocumentSet
Document1
Sentence1
Sentence2
...
Document2
Sentence1
Sentence2
...
...
Documentn
Sentence1
Sentence2
...

A DocumentSet object is built with all of the Documents in this topic in preparation for summarization

Realization/Compression:
remove unnecessary text elements

User-selected summarization routine

Random Summarization
QR Summarization
LLR Summarization
CLASSY Summarization

Parameter selection

Word counts over all corpora

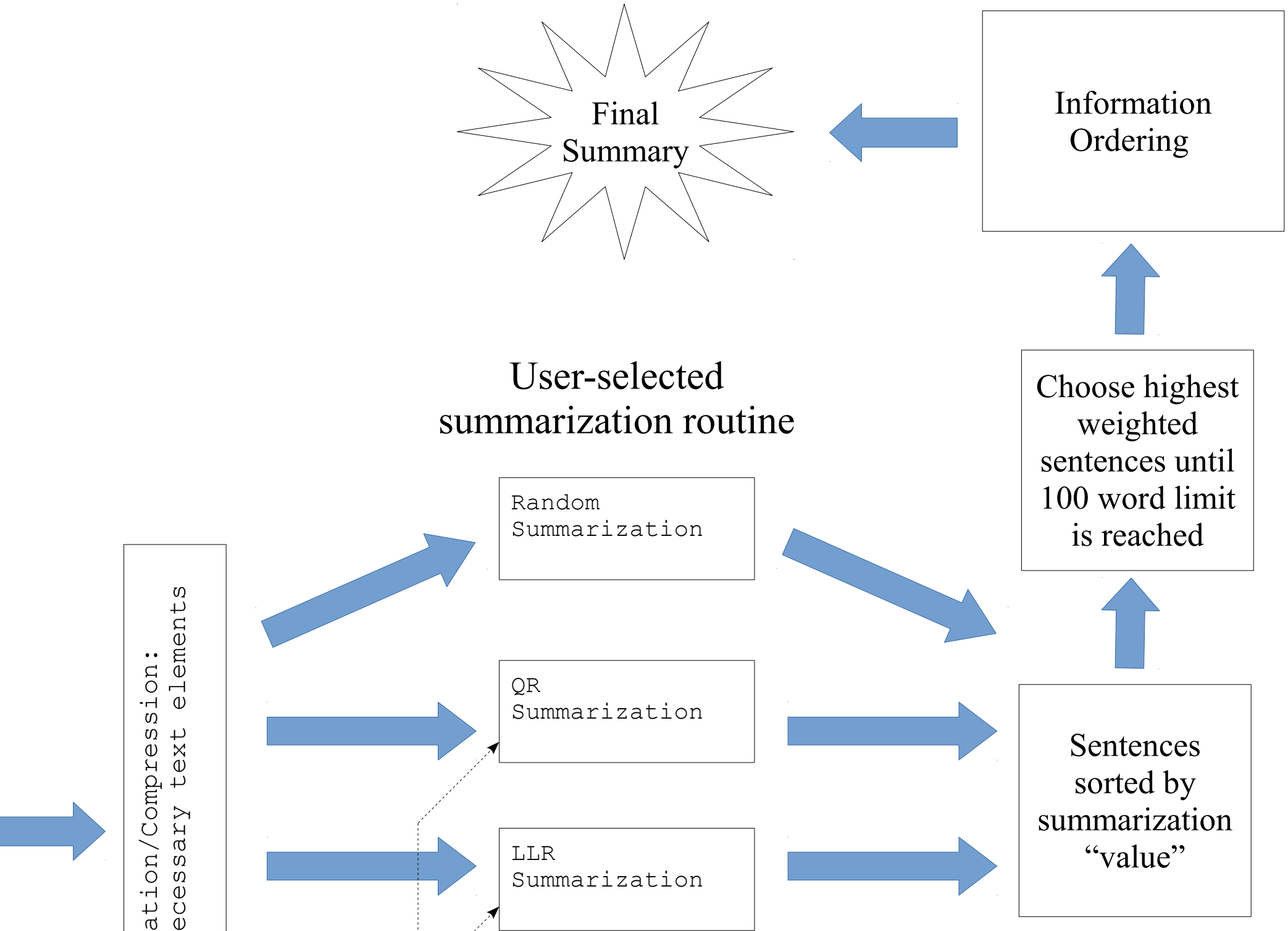
Training on DocumentSets with their model summaries from training corpus (2009)

