Summarization Evaluation & Some Model Systems

LING 573

Systems and Applications

April 3, 2018

Begin Recording!

Announcements

- GitLab accounts should all be created now
  - If this held up your D1, let me know.
- If using GitHub/GitLab:
  - Make sure you have invited David and I as collaborators, or as part of your organization

Roadmap

- Evaluation Recap
- Content Selection
  - Model classes
  - Unsupervised word-based models
    - Sumbasic
    - LLR
    - MEAD

Evaluation: Recap

Evaluation: Recap

- Evaluating summaries is hard.
- How do we quantify "aboutness?"
- Compare four approaches
  - Manual
  - ROUGE
  - Pyramid
  - "Model-Free"

Evaluation: Manual Quality Assessment

- What makes a good quality summary?
- NIST Evaluation Guidelines look at two dimensions:
  - Linguistic Quality
  - Content Quality

Manual Evaluation: Linguistic Quality

- Use **1-5** Scale for each:
  - **Grammaticality** — Should be fluent, artifact-free (no XML, datelines, etc)
  - **Non-Redundancy** — No unnecessary repetition
  - **Referential Clarity** — Should be easy to resolve pronouns, acronyms, etc.
  - **Focus** — Should stay on-topic
  - **Structure and Coherence** — Sentence transitions should be smooth.

Manual Evaluation: Content Quality

- Use **1-5** Scale for each:
  - **Responsiveness** — Amount of relevant content to the query
  - **Overall** — Overall, how well the summary responds to the query

