D4: Final Summary

Selection, Ordering, and Realization

Brandon Gahler
Mike Roylance
Thomas Marsh

Architecture: Technologies

Python 2.7.9 for all coding tasks

NLTK for tokenization, chunking and sentence segmentation.

pyrouge for evaluation

textrazor for entity extraction

attensity for entity and semantic information extraction

Stanford Parser for sentence compression

svmlight for training our ranking classifier

Architecture: Implementation

Reader - Extracts data from topic-focused document clusters

Document And Entity Cache - Entities, Sentences, Semantic Information

Extraction Clusterer - Ranks best sentences for output

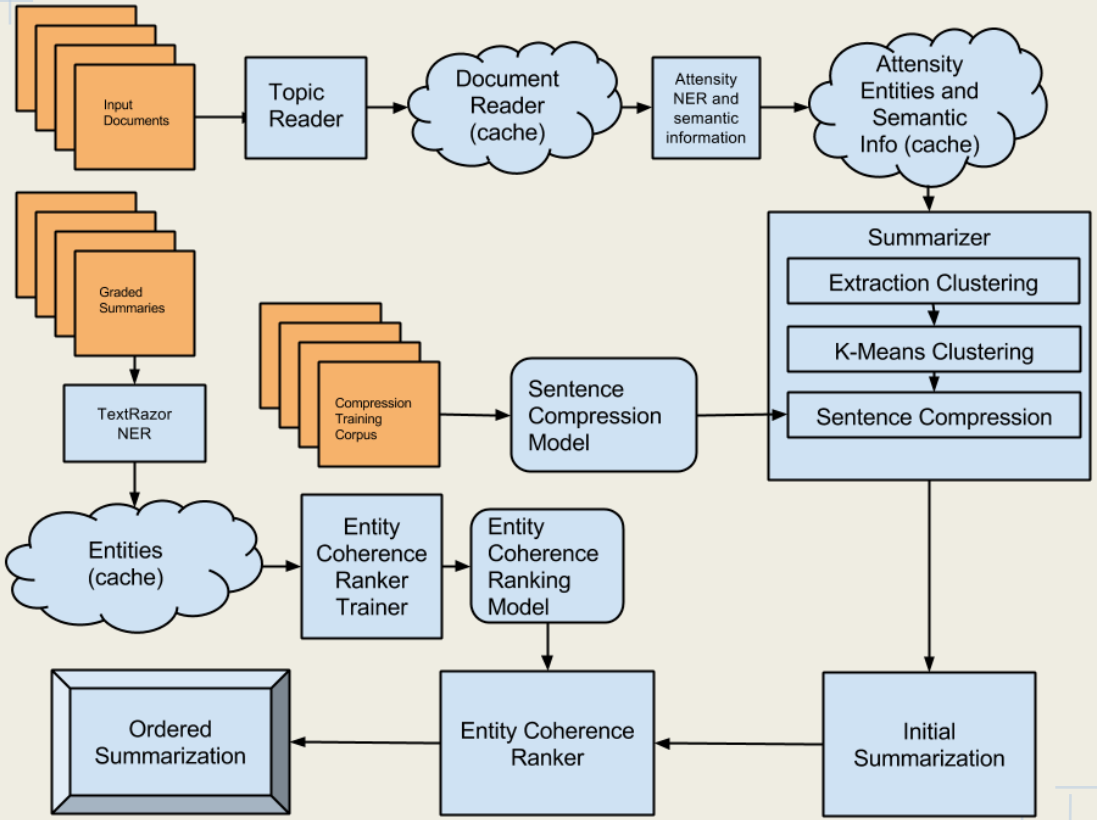
K-Means Clustering - Redundancy Reduction

Compressor - Compresses top sentences inline

Reorderer - Uses entity-coherence ranking to reorder

Evaluator - Uses pyrouge to call ROUGE-1.5.5.pl

Architecture: Block Diagram



Summarizer

Disabling Summary Technique Weighting/Voting Strategy:

Though we have a strong intuition that our technique weighting/voting scheme would eventually bear fruit, we continued to see little evidence for this. The empirical weight generator always appeared to select a single technique at 1.0 and others at 0.0. Because of this, we disabled this mechanism for this deliverable to reduce complexity. We were very sad about this, and hope to resurrect it in the future when we have time to examine what we may have done wrong.

We used the **Extraction Clustering** technique for our single selection strategy.

Extraction Clustering

- Different extractions used for comparison
- **Entity** (Named Entity Recognition)
  - Semantic information
    - Text
    - Domain Role (person, location etc)
- **Triple**
  - Subject, Predicate, Object
- **Fact**
  - Case frame building blocks
  - Element and mode
- **Keyword**
  - Root and POS

Extraction Clustering

Algorithm Enhancements:
We made several incremental refinements to our Extraction Clustering technique for this iteration:

- Normal “most important sentence(s)” extraction with score
- Added a new layer, K-Means Clustering to reduce redundancy.
  - Tried from 20-30 clusters
  - Shot for an average of 30-50 “points” per cluster (minimum of 1)
  - Forced to pick 1 sentence from each cluster.
  - Picked the top scored sentences (from Extraction Clustering)
- Explored root bigrams (word and noun) -
  - I loved to visit Essex. (loved->love) (morphology)
  - (I/PRONOUN, love/VERB), (love/VERB, to/INFINITIVE_TO)
  - (to/INFINITIVE_TO, visit/VERB), (visit/VERB, Essex/NOUN)

Extraction Clustering

Peripheral Enhancements:
We also made some peripheral enhancements to help our overall selection performance:

- Fixed a bug where our sentences were a bit too long, causing our reordering mechanism to actually be doing selection, and thereby changing our rouge scores.
- Removed all sentences with quotes. A pox on quotes. Forever. Amen.
- Finally removed those pesky info media headers once and for all with some awesome regular expression fu.
- Removed all sentences which did not have a verb.
- Normalized for sentence length to “other” compared sentence length

Sentence Compression

Overall Strategy:
Keep/delete sequence labeling with linear-chain CRF

- Linear SVM
- Written News Compression Corpus
- Features:
  - Current word features + 2 previous
  - Feature selection: top 10% chi-squared
  - Word level features
    - within X of start/end of sentence
    - capitalization
    - negation/punctuation/stopword
    - in upper X% of tfidf relative to rest of the sentence
    - stem and suffix

Sentence Compression

- Features:
  - Syntax features
    - Tree depth
    - Within a X phrase
    - 2 immediate parents
    - X from the left within parent phrase
  - Dependency features
    - Dependency tree depth
    - Mother/daughter of a X dependency
- Just before sentences are added to initial summaries (before ordering) we run the sentence through the compressor and output the compressed sentence instead.