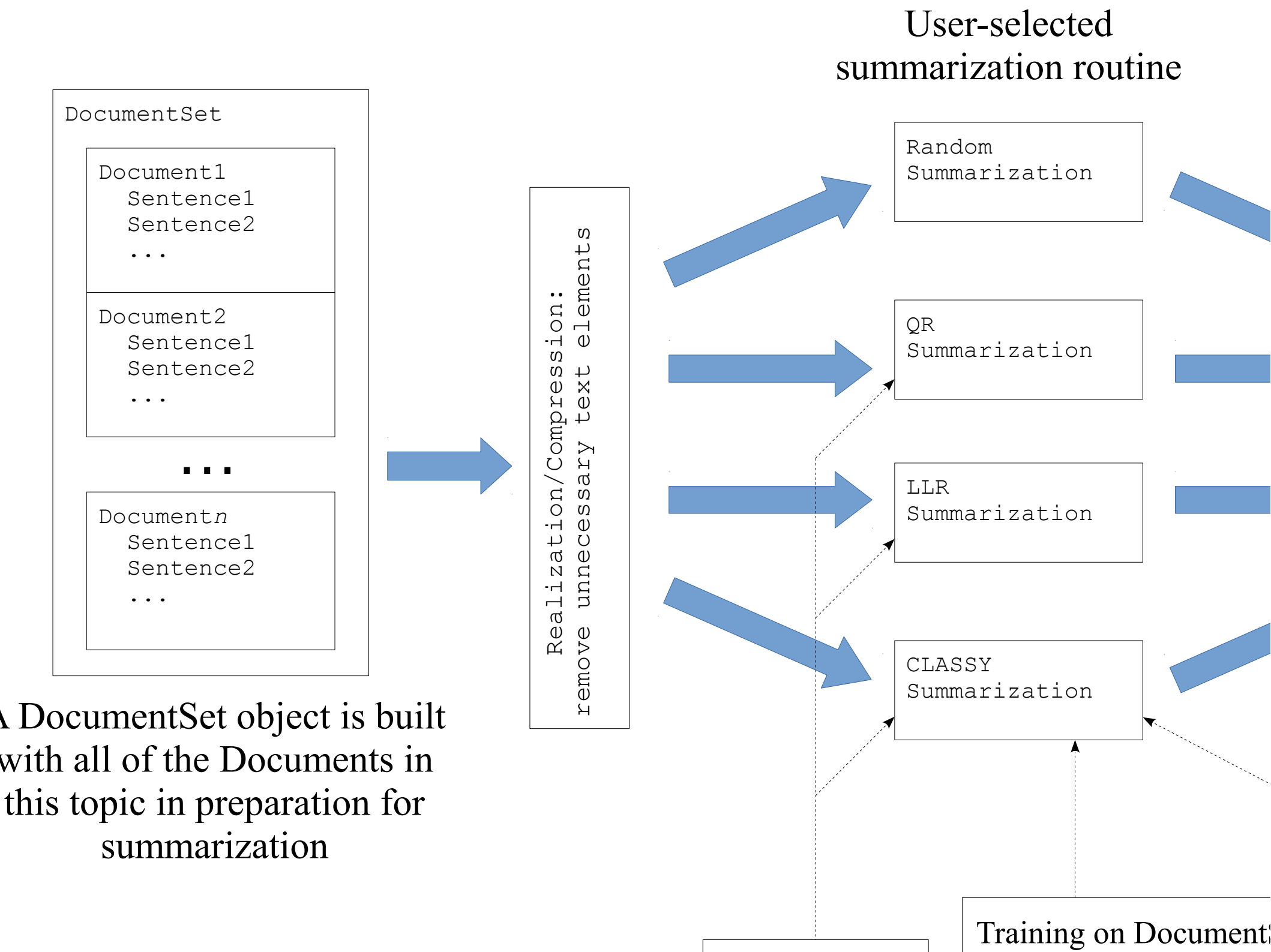
Sentences sorted by summarization "value"

Choose highest weighted sentences until 100 word limit is reached

Information Ordering

Final Summary


Final Summary
Information Ordering
User-selected summarization routine
Random Summarization
QR Summarization
LLR Summarization
Choose highest weighted sentences until 100 word limit is reached
Sentences sorted by summarization "value"
ation/Compression:
ecessary text elements


User-selected summarization routine
DocumentSet
Document1
Sentence1
Sentence2
...
Document2
Sentence1
Sentence2
...
...
Documentn
Sentence1
Sentence2
...
Realization/Compression:
remove unnecessary text elements
Random Summarization
QR Summarization
LLR Summarization
CLASSY Summarization
A DocumentSet object is built with all of the Documents in this topic in preparation for summarization
Training on Document

Documents with a common topic are extracted from the XML corpora by their Document IDs

Document
Sentence1
Sales of...
dt=yyyymmdd
psn=1
Sentence2
But its...
dt=yyyymmdd
psn=2
...

A document's text is broken into sentences and is added (a paragraph at a time) to a Document object. Sentences are stored as Sentence objects, which store the sentence string, a tokenized list of words in the sentence, the document date, the position in the paragraph – everything that might help us in our summarization

DocumentSet
Document1
Sentence1
Sentence2
...
Document2
Sentence1
Sentence2
...
...
Documentn
Sentence1
Sentence2
...

A DocumentSet object is built with all of the Documents in this topic in preparation for summarization

Realization/Compression:
remove unnecessary text elements

User-selected summarization routine

Random Summarization
QR Summarization
LLR Summarization
CLASSY Summarization

Parameter selection

Word counts over all corpora

Training on DocumentSets with their model summaries from training corpus (2009)

Sentences sorted by summarization "value"

Choose highest weighted sentences until 100 word limit is reached

Information Ordering

Final Summary

Location of Content Realization

- Mostly based on sentence trimming techniques of CLASSY 2006
- Makes sentences more topic-focused, so better done before summarization
- Anaphora resolution not implemented

→ Realize content before summarization

Content Realization

- Remove datelines, like “NEW YORK (AP) -- ...”
- Remove parenthetical expressions (also those marked off by – ... –)
- Remove age specifications, like “Smith, 67, ...”
- Remove entire sentences which contain:
  - “COPYRIGHT”, “New York Times” or “Associated Press”
  - phone numbers, web or email addresses
  - less than 5 words

Content Realization

- For more sophisticated realization, we attempted using the Stanford parser

  “(ROOT (S (ADVP (NP (CD **Two**) (NNS **days**)) (RBR **earlier**)) (, ,) (NP (NP (DT a)**...**”

  - too much overhead to parse sentences individually
  - it resplit multiple sentences, sometimes poorly
- Resorted to regular expressions
- But parses gave us useful information

Content Realization

- Remove time expressions at the beginning of the sentence, like “Two years ago,” “A month earlier,” etc.
- Remove adverb phrases at the beginning of the sentence, like “However,” “Not surprisingly,” “Now,” “And likewise,” etc.
- Remove attributions at the beginning or end of sentences, like “Smith said that …” “, CNN reported.” “, authorities said.”