Sentence Compression

Results

- 79.4% accuracy w/ word features, 82.7% with syntax and dependency
- Tendency to remove entire sections, rather than individual superfluous words
  - A co-defendant in the O.J. Simpson armed robbery case told a judge Monday he would plead guilty to a felony and testify against Simpson and four others in the hotel room theft of sports collectibles from two memorabilia dealers.
  - If it were fully loaded, the ship's deck would be lower to the water, making it easier for pirates to climb aboard with grappling equipment and ladders, as they do in most hijackings.
- No rouge score improvement
- Not used in final version

Sentence Ordering

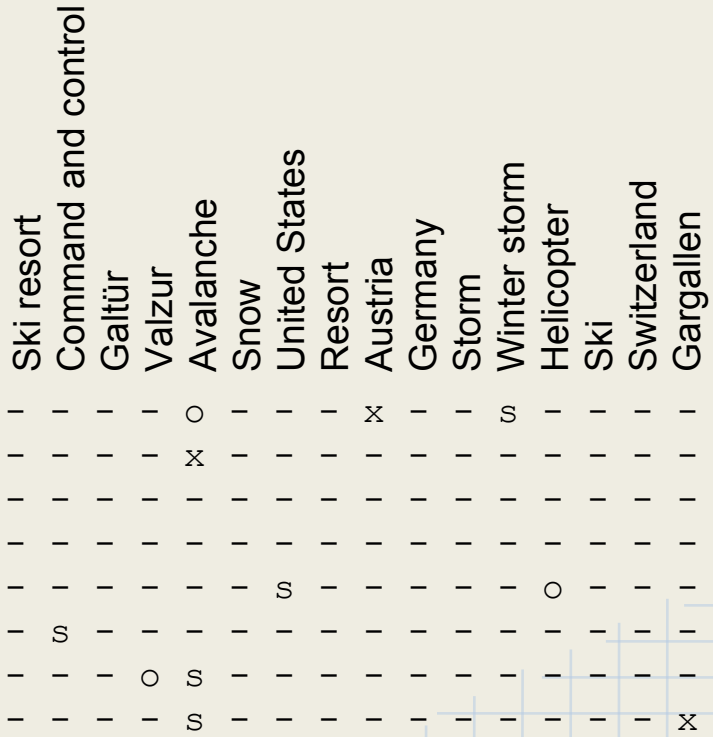
Entity-Based Coherence solution similar to Barzilay and Lapata (2005).

- **NER**: We used a named entity recognizer to extract entities to use in the transition grids.
  - Entities were originally extracted via TextRazor https://www.textrazor.com/

Entity Coherence

Improvements:

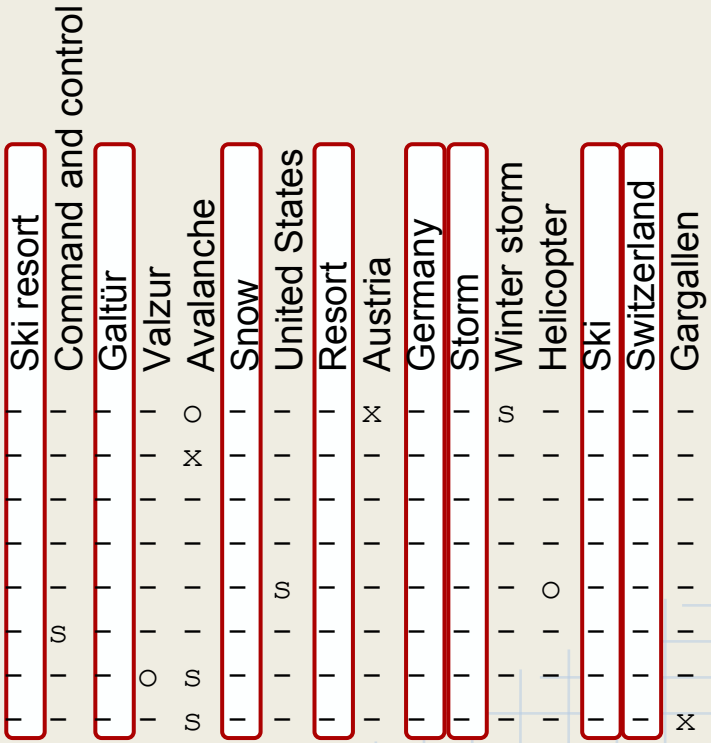
1. Removed unused entities from transition graph
2. Added Tuning Parameter for entity frequency
3. Trained on graded summaries
4. Greatly improved performance



Entity Coherence

Improvements:

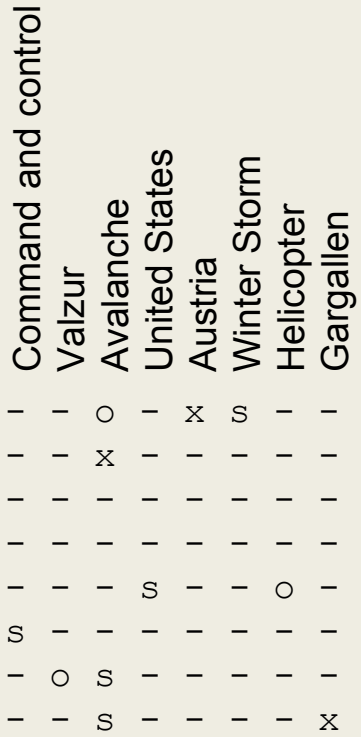
1. **Removed unused entities from transition graph**
2. Added Tuning Parameter for entity frequency
3. Trained on graded summaries
4. Greatly improved performance



Entity Coherence

Improvements:

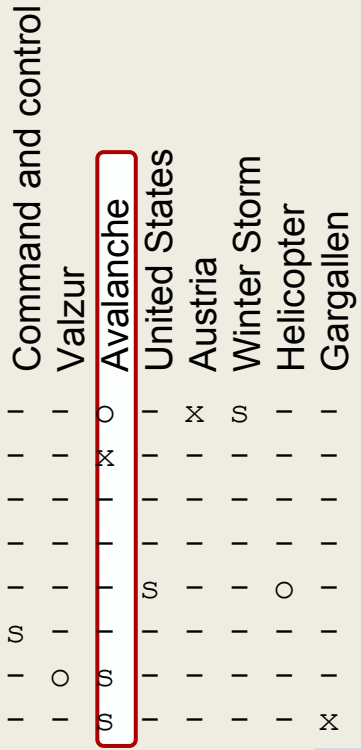
1. **Removed unused entities from transition graph**
2. Added Tuning Parameter for entity frequency
3. Trained on graded summaries
4. Greatly improved performance



Entity Coherence

Improvements:

1. Removed unused entities from transition graph
2. **Added Tuning Parameter for entity frequency**
3. Trained on graded summaries
4. Greatly improved performance



Entity Coherence

Improvements:

1. Removed unused entities from transition graph
2. **Added Tuning Parameter for entity frequency**
3. Trained on graded summaries
4. Greatly improved performance

Avalanche

Final Results

Average ROUGE scores for the Devtest Data:

ROUGE Technique	Recall	Precision	F-Score
ROUGE1	0.23577	0.29921	0.26186
ROUGE2	0.07144	0.09095	0.07949
ROUGE3	0.02821	0.03621	0.03151
ROUGE4	0.01271	0.01624	0.01419

Final Results

Average ROUGE scores for the Evaltest Data:

ROUGE Technique	Recall	Precision	F-Score
ROUGE1	0.26140	0.27432	0.26699
ROUGE2	0.06851	0.07162	0.06984
ROUGE3	0.02268	0.02342	0.02298
ROUGE4	0.00950	0.00976	0.00960

Final Results

Change in Average ROUGE scores From D3 to D4 for DevTest Data:

ROUGE Technique	Recall		Precision		F-Score	
ROUGE1	0.23577	-0.02%	0.29921	+21.02%	0.26186	+8.79%
ROUGE2	0.07144	+14.21%	0.09095	+41.07%	0.07949	+25.44%
ROUGE3	0.02821	+41.40%	0.03621	+76.81%	0.03151	+56.14%
ROUGE4	0.01271	+92.87%	0.01624	+141.67%	0.01419	+113.70%

Final Results

Apples to Oranges: D3 Devtest results compared to D4 Evaltest results:

ROUGE Technique	Recall		Precision		F-Score	
ROUGE1	0.26140	+10.85%	0.27432	+10.95%	0.26699	+10.92%
ROUGE2	0.06851	+9.53%	0.07162	+11.09%	0.06984	+10.21%
ROUGE3	0.02268	+13.68%	0.02342	+14.36%	0.02298	+13.88%
ROUGE4	0.00950	+44.16%	0.00976	+45.24%	0.00960	+44.58%

Future Work

1. **Add Coreference Resolution to Entity Coherence:** This is next! We have coref resolution in the project, we just haven't hooked it up to the Entity Coherence feature.