Content Realization

- Remove “however” and “also” everywhere
- Remove relative clauses marked off with commas, such as:
  - That storm, which developed into a full-fledged hurricane Monday, was...
  - said the official, who asked not to be identified.

Content Selection - LLR

- Use LLR to find content words in document set
- Select sentences with highest percentage of content words for the summary
- Cosine similarity to avoid redundancy

Content Selection – QR

- Represent each sentence as vector of content words
- Weight each vector by the sentence's position in its document
- Choose largest vector
- Modify remaining vectors so they are orthogonal to the chosen vector

Content Selection - CLASSY

- Hidden Markov Model trained on features of summary sentences of training data
- Use HMM to compute forward-backward weights for each sentence in test data
- Select sentences with highest weights (6 times the number needed)
- QR Matrix Decomposition used to make 100 word summary from candidates
- Updates included modifications to feature compression by linear transformations/kernels; this still did not impact ROUGE scores

Information Ordering

- Used chronological expert, succession expert, precedence expert from Bollegala et. al. (2012)

All Results

devtest 2010	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Random	0.14955	0.02486	0.00453	0.00085
LLR	0.28810	0.08618	0.03132	0.01206
QR	***0.26193***	***0.07238***	***0.02276***	***0.00722***
HMM+QR	0.23568	0.06242	0.01883	0.00546

evaltest 2011	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Random	0.15862	0.02885	0.00814	0.00388
LLR	0.31978	0.10319	0.04335	0.02203
QR	***0.31851***	***0.10042***	***0.03994***	***0.01852***
HMM+QR	0.28405	0.08206	0.02852	0.01188

QR Results Over Time

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
D2	0.23280	0.05685	0.01540	0.00380
D3	0.26771	0.07797	0.02725	0.00959
D4.devtest	0.26193	0.07238	0.02276	0.00722
D4.evaltest	**0.31851**	**0.10042**	**0.03994**	**0.01852**

Observations

- LLR system has highest ROUGE scores
- Manually, we felt the QR summaries were easier to read, more on topic, and had better information ordering
- QR is biased toward longer sentences – means summaries can be well under 100 words, resulting in lower ROUGE scores

Error Analysis – Compression

- CLASSY compression techniques are very conservative, so errors are minimized
- We neglected to correct “a” vs. “an” when intervening content removed
- Some punctuation and case issues persist but they are very rare
- More could be compressed – instead of adhering to CLASSY further compression could have resulted in more information in the summary

Error Analysis – LLR and QR

- LLR and QR both generally output readable summaries with important information
- QR has an appropriate lead sentence and very articulate albeit compressed sentences
- LLR occasionally returns choppy sentences
- Since LLR summaries have more sentences, more potential for poor information ordering

Error Analysis - CLASSY

- CLASSY summaries do not contain the most important information
- The 2009 training data has no start summaries which severely impacts the Markov assumption
- The Chi-square distribution compression muddles the feature vector values – linear transformations fix this problem but not enough to compensate for the training problem

All Results

devtest 2010	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Random	0.14955	0.02486	0.00453	0.00085
LLR	0.28810	0.08618	0.03132	0.01206
QR	***0.26193***	***0.07238***	***0.02276***	***0.00722***
HMM+QR	0.23568	0.06242	0.01883	0.00546

evaltest 2011	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Random	0.15862	0.02885	0.00814	0.00388
LLR	0.31978	0.10319	0.04335	0.02203
QR	***0.31851***	***0.10042***	***0.03994***	***0.01852***
HMM+QR	0.28405	0.08206	0.02852	0.01188

Final Summarization System

Ruth Morrison
Florian Braun
Andrew Baer

Contents

- System Overview
- Approaches
  - Preprocessing
  - Centroid Creation
  - Sentence Extraction
  - Sentence Ordering
  - Realization
- Results and Discussion
- Conclusion

Overview: Influences

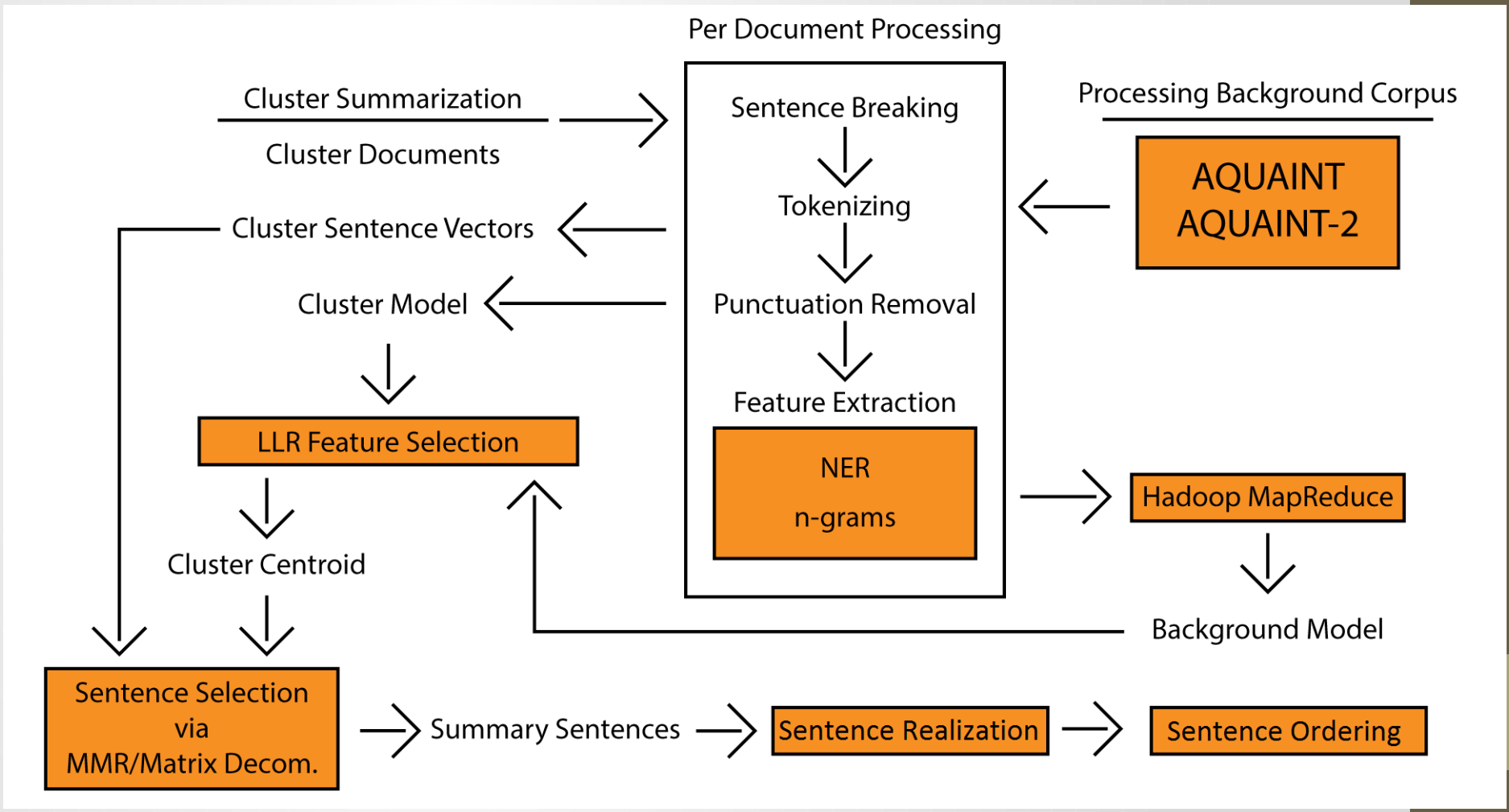
- MEAD (Radev et. Al., 2000)
  - Centroid based model
  - Some scoring measures for use in extracted summaries
- CLASSY (Conroy et al., 2004)
  - Log Likelihood Ratio (LLR) to detect features in the cluster when compared to the background corpus
  - Initially reduced redundancy using matrix reduction.
- Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: An entity-based approach. In Proceedings of the 43rd Annual Meeting of the ACL, pages 141–148. Association for Computational Linguistics.
  - Inspired the information ordering approach
- Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and

  producing summaries

Overview: Corpus usage

- Model:
  - AQUAINT and AQUAINT2
- Document clusters:
  - AQUAINT and AQUAINT2
- The clusters of documents to be classified are generally 6-10 articles, while the two corpora are around 2 million articles.
- Because of this, we believe that pulling our model and articles from the same corpora will not negatively impact the results.

System Overview



Approach: Model creation

- Background processing for LLR calculation
- Sentence breaking
- Feature vectors
  - Unigrams, trigrams, and named entities.
  - Punctutation removal, stopword removal, and lowercasing were done for the creation of n-gram features.
- NLTK was used for sentence breaking, tokenization, NER and stopword removal.
- The NLTK NE Chunker does a poor job of categorizing the types of names, so we kept it in binary mode.
- Feature types are kept separate to maintain the probability space.
  - Each are kept as their own model, enabling us to load any combination of features we want into the summarizer.

Approach: Centroid Creation

- Similar preprocessing: sentence breaking and vectorization
- Feature counts are stored to compute LLR and then binarized.
- Calculate LLR of all features of a given type.
  - Any feature above a threshold (10.0 for us) is weighted as 1, and any feature below is weighted as 0.
  - Allows retention of features on a per type basis.
  - More favorable approach than simply Top N features from all type by LLR value.
  - A variable number of active features could capture differences in in topic signature that may not be captured when every cluster centroid is kept to an arbitrary number of non-zero weighted features.