2. **Reenable voting-based technique aggregation** and run machine-learning algorithms to generate the best weights.

3. **Fix some bugs we found.** we found some.

References

Heinzerling, B and Johannsen, A (2014). pyrouge (Version 0.1.2) [Software]. Available from https://github.com/noutenki/pyrouge

Lin, C (2004). ROUGE (Version 1.5.5) [Software]. Available from http://www.berouge.com/Pages/default.aspx

Roylance, M (2015). Attensity ASAS (Version 0.1) [Software]. Available from http://www.attensity.com

Crayston, T (2015). TextRazor (Version 1.0) [Software]. Available from https://www.textrazor.com/

Jaochims, T (2002a). SVMLight (Version 6.02) [Software]. Available from http://svmlight.joachims.org/

Barzilay, R., & Lapata, M. (2008). Modeling local coherence: An entity-based approach. Computational Linguistics, 34(1), 1-34.

Jurafsky, D., & Martin, J. H. (2009). Speech & language processing. Pearson Education India.

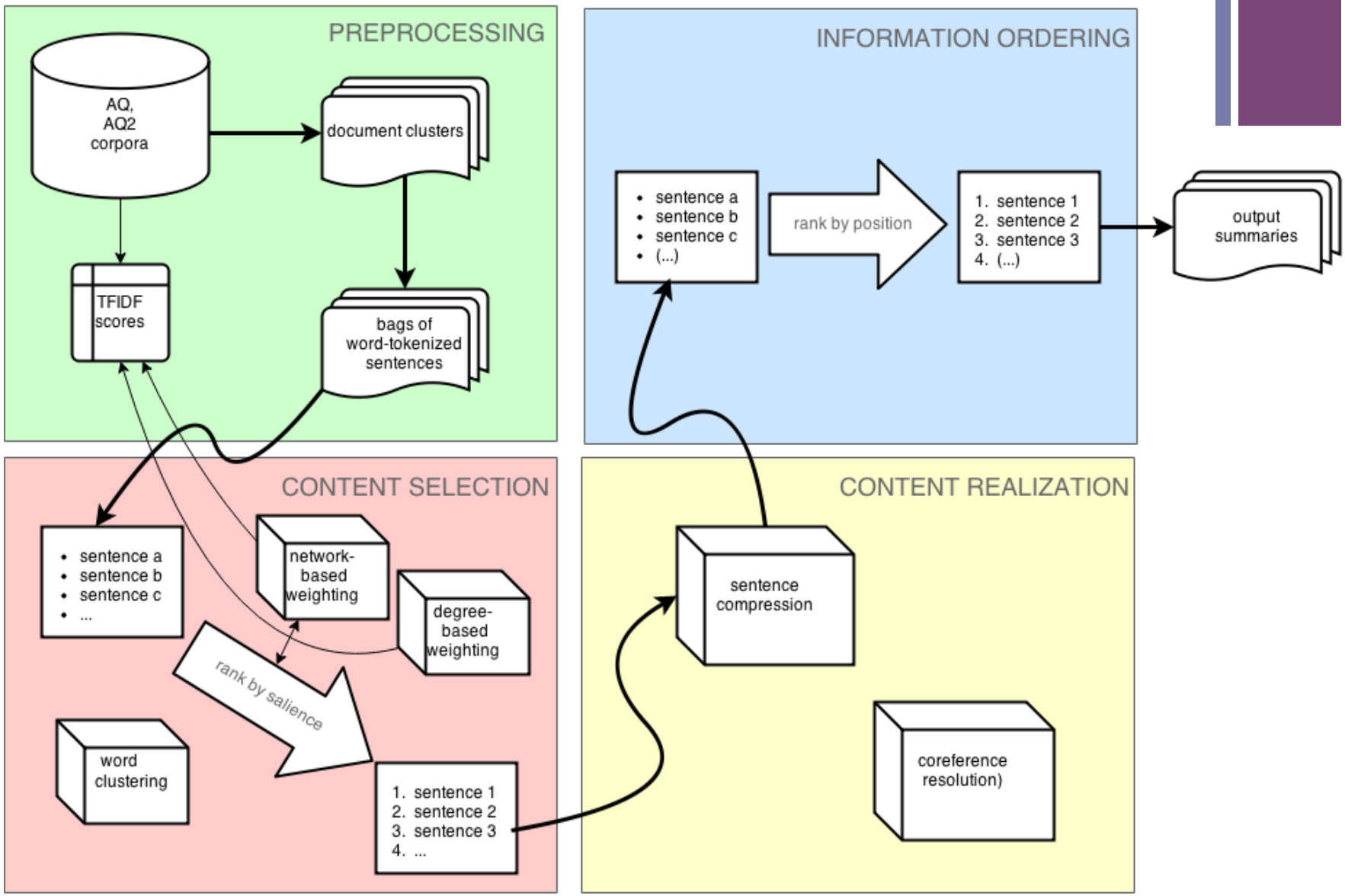
Radev, D, et al. (2006). MEAD (Version 3.12) [Software]. Available from http://www.summarization.com/mead/

P.A.N.D.A.S.

(Progressive Automatic Natural Document Abbreviation System)

Ceara Chewning, Rebecca Myhre, Katie Vedder

System Architecture



Changes From D4

- Cleaned up scores.
- Confirmed that coreference resolution, word clustering, and topic orientation did not improve results.
- Tried lowercasing, stemming, and stopping when calculating tfidf and comparing sentences.

Content Selection

Content selection

- Graph-based, lexical approach inspired by (Erkan and Radev, 2004).
- IDF-modified cosine similarity equation, using AQUAINT and AQUAINT-2 as a background corpus:

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

- Sentences ranked by degree of vertex.
- Redundancy accounted for with a second threshold.

Failed Attempts: Prestige-Based Node Weighting

- Tried to implement iterative method that weighted node scores based on prestige of adjacent nodes:

$$S_{new}(u) = \frac{d}{N} + (1-d) \sum_{v \in adj(u)} \frac{S_{old}(v)}{deg(v)}$$

- Didn't outperform naïve, degree-based node scoring.

Failed Attempts: Topic Orientation

- Generated larger set of topic words by including headlines of cluster's documents in the topic.
- Used Otterbacher et al.'s approach to include topic word overlap in LexRank-based scoring:

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

$$p(s|q) = d \frac{rel(s|q)}{\sum_{z \in C} rel(z|q)} + (1 - d) saliency_x$$

- A *d* value of *0.5* produced best results, but still did not improve ROUGE scores.

Failed Attempts: Word Sense Clustering

- Wanted to create clusters of words based on the words that co-occur with them in their context window, then use those clusters to have similar words count as one word when measure sentence similarity- i.e.
- Used Word2Vec to make the word vectors and calculate similarity, then sklearn.cluster's Kmeans to do unsupervised clustering over all the words in the document cluster. K = size of vocabulary/ 5
- When calculating new tfidf scores, replace words with their word cluster ID if it exists, and do the same for all documents as the background corpus.