Approach: Sentence Extraction

- Create a sentence score based on three components:
  - Cosine similarity between sentence and centroid
  - The position of the sentence within the document it occurs in
    - Organized in decreasing value from 1 to 0, with the first sentence having one and the last having 0.
  - Overlap between the score of the first sentence and the current sentence
    - Dot product of the sentence vectors
    - If there's a headline it is treated as the first sentence

Approach: Sentence Extraction

- Each subscore is weighted and added together for the total score.
- Makes the first sentence the highest scoring sentence
- To create a more query oriented summary we changed from first sentence overlap to Topic overlap
- We began by using the weights from the technical paper: 3 for similarity, 2 for the position, and 1 for the overlap.
  - Reweighting improved the ROUGE score and gave more coherent summaries when judged by a human.
  - Went to a 2, 1, 3 weighting

Approach: Sentence Extraction

- An attempt was made at a more sophisticated scoring method
  - Parse model summaries and article sentences with the NLTK-based Stanford Parser
  - Give them a scaled score based on the sum of the logprobs in the set of probabilities generated from the models
  - The score is weighted and added to scores for centrality/position/topic overalap.
- Took far too long
- Many OOV issues due to small relative size of the training data.

Approach: Redundancy Reduction

- Originally used Matrix Decomposition
- Got better results with MMR and changed to that for D3.
  - MMR applies a penalty to the score of a sentence as factor of it and the most similar sentence to it that is already in the summary.
  - In our use we chose the highest scoring sentence to be the first sentence of the summary

$$R = Ranked\ list\ of\ documents$$

$$Q = Query$$

$$S = Subset\ of\ R\ of\ already\ selected\ Documents$$

$$MMR \stackrel{def}{=} \underset{D_i \in R-S}{arg\ max}(\lambda(sim(D_i, Q) - (1-\lambda) \underset{D_j \in S}{max}\ sim(D_i, D_j)))$$

Approach: Redundancy Reduction

- Reasoning for switching to MMR:
  - MMR lead to an improvement in ROUGE scores when combined with unigram features.
  - Matrix reduction had the best scores when using trigram features, but it still scored worse then MMR on unigrams.
  - Once we switched to reweighted scoring we got better results from unigrams + NER

Approach: Information Ordering

- Inspired by entity coherence based ordering
  - Barzilay and Lapata, 2005
- Simplified version with no SVM learning
- Integrated with chronological ordering
- Originally planned to find coreference chains referring to the same reference in different documents in the same cluster, and go with the idea that sentences that contained many references to the same thing would be more likely to occur near each other.
  - Used named entities to identify entities instead, because we already had NER implemented.

Approach: Information Ordering

- Current algorithm picks the first sentence of the first document as the first sentence of the summary.
- Subsequent sentences are given a score equal to the reverse index of the sentence when all were ordered chronologically added to twice the number of shared entities between the sentence and the last sentence added to the summary
  - ‘reverse index’ example: with four sentences, the chronologically first scores 3, the next one 2, and so on.
- The sentence with the greatest score is added to the summary.
- The process repeats until there is only one sentence left to add.

Approach: Realization

- Focuses on compressing single sentences by removing extraneous information.
  - Sentence level compression without regard for the other sentences in the document from which a sentence stems or the summary it's in.
- Independent module within the pipeline so it could be done at different points to observe how it affected the summaries.
  - Locations attempted: before centroid creation, after centroid creation but before content selection, and after content selection.
  - Compression ended up being a final post-processing step.

Approach: Realization

- Three main components:
  - Remove any text spans enclosed in parenthesis.
    - Ex: “We focused on automatic summarization (similar to the TAC shared task) for 573” would become “we focused on automatic summarization for 573”.
  - Remove temporal expressions based on relative time (tomorrow, yesterday), days of the week without dates, months without a year, and similar cases.
  - Remove adjectives when they reference nouns.
    - This is retained despite being uncertain about its benefits.
    - Occasionally makes things nonsensical
      - “Saudi Arabia rejected a demand for an investigation into the assassination of Lebanese Prime Minister Rafik Hariri, saying Lebanon is a country.”

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Unigrams + Ne	**0.27637**	**0.07983**	**0.02795**	**0.01111**
Unigrams	0.26995	0.07525	0.02480	0.00882
Trigrams + Unigrams + Ne	0.26059	0.07319	0.02600	0.00974
Trigrams + Unigrams	0.25469	0.07312	0.02599	0.00964
Trigrams + Ne	0.22675	0.06501	0.02297	0.00932
Ne	0.23557	0.06463	0.02348	0.00849
Trigrams	0.22392	0.06373	0.02252	0.00925

Table 4: Results from the 2010 Data Using MMR and Re-weighting

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Unigrams + NE	**0.26730**	**0.07125**	**0.02296**	**0.00771**
Unigrams	0.26196	0.06794	0.02173	0.00732
Trigrams + Unigrams + NE	0.25267	0.06591	0.02233	0.00854
Trigrams + Unigrams	0.24307	0.06341	0.02176	0.00775
NE	0.23383	0.06235	0.02233	0.00820
Trigrams + NE	0.22096	0.06023	0.02067	0.00851
Trigrams	0.21185	0.05512	0.01847	0.00720

Table 5: Results from the 2010 Data Using MMR, Re-weighting, Compression

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Unigrams	0.32343	0.10054	0.03953	0.02016
Unigrams + NE	**0.31933**	**0.09669**	**0.03691**	**0.01895**
Trigrams + Unigrams	0.29895	0.08695	0.03274	0.01679
Trigrams + Unigrams + NE	0.29565	0.08567	0.03150	0.01547
Trigrams	0.26349	0.07807	0.03234	0.01722
Trigrams + NE	0.26646	0.07645	0.02995	0.01564
NE	0.25005	0.06507	0.02324	0.01138