Used this tutorial to lean Word2Vec and Kmeans:
https://www.kaggle.com/c/word2vec-nlp-tutorial/details/part-3-more-fun-with-word-vectors

Some Success: Lowercase, Stem, Stop

- We tried to lowercase, stem, and remove stopwords for all words when calculating tfidf scores, clustering words, and comparing sentences for content selection
- We used NLTK's English Lancaster stemmer and list of stopwords.
- This improved our ROUGE scores marginally, or did not, depending on what other features we had enabled.

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Without casing	**0.24756**	**0.06219**	0.02157	0.00861
With casing	0.24411	0.05755	0.01892	0.00771

Some Success: Query/Topic word weighting (headline)

d-value	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
0.1	0.24423	0.05824	0.01906	0.00794
0.3	0.24345	0.06012	0.02108	0.0082
0.5	**0.24756**	**0.06219**	0.02157	0.00861
0.7	0.24544	0.05918	0.0196	0.008
0.9	0.241	0.05798	0.01975	0.00772
1	0.24577	0.06054	0.02076	0.0084

+

Information Ordering

Information Ordering

Sentences are ordered by position of sentence within the original document:

$$pos(s) = \frac{I(\text{sentences in which } s \text{ occurs})}{C(\text{sentences in document})}$$

Information Ordering: A Cherry-Picked Example

BEFORE ORDERING

"Theo didn't want any police protection," of van Gogh in a telephone interview.

Van Gogh received many threats after the film was shown but always laughed them off.

The friends and family of Van Gogh had asked for people to make as much noise as possible in support of the freedom of speech.

Writer-director Theo van Gogh, a descendant of the artist Vincent van Gogh, was attacked shortly before 9 a.m. as he rode his bicycle through Amsterdam's tree-lined streets toward the offices of his production company.

AFTER ORDERING

Writer-director Theo van Gogh, a descendant of the artist Vincent van Gogh, was attacked shortly before 9 a.m. as he rode his bicycle through Amsterdam's tree-lined streets toward the offices of his production company.

The friends and family of Van Gogh had asked for people to make as much noise as possible in support of the freedom of speech.

"Theo didn't want any police protection," of van Gogh in a telephone interview.

Van Gogh received many threats after the film was shown but always laughed them off.

+

Content Realizaton

Content Realization: Sentence Compression

- Goal: to fit more relevant words into the 100-word limit, and reduce the number of redundant or non-information-full words, to hopefully better our topicality judgments.

Content Realization: Sentence Compression

- Regular Expression Substitutions
  - Remove parentheses around entire sentences
  - Turn double-backticks (``) into quotes
  - Do more byline reduction (most of which is done in the preprocessing step)
  - Remove non-absolute dates (eg. "last Thursday", "in March")
- Dependency Tree Operations
  - Remove prepositional-phrase asides (prepositional phrases beginning with a comma)
  - Remove beginning-of-sentence adverbs and conjunctions
  - Remove attributives
- Other
  - Cleanup
  - Replace contract-able phrases with their contractions (eg. "did not" => "didn't)
- **New**
  - **Remove all quotes**

Compression

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
No compression	0.24153	0.05904	0.01985	0.00813
Post compression	0.24277	0.05941	0.02051	0.00822
Pre compression	**0.24756**	**0.06219**	0.02157	0.00861

Failed Attempts: Coreference Resolution

- Wanted to consider coreferenced entities when calculating cosine similarity.
- Used Stanford CoreNLP to obtain sets of coreferenced entities.

  `(3,5,[5,6]) -> (2,3,[1,4]), that is: "his" -> "Sheriff John Stone"`
- Selected which string to replace other coreferences with:
  - Identifyed all realizations of entity as potential candidate;
  - Filtered out pronouns and any realization with more than 5 tokens (which tended to contain errors);
  - Picked longest remaining candidate.
- Filtered which coreferences to replace:
  - Didn't replace 1st and 2nd person pronouns, to avoid weighting sentences with these words more highly.
  - Didn't replace strings with more than five tokens (again: lots of errors).
- Didn't improve ROUGE scores.

Coreference resolution

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Without:	**0.24756**	**0.06219**	0.02157	0.00861
With:	0.24347	0.05803	0.01959	0.00771

Final settings

Feature	Value
COMPRESSION	before selection
SIMILARITY THRESHOLD	0.1
QUERY WEIGHT	0.5
TFIDF MEASURE USED	idf
WEIGHTING METHOD	own
COREFERENCE RESOLUTION	FALSE
USE COREF REPRESENTATION	FALSE

Results

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Top N	0.21963	0.05173	0.01450	0.00461
Random	0.16282	0.02784	0.00812	0.00334
MEAD	0.22641	0.05966	0.01797	0.00744
PANDAS:				
D2	0.24886	0.06636	0.02031	0.00606
D3	0.24948	0.06730	0.02084	0.00662
D4-dev	0.24756	0.06219	0.02157	0.00861
D4-eval	**0.27315**	**0.07020**	**0.02464**	**0.01137**

Related Reading

Christopher D. Manning, Mihai Surdeanu ad John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the As- sociation for Computational Linguistics: System Demonstrations*, pages 55–60.

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

Jahna Otterbacher, Günes Erkan, and Dragomir R. Radev. 2005. Using Random Walks for Question- focused Sentence Retrieval. In *Proceedings of Hu- man Language Technology Conference and Confer- ence on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 915–922, Van- couver, British Columbia, October.