Table 6: Results from the 2011 Data Using MMR and Re-weighting

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Unigrams	0.30631	0.08766	0.03283	0.01601
Unigrams + NE	**0.30240**	**0.08593**	**0.03178**	**0.01517**
Trigrams + Unigrams	0.28896	0.07773	0.02707	0.01328
Trigrams + Unigrams + NE	0.28317	0.07535	0.02657	0.01308
Trigrams	0.24834	0.06833	0.02610	0.01228
Trigrams + NE	0.25283	0.06783	0.02519	0.01211
NE	0.24297	0.05887	0.01914	0.00870

Table 7: Results from the 2011 Data Using MMR, Re-weighting, Compression

Results

- The average human results were as follows, based on a 1-5 scale:
  - Sichuan Earthquake: 2
  - Korean oil spill: 2
  - Cough Syrup Infants: 3.3
  - I-35 Collapse: 2
  - Sleep Deficit: 3
  - Java Ferry Disaster: 3.3
  - Glasgow Airport Bomb: 2
  - Simpson Armed Robbery: 3.3
- Average human score of2.625

Improved extraction and Text realization

Ahmed Aly, Abdelrahman Baligh, Veljko Miljanic

Overview

System architecture overview

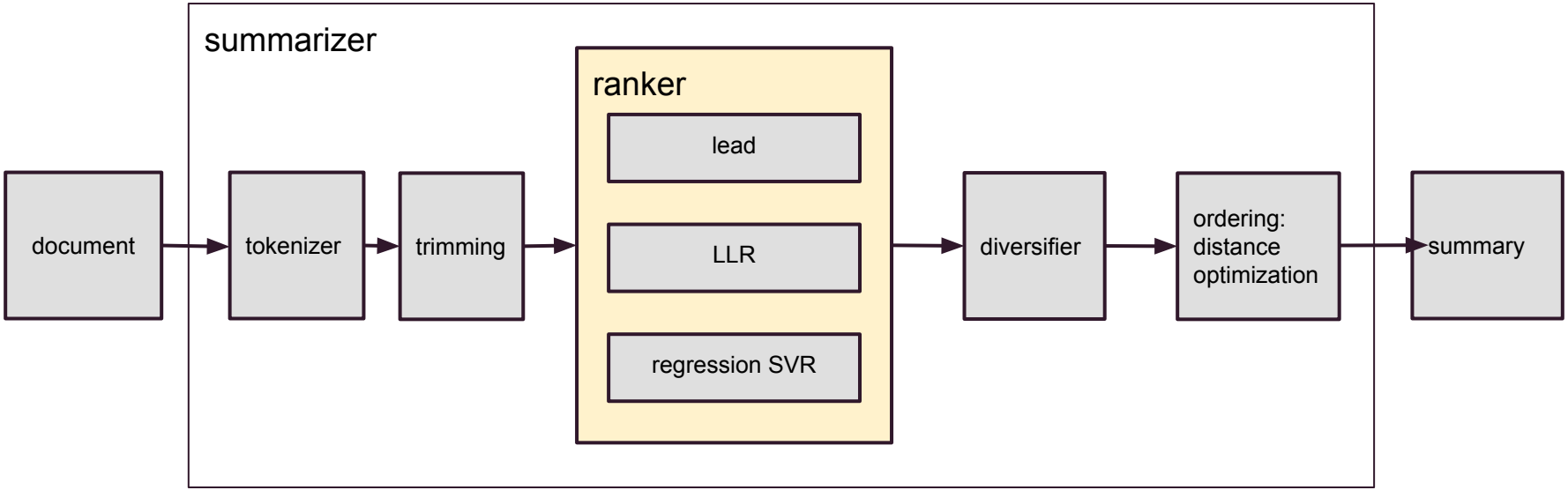
Extraction improvement experiments

- Information-Content Based Sentence Extraction
- ranker ensemble
- change SVR target

Text realization

- sentence trimming (CLASSY 2006)
- trimming order

System architecture overview



System architecture overview

Content extraction

- Our approach is to solve content extraction as sentence ranking problem
- We want to build ML based ranker that could combine many features to rank sentences
- Baseline systems are Lead and LLR

Ordering

- Maximizing COS similarity between adjacent sentences (TSP)

Text realization

- We trim the sentences as a first step, we apply trimming similar to CLASSY

Regression ranker

- Features:
  - f1: LLR
  - Paragraph: f2:paragraph number
  - Sentence: f3: sentence length, f4: quotation
  - Document: f5:sentence position
- Outputs:
  - Sentence ROUGE-1 F score / R score

LLR Improvements

- Information-Content Based Sentence Extraction
  - Treat the whole unigrams in the document as query terms
  - Repeat the following:
    - Score all sentence according to similarity to the query
    - Pick the sentence with the top score
    - Remove all the unigrams that are covered by the picked sentence from the query
- Results weren't good:
  - 1 ROUGE-1 Average_R: 0.19026 (95%-conf.int. 0.16888 - 0.21171)
  - 1 ROUGE-2 Average_R: 0.04713 (95%-conf.int. 0.03703 - 0.05810)
  - 1 ROUGE-3 Average_R: 0.01467 (95%-conf.int. 0.00998 - 0.02030)

LLR Improvements

- Classifier ensemble
  - If the different classifiers were good at a different set of problems, then perhaps combining them linearly will give an output that is better than the sum of its parts
  - new score = alpha_1 * classifier_1 + alpha_2 * classifier_2 + alpha_3 * classifier_3, where alpha_1 + alpha_2 + alpha_3 = 1
  - it did not.
  - results were always lower than the best ranker.