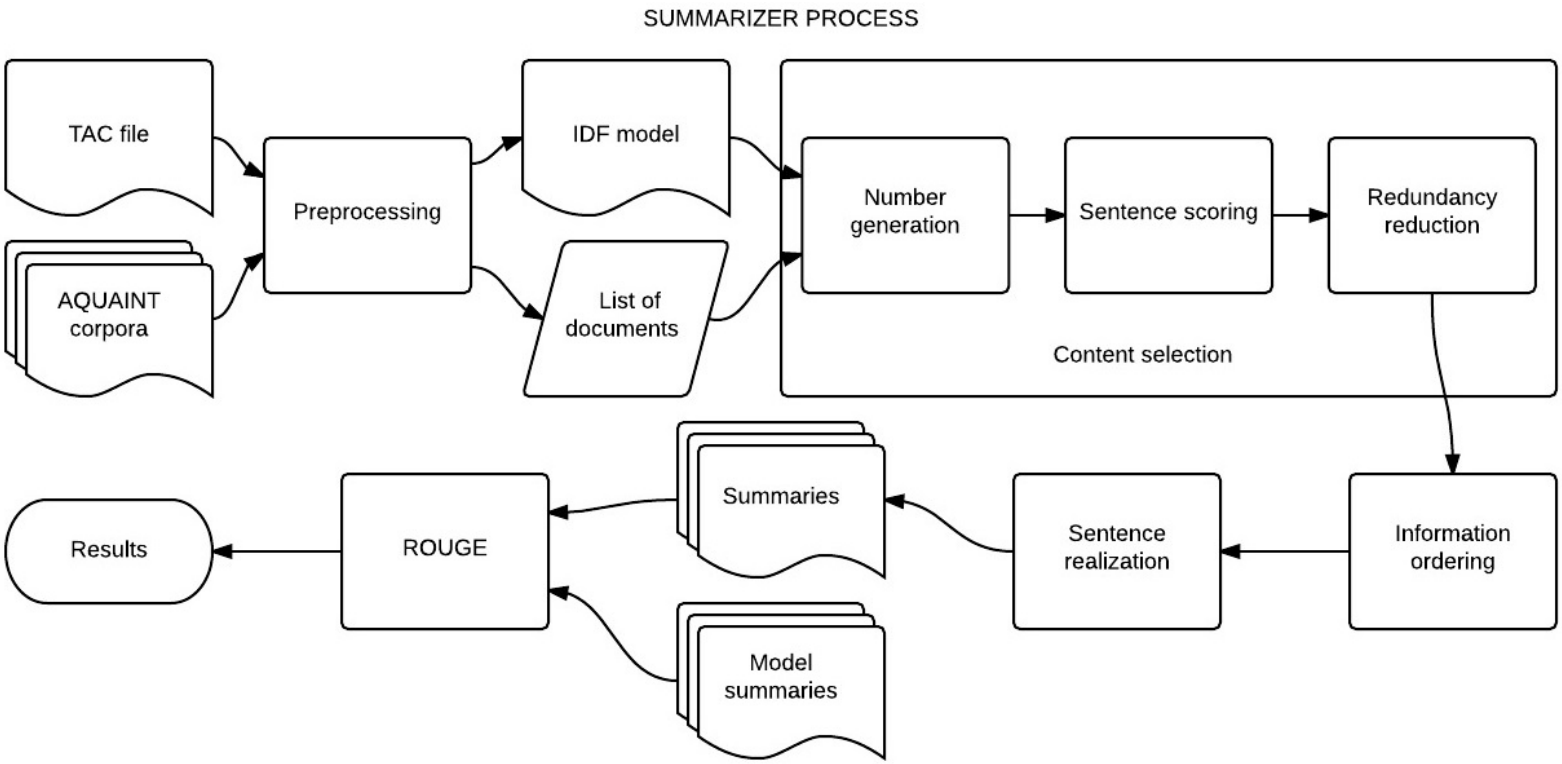
Automatic summarization project
- Deliverable 4 -

Anca Burducea
Joe Mulvey
Nate Perkins

June 2, 2015

Outline

Overall Summary - System Design



Overall Summary - Content Selection

- topic clustering
  - cluster topics based on cosine similarity
  - choose highest ranked sentence in cluster
- sentence scoring
  - methods include: tf-idf with topic signature, position, LLR, NER count, headline, topic (query), average length
  - normalize, apply weights, combine methods
- final system uses: tf-idf 0.7, position 0.3 (Radev et al. 2004)

Overall Summary - Information Ordering

- goal: order sentences that make the final summary
- block ordering (Barzilay et al. 2002)
  - compare two sentences by the original cluster they came from
  - group sentences whose cluster has a high percentage of coming from the same topic segment (window of 5 sentences)
- sort blocks internally by time stamp
- sort each block by time stamp

Outline

Sentence Realization

- used Stanford parser to parse each sentence
- removed insignificant nodes (before content selection) (Silveira & Branco, 2014)
- cleaned up errors (punctuation, capitalization) caused by pruning nodes (after content selection and information ordering)

Sentence Realization - Prune Nodes

- Wh-adverbial/adjectival phrases: I ran home when I saw him.
- interjections: Well, I like chicken.
- parentheticals: Michael (a.k.a. Mike) is cool.
- fragments: On Thursday.
- direct child of ROOT that is not a clause: The house on the left.
- initial prepositional phrases: Last Sunday his boat sunk.
- gerunds surrounded by commas: This city , raining all the time, sucks.
- adverbs that are direct child of S node: It seriously sucks.

Sentence Realization - Fix Bugs

- remove location header from first sentences
  - **ATHENS, Greece** – A Cypriot passenger plane with 121 people ⇒ A Cypriot passenger plane with 121 people
- fix sentences incorrectly split (NLTK's sentence tokenizer)
  - "We've never had a Category 5 hurricane hit the east coast and this storm is just under **that.** ⇒ "We've never had a Category 5 hurricane hit the east coast and this storm is just under that."
- fix punctuation/capitalization errors caused by pruning nodes
  - **, t**he officers have said they thought Diallo had a gun. ⇒ The officers have said they thought Diallo had a gun.

Sentence Realization - Mixed Results

- some good, some bad results from sentence realization
- actual good example
  - remove initial PP, fix resulting punctuation/capitalization
  - **Through their lawyers,** the officers have said they thought Diallo had a gun. $\Rightarrow$ The officers have said they thought Diallo had a gun.
- actual bad example
  - remove WHADVP nodes when child of SBAR
  - "Rescue ships collected scores of bloated corpses Monday from seas close to **where** an Indonesian ferry sank in the Java Sea" $\Rightarrow$ "Rescue ships collected scores of bloated corpses Monday from seas close to an Indonesian ferry sank in the Java Sea"

Outline

Final Results - Deliverable Comparisons

ROUGE R scores:

	LEAD	D2	D3	D4
ROUGE-1	0.19143	**0.25909**	0.25467	0.25642
ROUGE-2	0.04542	0.06453	**0.06706**	0.06696
ROUGE-3	0.01196	0.01881	**0.02043**	0.02015
ROUGE-4	0.00306	**0.00724**	0.00642	0.00643

Final Results - Eval Numbers

ROUGE scores:

	R	P	F
ROUGE-1	0.30459	0.33251	0.31699
ROUGE-2	0.09399	0.10111	0.09714
ROUGE-3	0.03553	0.03752	0.03639
ROUGE-4	0.01786	0.01850	0.01813

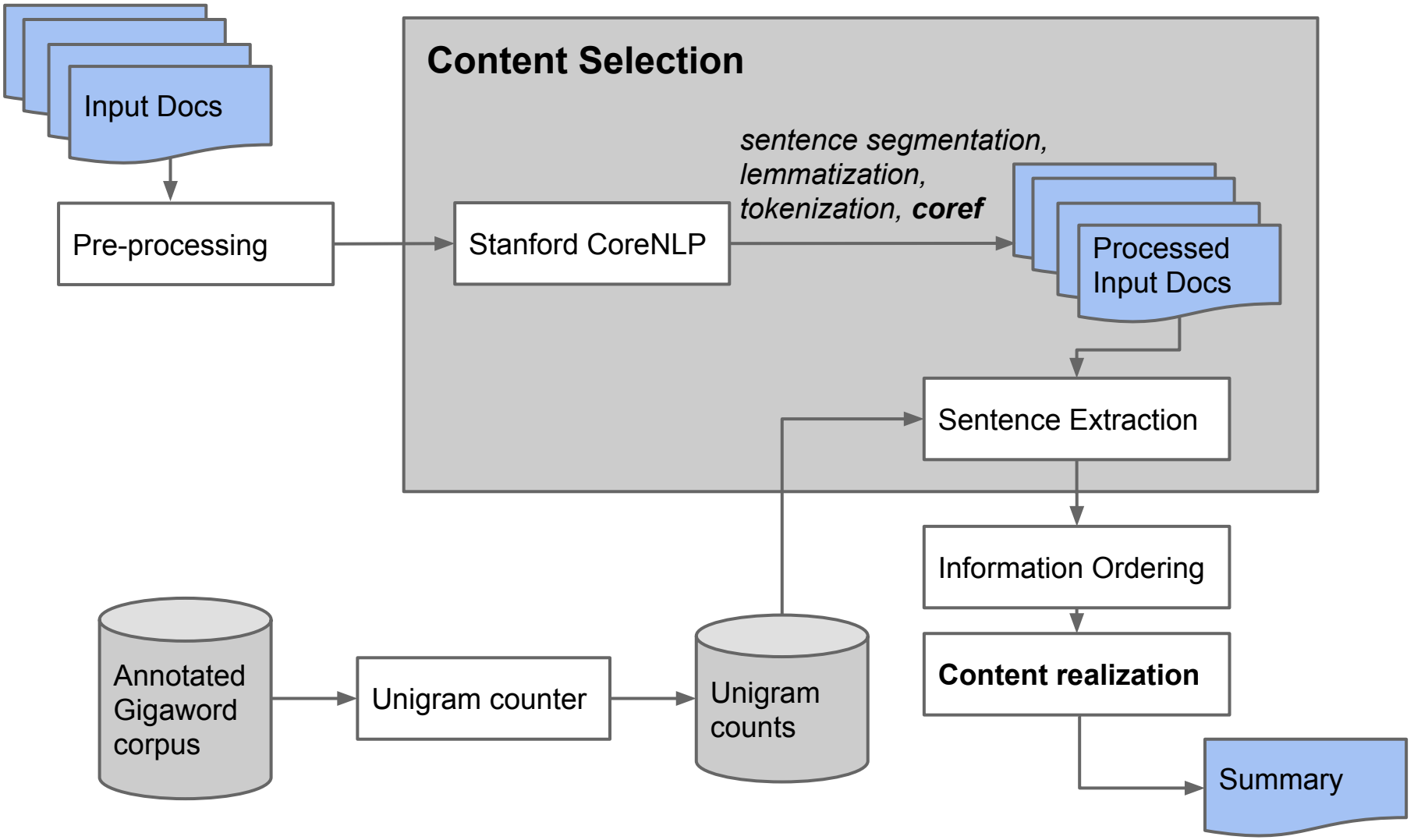
Final Results - Summary Example

"Monitoring before the earthquake did not detect any macroscopic abnormalities, and did not catch any relevant information," said Deng Changwen, deputy head of Sichuan province's earthquake department. The 7.8-magnitude earthquake struck Sichuan province shortly before 2:30 pm on Monday. The ASEAN Inter-Parliamentary Assembly on Wednesday expressed its condolence and sympathy to China following the devastating earthquake in Sichuan province. Vietnam has expressed deep sympathies to China at huge losses caused by an earthquake in China's southwestern Sichuan province, Vietnam News Agency reported Tuesday. The German government announced on Tuesday that it is to provide 500,000 euros in aid for earthquake victims in Sichuan Province of China.

LING 573 Deliverable #4

George Cooper, Wei Dai, Kazuki Shintani

System Overview



Pre-processing

Sentence Segmentation Effort

- Stanford CoreNLP segments sentences wrong for sentences like:
  - "Did you question this procedure?" the judge asked.
  - It is parsed as two different sentences:
    - "Did you question this procedure?"
    - the judge asked.
- Used NLTK but same thing happened...
- So, concatenated these sentences back together, after NLTK, and told Stanford CoreNLP to segment by newlines
- But ROUGE score didn’t improve

Content Selection

Algorithm Overview

- Modeled after KLSum algorithm
- Goal: Minimize KL Divergence between summary and original documents
- Testing every possible summary is $O(2^n)$, so we used a beam search over log-likelihood weighted vectors

Incorporating Coreferences

- Use Stanford CoreNLP's coreferences
- When the pos tag is personal pronoun, substitute it with the coreference representative for content selection
- But don't replace the word itself into the final summary
- Conditionally apply coref substitution, based on lemmas (he, she, etc), capitalization, number of words, and threshold per sentence, etc

Information Ordering

Information Ordering I

- Cluster the articles by topic
  - $distance(D_x, D_y) = 1 - \frac{D_x \cdot D_y}{|D_x||D_y|}$
  - merge pair of clusters when the distance is lower than a threshold (< 0.5).
- Order over clusters by CO
  - pick the date of the earliest article in a cluster as the date of cluster, then sort the clusters.
- Order sentences within each cluster by CO
  - use combination of article date and in article sentence order to sort the sentences.

Information Ordering II

- Cluster sentences by topics with LDA
  - Create lemma vectors corpora of original document collections, filtering out stop words.
  - Generate topics cluster using Latent Dirichlet allocation model (set the number of topic to 3).
  - Cluster selected sentences based on the topics.
- Order clusters by CO
- Order sentences within each cluster by CO

Information Ordering III

- Set the most representative sentence always the first sentence in the summary.
- Set the very short sentences to the end of the summary, length < 3 (after filtering out stop words)
- Order the other sentences based on the approach in Information Ordering I and II.

Content Realization

Sentence Compression

- We created nine hand-written sentence compression rules based on the phrase structure parse of the sentence from Stanford CoreNLP
- A rule only fires if doing so decreases the KL-divergence between that sentence and the document collection
- Compression rules do not change the vector representations of the sentence or the document collection

Sentence Compression

- Rules are executed in the order of the number of words they would eliminate, smallest to largest

Compression Rules

Remove Parentheticals

- Remove nodes of type PRN
- **Example:** “The central and provincial governments have invested 160 million yuan ~~(nearly 20 million US dollars)~~ into panda protection programs since 1992.”

Remove temporal NPs

- Remove nodes of type NP-TMP
- **Example:** "~~Today,~~ a major treatment strategy is aimed at developing medicines to stop this abnormal protein from clumping."

Remove adverb phrases

- Remove nodes of type ADVP
- **Example:** “Hugs have become a greeting of choice even~~, sometimes,~~ between strangers.”

Remove prepositional phrases

- Remove nodes of type PP
- **Example:** “The SEPA confirmed the "major pollution" ~~of the Songhua River~~ ~~on Wednesday~~.”

Remove relative clauses

- Remove nodes of type WHNP whose parent is an SBAR
- **Example:** “But ads also persuade people to spend money on unnecessary drugs~~, which is a bad thing for their health and for insurance premiums~~.”

Remove adjectives

- Remove nodes of type JJ, JJR, ADJP, and S whose parent is an NP
- **Example:** “Out of his death comes a ~~stronger~~ need to defend the fresh air of Lebanon.”

Remove introductions

- Remove nodes of type "S → ~~SBAR ,~~ …"
- **Example:** "~~Though the plane was out of radio contact with the ground for more than an hour after that,~~ it appeared that at least some passengers remained conscious."

Remove attributives

- Remove nodes of type “S → S , ~~NP VP~~ .” and “S → `` S , " ~~NP VP~~ .”
- **Example:** “The Warapu village had also been completely destroyed, with 11 confirmed deaths and many missing~~, Igara said~~.”

Remove second element of conjoined phrases

- Remove nodes of type “XP ~~CC XP~~”
- **Example:** “Then there is the Chinese oyster, which governors in Maryland and Virginia believe might resist disease ~~and provide a natural pollution filter~~.”

Remove initial conjunctions

- Remove nodes of type “~~CC~~ ...”
- **Example:** “~~But~~ it's also frisky and funny, with a streak of unconditional kindness as wide as the screen.”

Attempted Improvements

- Replace words in the original documents with the appropriate contractions (e.g. “can not” → “can’t”)

Post-processing

- Clean up partial quotation marks in the summaries.
  - Count the quotation marks in each sentence in the summary, if odd number, check the sentence:
    - A quotation mark found at the first or last place in a sentence, add a quotation mark at the last or the first place.
    - A quotation mark found in the middle of a sentence, check the original article the sentence belongs to, add a quotation mark at front or end based on the original texts.