LLR Improvements

- Other improvements
  - Changed SVR target to be ROUGE-1 F-Score instead of ROUGE-L
  - 1 extra feature: if the end of the sentence has a punctuation mark in the end.
- Final Results:

1 ROUGE-1 Average_R: 0.31628 (95%-conf.int. 0.29445 - 0.33809)

1 ROUGE-2 Average_R: 0.09503 (95%-conf.int. 0.07897 - 0.11080)

1 ROUGE-3 Average_R: 0.03416 (95%-conf.int. 0.02530 - 0.04370)

1 ROUGE-4 Average_R: 0.01635 (95%-conf.int. 0.01093 - 0.02246)

Sentence diversification (no changes)

- Since we're maximizing expectation for ROUGE score, we need to account for shared information between selected sentences.
- We penalize each sentence for redundant information with what's already selected
- As long as we have place in the summary:
  - Take the top sentence
  - For all remaining sentences penalize shared n-grams with selected summary
  - Repeat

Sentence diversification (no changes)

- N–gram penalization:

$$\sum_{n=1}^{4}[1 - (\sum redundant\ n.grams\ / \sum total\ n.grams\ in\ sentence)^{1/\alpha_n}]$$

- Where $\alpha 1$, $\alpha 2$, $\alpha 3$, $\alpha 4$ are the penalty weights for unigrams, bigrams, trigrams, quadgrams respectively.
- Our experiments suggest that the optimum values for alphas is 0.25 each.

Ordering (no changes)

Similar to CLASSY 2006:

- Find order that maximizes sum of COS similarities (tf-idf)

Optimization algorithm

1. start with rank order
2. for each sentence i
    a. for each sentence k
        i. swap sentences i and k if it improves the score
3. if score was improved in last iteration
    a. goto 2.
4. done

Text realization

- Trimming is applied as a first summarization step (before selection)
- Algorithm:
  - We use (Stanford) POS and NER to tag input
  - Remove LOCATION, DATE (AGENCY) patterns in fist document sentence: "LARAMIE, Wyo.(AP) -- Police stepped up patrols today ..."
  - Remove "... said" suffixes: "... but spent no time behind bars, Frank said."
  - Remove "..., he said, ..." in midle of sentence: "For example, she said, it could be..."
  - Remove words in brackets:
  - Remove sentence with < 70% of lowercase and sentences shorter than 3 words
- Trimming slightly improved ROUGE1: 0.88%

References

- Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. University of Southern California.
- Kai Hong et al. 2014. Improving the Estimation of Word Importance for News Multi-Document Summarization. University of Pennsylvania.
- Daniel Mallett et al. 2004. Information-Content Based Sentence Extraction for Text Summarization. Department of Computing Science, University of Alberta, Canada.
- John M. Conroy et al. 2006. CLASSY Query-Based Multi-Document Summarization. IDA/Center for Computing Sciences.

The End

Summarization Task

Tristan Chong
Syed Sameer Arshad

For Content Realization we improved three things

- Content Selection
  - Choose better sentences from the source documents.
- Compression
  - Remove unwanted material from chosen sentences.
- Readability
  - Make sure that the final summary flows smoothly.

Content Selection

- Reduced the number of sentences considered for the summary by omitting the following:
  - Any sentence that contained quoted speech.
  - Any sentence that had a question mark.
- Improved granularity of source sentences by chopping all semi-colon sentences into two along the semi-colon boundary.
  - This way, shorter useful content that exists on only one side of a semicolon can be taken instead of a longer sentence.

Compression

- We removed subordinate clauses from chosen sentences:
  - We did this by using a regular expression to detect when more than one word existed between commas.
  - We only did this for the first subordinate clause we found in a sentence using this approach.
    - We wanted to avoid being too destructive.

Compression

- We also attempted to remove unwanted strings of adjectives and adverbs.
  - Motivation: “Tristan found a big, green, slimy, dirty, creepy, smelly slug today.” would be better summarized as “Tristan found a big slug today”.
  - This involved POS tagging and detecting ADJP and ADVP.
  - Very challenging…
  - Results were found to be problematic
  - Ended up being too destructive.
    - Example: “Computerized high schools are a new concept these days” becomes “High schools are a new concept these days”.

Compression

- We also attempted to try and remove temporal expressions.
  - Example: “On the 30th of June in the summer of 2003, Sameer graduated from high school.” should become “Sameer graduated from high school.”
  - We noticed that this too was problematic, because it ended up removing names of people like “June”. It also broke sentences into incoherent chunks: “April showers bring May flowers” would become “Showers bring flowers”.

Readability

- We made the following attempts to improve readability:
  - Removed unnecessary whitespace left behind from deletions and source-material errors.
  - Capitalized the first letter of all sentences.
  - Ensured that the sentence ordering in the summary was partially influenced by the sentence ordering of the source documents.
  - The remaining influence on sentence ordering came from trying to match the thematic patterns of when a theme appeared in a source document.