EX: John Kerry supports stem cell research."

The young killers of the … ," Gore said.

… saying: " The government is responsible for ...

Results

Results: Coref Substitution

coref substitution max count	max occurrence scope	max word count	ROUGE1	ROUGE2	ROUGE3	ROUGE4
baseline (no substitution)			0.31045	0.09215	0.03379	0.01247
1	document	1	0.31045	0.09215	0.03379	0.01247
1	document	2	0.31010	0.09197	0.03379	0.01247
1	document	3	0.31189	0.09294	0.03409	0.01279
1	**document**	**4**	**0.31206**	**0.09312**	**0.03418**	**0.01279**
1	sentence	1	0.31045	0.09215	0.03379	0.01247
1	sentence	2	0.31047	0.09197	0.03379	0.01247
1	sentence	3	0.30942	0.08925	0.03169	0.01162
1	sentence	4	0.31148	0.09052	0.03283	0.01251

Results: Coref Substitution

coref substitution max count	max occurrence scope	max word count	ROUGE1	ROUGE2	ROUGE3	ROUGE4
baseline (no substitution)			0.31045	0.09215	0.03379	0.01247
2	document	1	0.31045	0.09215	0.03379	0.01247
2	document	2	0.30991	0.09181	0.03379	0.01247
2	document	3	0.31015	0.09222	0.03380	0.01256
2	document	4	0.31166	0.09258	0.03389	0.01256
2	sentence	1	0.31045	0.09215	0.03379	0.01247
2	sentence	2	0.31000	0.09210	0.03408	0.01267
2	sentence	3	0.30470	0.08795	0.03119	0.01095
2	sentence	4	0.30388	0.08712	0.03100	0.01085

Results: Coref Substitution

coref substitution max count	max occurrence scope	max word count	ROUGE1	ROUGE2	ROUGE3	ROUGE4
baseline (no substitution)			0.31045	0.09215	0.03379	0.01247
3	document	1	0.31045	0.09215	0.03379	0.01247
3	document	2	0.30991	0.09181	0.03379	0.01247
3	document	3	0.30980	0.09195	0.03371	0.01256
3	document	4	0.30918	0.09113	0.03315	0.01228
3	sentence	1	0.31045	0.09215	0.03379	0.01247
3	sentence	2	0.31000	0.09210	0.03408	0.01267
3	sentence	3	0.30515	0.08877	0.03156	0.01113
3	sentence	4	0.30245	0.08575	0.03004	0.01014

Results: Coref Substitution

pronouns	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
she	0.31009	0.09206	0.03407	0.01285
he	0.31009	0.09206	0.03407	0.01285
they	0.31045	0.09215	0.03379	0.01247
she, he	0.31145	0.09212	0.03298	0.01177
she, they	0.31009	0.09206	0.03407	0.01285
he, they	0.31145	0.09212	0.03298	0.01177
she, he, they	**0.31206**	**0.09312**	**0.03418**	**0.01279**

Results: compression rules

size	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
no compression	0.30828	0.09152	0.03384	0.01265
parentheticals	**0.31189**	**0.09293**	**0.03397**	**0.01256**
temporal NPs	0.30616	0.09136	0.03372	0.01265
adverb phrases	0.31189	0.09284	0.03388	0.01247
prepositional phrases	0.31320	0.09142	0.03185	0.01149
relative clauses	0.31065	0.09250	0.03391	0.01243
adjectives	0.30542	0.08760	0.03017	0.00975
introductions	0.30873	0.09168	0.03384	0.01255
attributives	0.30678	0.09105	0.03392	0.01283
conjunctions (1)	0.30939	0.09049	0.03275	0.01219
conjunctions (2)	0.30980	0.09200	0.03413	0.01265

Results: compression rules

size	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
parentheticals	**0.31189**	**0.09293**	**0.03397**	**0.01256**
parenth. + adv. phr.	0.31145	0.09284	0.03388	0.01247
parenth. + rel. clause	0.31050	0.09194	0.03372	0.01243
parenth. + intro.	0.30829	0.09120	0.03364	0.01255
parenth. + conj. (2)	0.30662	0.08967	0.03219	0.01088
adv. phr. + rel. clause	0.31070	0.09060	0.03303	0.01212
adv. phr. + intro.	0.31214	0.09275	0.03388	0.01247
adv. phr. + conj. (2)	0.30828	0.09025	0.03154	0.01060
intro. + conj. (2)	0.31109	0.09240	0.03382	0.01233
intro. + rel. clause	0.31193	0.09290	0.03411	0.01243
conj. (2) + rel. clause	0.31024	0.09216	0.03413	0.01255

Results: effect of KL-divergence on compression rules

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
parentheticals (with KL-divergence)	**0.31189**	**0.09293**	**0.03397**	**0.01256**
parentheticals (without KL-divergence)	0.30803	0.09122	0.03374	0.01265
no compression	0.30828	0.09152	0.03384	0.01265

Results: D4 final ROUGE scores

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
devtest	0.31189	0.09312	0.03409	0.01279
evaltest	0.34491	0.10569	0.03840	0.01827

Discussion

Potential Improvements

- Incorporate global word probabilities
- Try more targeted sentence compression patterns
- Use coreference to prevent pronouns/shortened forms from occurring in the summary without or before the corresponding full form
- Using NER to adjust unigram weight

Summarization Task - D4

LING573

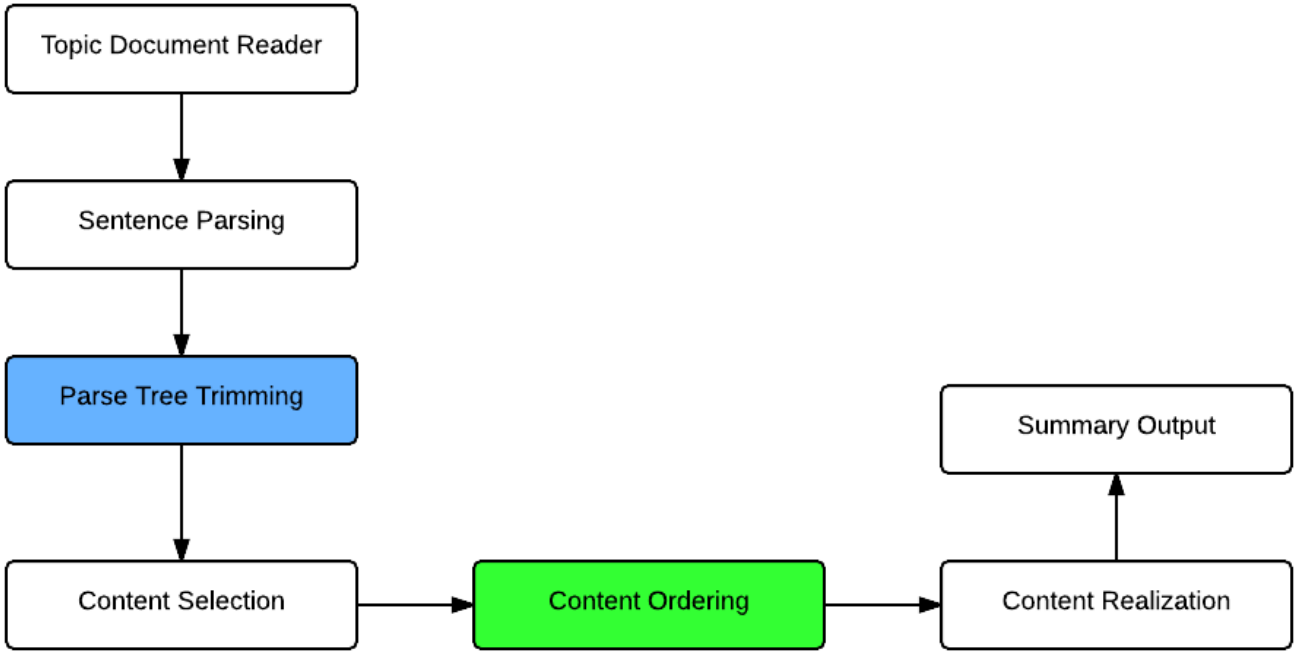
Team Members

John Ho

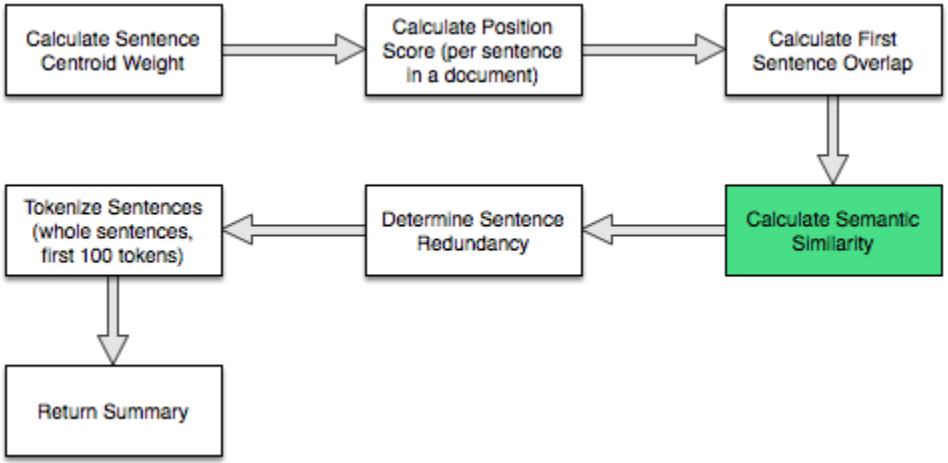
Nick Chen

Oscar Castaneda

System Overview



Content Selection



Compression & Parse Tree Trimming

- We created a function that removed *all* adjectives and adverbs from sentences, but we decided not to use it since it lowered our ROUGE scores.

Semantic Similarity

- Sentence Similarity Based on Semantic Nets and Corpus Statistics (Yuhua Li and David Mclean and Zuhair B and James D. O'shea and Keeley Crockett)
- WordNet
- Semantic Similarity and Word Order Similarity

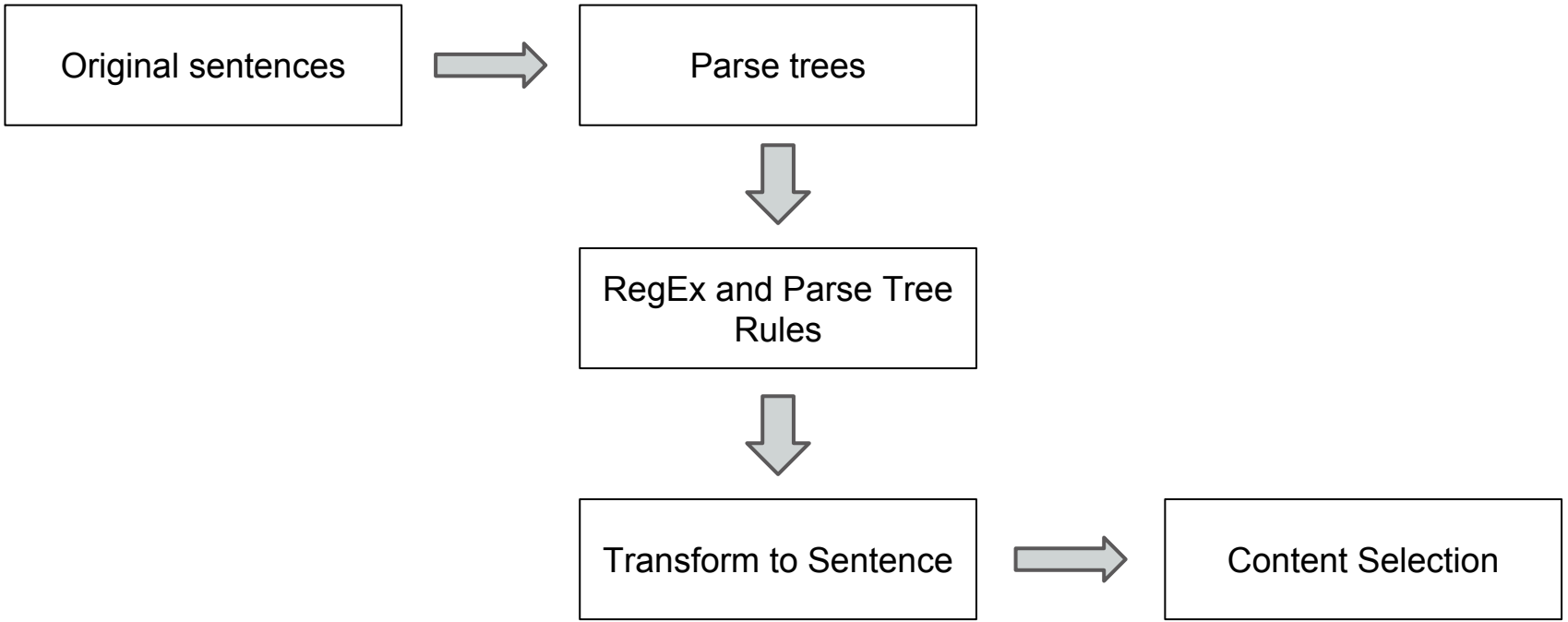
Sentence Ordering

- Modeling Local Coherence: An Entity-Based Approach (Barzilay and Lapata)
- Ignore the salience measure
- SVM RANK for ML
- MaltParser for dependency
- Stanford Dependencies list

Sentence Compression

- We added preprocessing rules that we ran before content selection in order to reduce the amount of “noise” in our input data.
- We tried applying rules that eliminated the sentence POS that matched what was done in the CLASSY system.
- Rules based on Stanford Tree Parsing

Sentence Compression Diagram



Sentence Compression Rules

- Applied rules:
  - Initial adverbials and conjunctions
  - Gerund phrases
  - Relative clauses / appositives
  - Other adverbials (focused on those that appear at the end)
  - Numeric data
  - Attributives
  - Junk data (things that didn’t parse / SBARS)

Results

Degressed!

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
D2	0.21573	0.06417	0.02399	0.00981
D3	0.23455	0.06784	0.02657	0.01093
D4	0.22260	0.04718	0.01473	0.00619
~~Improvement~~ Decline	1.29%	24.32%	34.36%	24.05%

Results

D4 Results

	ROUGE-1	ROUGE-2	ROUGE-3	ROUGE-4
Eval	0.24821	0.05506	0.01699	0.00694
D4/Dev	0.22260	0.04718	0.01473	0.00619

Challenges

- Parse Tree (Stanford LexParse)
- Keywords

Questions?

Thanks for listening!