Deeper processing

2 related proposed tasks:

- Task A: Detect unnecessary phrases by removing all words with POS tags associated to adjectives and adverbs (RB, RBR, RBS, JJ, JJR, JJS) followed by a noun. Example: "The quick brown fox" becomes "The fox" while "The dirty creepy fox was quick and brown" becomes "The fox was quick and brown"

- Task B: Kill all subordinate clauses recognized by a parser. This would have involved finding the right node in a tree and killing it

- These ideas sounded good in theory, but were ineffective in practice

Deeper processing

Decided to use the Stanford parser during content realization (fewer sentences to deal with)

- This was still too slow to be viable
- Removing entire phrases often resulted in nonsense

Explored other parsers included in NLTK:

- Chart parser
- Recursive descent parser
- Shift reduce parser
- Left corner chart parser
- Bllip parser (not installed)
- Malt parser (not installed)

These parsers required context-free grammars

- Generated CFGs with the Stanford parser output
- Does not work for previously-unseen data

Experimenting with MEAD and Topic Orientation Weights

- The variation in this optimization task was negligible.
- Our weights happened to be pretty good to begin with.
- We left them the same.
- Centroid Weight = 3
- Position Weight = 2
- First Sentence Similarity Weight = 1
- Topic Orientation Weight = 1.8

High Ranked Summary

A crane-carrying vessel slammed into a Hong Kong-registered oil tanker in seas off South Korea's west coast Friday causing the tanker to leak about 110,000 barrels of crude oil, the Maritime and Fisheries Ministry said. The Coast Guard dispatched 34 coast guard, navy and other government vessels as well as four helicopters to stop the spread of the spill and collect the oil, Jung said. The spill was believed to be South Korea's largest, according to the Coast Guard and the ministry. A spill in 1995, previously considered the largest, involved a leak of about 5,035 tons -- about 35,000 barrels -- of crude and fuel oil.

Another High-Ranked Summary

Major makers of over-the-counter infant cough and cold medicines announced Thursday that they were withdrawing their products from the market for fear that they could be misused by parents. The U.S. government is warning parents not to give cough and cold medicines to children under 2 without a doctor's order, part of an overall review of the products' safety and effectiveness for youngsters. The move comes two weeks after safety reviewers within the Food and Drug Administration urged the agency to consider an outright ban of over-the-counter cough and cold products for children under the age of 6. Even the industry's own trade association, the Consumer Healthcare Products Association, recommended two weeks ago that the products should no longer be used for infants. Next week, a committee of outside experts will meet to consider the safety of these medicines and offer recommendations to the agency.

Final Results

Measure	Test Set	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Precision	DevTest	0.22458	0.04878	0.01461	0.00296
	EvalTest	0.28066	0.06852	0.02396	0.01119
Recall	DevTest	0.20274	0.04450	0.01333	0.00317
	EvalTest	0.24961	0.06234	0.02234	0.01054
F-Measure	DevTest	0.21263	0.04644	0.01391	0.00306
	EvalTest	0.26374	0.6516	0.02308	0.01083

Average Readability Score
3.333 / 5

Thank you!

Summing Up

Ling 573
Systems and Applications
June 4, 2015

Results in Context: TAC 2010 (dev)

- 0.09574
- 0.09418
- ....
- CDS: 0.09312
- GMW: 0.08678
- BBM: 0.07983
- A: 0.0723*
- B: 0.07144
- C: 0.06784
- D: 0.06730
- E: 0.06706
- MEAD: 0.05927
- .....
- LEAD: 0.05376
- 9 systems
- 0.00506

Results in Context: TAC 2011 (eval)

- 0.13440
- 0.12994
- ....
- CDS: 0.10569
- HLM: 0.10042*
- GMW: 0.09761
- A: 0.09669
- B: 0.09503
- C: 0.09399
- D: 0.07020
- MEAD: 0.0862
- Other systems....
- E: 0.06851
- LEAD: 0.0641
- F: 0.04718
- 3 systems
- 2 systems score: 0.0 R2

	1110	1115	1119	1122	1127	1130	1133	1143	Avg
HLM	**4.0**	4.0	3.3	**3.7**	3.3	3.7	3.3	3.7	3.625
CDS	3.0	4.0	1.7	3.3	3.0	4.0	**4.3**	**5.0**	3.542
GMW	3.0	4.3	2.3	3.3	**3.7**	**5.0**	3.3	2.0	3.375
A	3.7	**4.7**	**4.0**	2.3	3.0	2.7	2.7	3.7	3.333
B	3.7	3.3	1.7	3.3	2.7	3.0	3.7	2.7	3.0
C	1.7	3.3	2.3	2.3	3.3	4.7	3.0	2.3	2.875
D	3.3	3.3	2.3	3.3	1.7	2.7	2.0	4.3	2.875
E	2.0	2.0	3.3	2.0	3.0	3.3	2.0	3.3	2.625
F	3.3	2.3	2.7	2.3	1.3	1.3	4.0	3.7	2.625
G	2.7	2.3	1.7	2.3	2.3	2.0	2.7	3.3	2.416
	3.033	3.367	2.533	2.833	2.733	3.233	3.1	3.4	

Readability

Team	Average Readabilty (over 8 topics)
TAC 2011 best	3.75
Heath, Lockwood, Marsh	3.625
Cooper, Dai, Shintani	3.542
Garnick, McCranie, Whelan	3.375
A	3.333
LEAD	3.12
B	3.0
C	2.875
D	2.875
E	2.625
F	2.625
MEAD	2.5
G	2.475

Overview

- Great work!
- Lots of different approaches
- Tradeoffs between:
  - Components, combining, tuning
  - ROUGE, readability

Last Notes

- Thank you all!

- Team evaluations:
  - Linked from course webpage
  - REQUIRED

- Course evaluations
  - Linked from course webpage, email