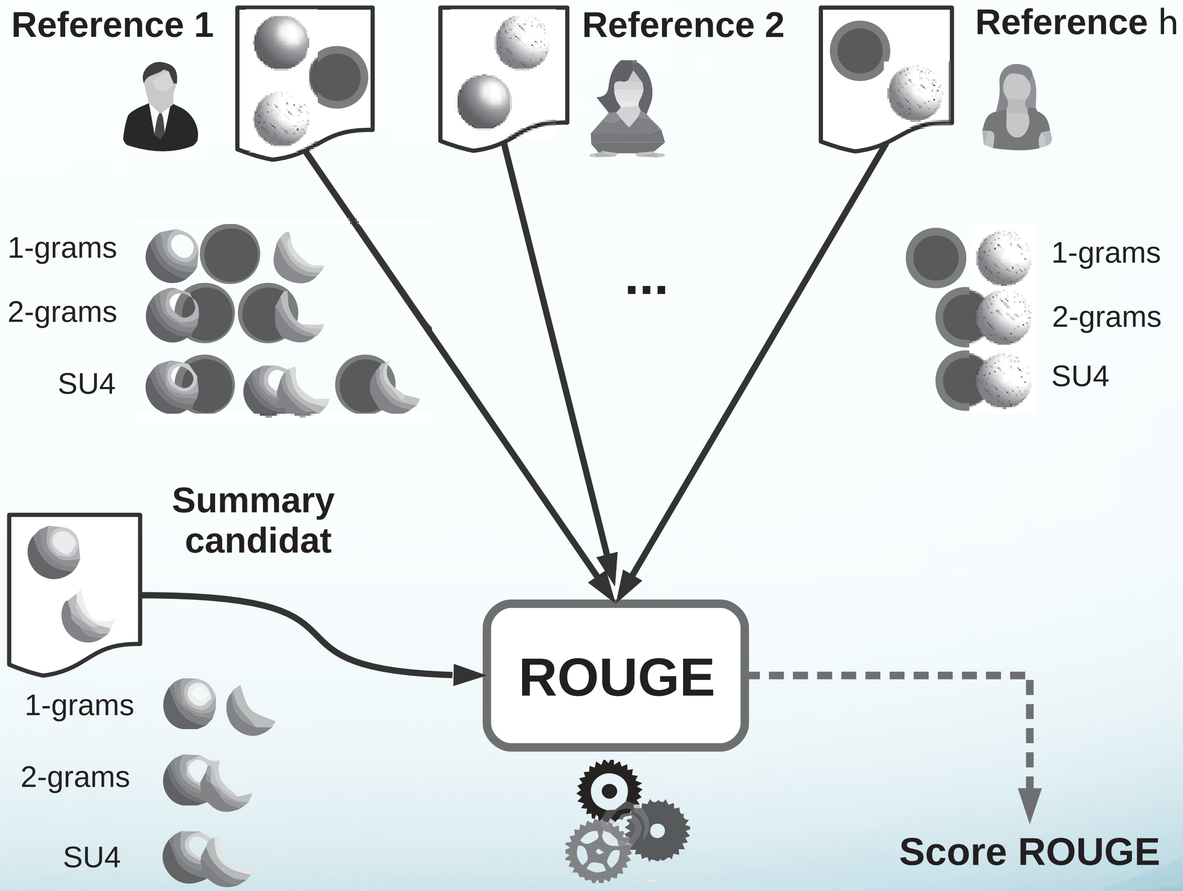
Evaluation

- With these manual analyses in mind, how do existing intrinsic metrics evaluate?
- Generally, by comparing against summaries written by humans
  - String matches (ROUGE)
  - Semantic matches (Pyramid)

Evaluation: ROUGE-*n*

- *n*-gram overlap between system output and reference summaries

$$\text{ROUGE-}n = \frac{\sum_{n\text{-grams}} \in \{\text{Sum}_{\text{sys}} \cap \text{Sum}_{\text{ref}}\}}{\sum_{n\text{-grams}} \in \text{Sum}_{\text{ref}}}$$



via (Torres-Moreno, 2014, p. 258)

Evaluation: ROUGE-Sγ

- Skipgrams, with up to γ intervening elements

Sentence	***Intellectuals solve problems; geniuses prevent them***	**Count**
Rouge-1	Intellectuals ▸ solve ▸ problems ▸ geniuses ▸ prevent ▸ them	6
Rouge-2	Intellectuals solve ▸ solve problems ▸ problems geniuses ▸ geniuses prevent ▸ prevent them	5
Rouge-SU4	Intellectuals solve ▸ Intellectuals … problems ▸ Intellectuals … geniuses ▸ solve problems ▸ solve … geniuses ▸ solve … prevent ▸ problems geniuses ▸ problems … prevent ▸ problems … them ▸ geniuses prevent ▸ geniuses …them ▸ prevent them	12

Evaluation: ROUGE-L

- L, for Longest Common Substring (LCS)
- Greedily find LCS matches between system and references

ROUGE: Gaming The System

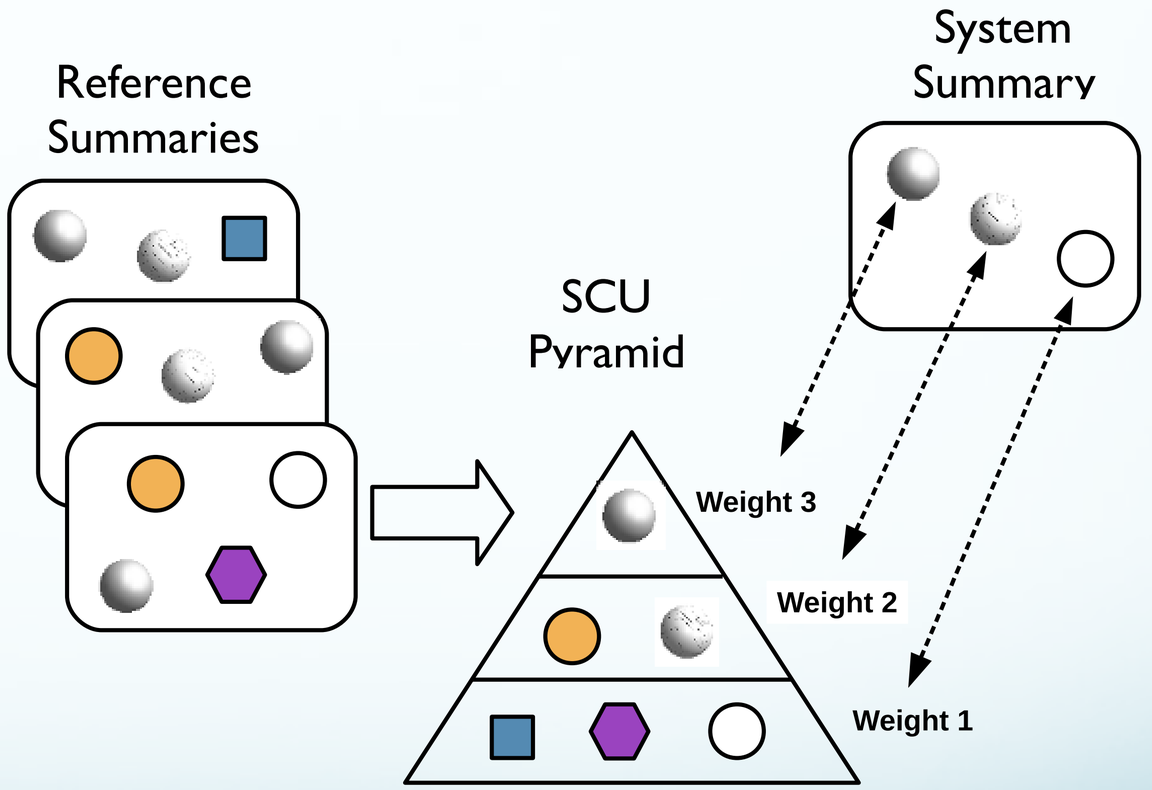
- Sjöbergh (2007):
  - Find most frequent bigram among topic set
  - Seed Markov chain generator with this bigram
    - Proceed to generate summary based on n-gram language model:

> *of the hurricane andrew had been injured and the storm caused by the gulf of mexico and louisiana and at least 150 000 people died on the us insurers expect to the florida and there are likely to be concentrated among other insurers have been badly damaged a result of damage caused in the state and to dollars 20bn of new orleans then to pay out on monday and new iberia associated tornadoes devastated inhabitants of miami…*

- …don't do this.

Evaluation: Pyramid

- Summary Content Units (SCUs)
  - Semantic-based, sub-sentential units
  - Weighted by prevalence among documents
- Optimal score uses only top weights until none remain, then lower weights.
- Compute score by:
  - Sum all SCUs × weight
  - Ratio of System :: Optimal
- ("Modified" normalizes by avg. # SCUs instead of all SCUs — helps with recall)



“Model-Free” Evaluation

- ROUGE, Pyramid:
  - Both rely on target summaries written by humans.
- Options for evaluation without references?

“Model-Free” Evaluation

- Model summaries and topic documents as n-gram distributions
- Compare divergence of distributions
  - Kullback–Leibler, Jensen–Shannon
- Check correlation
- Multiple implementations:
  - SIMetrix — (Louis & Nenkova, 2009)
  - Fresa — (Saggion et. al, 2010)

Assessing Evaluation Metrics

- Extrinsic evaluation is hard for this task
- Will often see this metrics in the literature reported as correlation to manual

Assessment

- Correlation with manual score-based rankings
  - Distributional measure well-correlated, similar to ROUGE2

Features	pyramid	respons.
JS div	-0.880	-0.736
JS div smoothed	-0.874	-0.737
% of input topic words	0.795	0.627
KL div summ-inp	-0.763	-0.694
cosine overlap	0.712	0.647
% of summ = topc wd	0.712	0.602
topic overlap	0.699	0.629
KL div inp-summ	-0.688	-0.585
mult. summary prob.	0.222	0.235
unigram summary prob	-0.188	-0.101
regression	0.867	0.705
ROUGE-1 recall	0.859	0.806
ROUGE-2 recall	0.905	0.873

Spearman correlations from Louis & Nenkova (2009), with $p < 0.000001$

Content Selection & Example Systems

Content Selection & Example Systems

- Start drilling down into the task with a few example systems
- Focus on content selection
  - Most important task in extractive approach, if optimizing for ROUGE-*n*

Content Selection

- **Information-source based**
  - Words, discourse (position, structure), POS, NER, etc
- **Learner-based**
  - Supervised — classification/regression, unsupervised, semi-supervised
- **Models**
  - Graphs, LSA, ILP, submodularity, Info-theoretic, LDA

Content Selection

- **Information-source based**
  - **Words**, discourse (position, structure), POS, NER, etc
- **Learner-based**
  - Supervised — classification/regression, unsupervised, semi-supervised
- **Models**
  - Graphs, LSA, ILP, submodularity, Info-theoretic, LDA

SumBasic (Nenkova & Vanderwende, 2005)

- Intuition:
  - Word frequencies in document set correlate with words in human summaries:
  - Percentage of top N words in document set that appear in a reference summary:
    - Top 5 — 94.66%
    - Top 8 — 91.25%
    - Top 12 — 95.25%

SumBasic (Nenkova & Vanderwende, 2005)

1. Estimate word probabilities from doc(s)
2. Score sentences
   - Sentence weight = average probability of words in sentence
   - (having removed stopwords)
3. Pick best scoring sentence, add to summary.

$$Weight(S_j) = \sum_{W_i \in S_j} \frac{p(w_i)}{|\{w_i | w_i \in S_j\}}$$

Scoring Formula

SumBasic (Nenkova & Vanderwende, 2005)

4. Reweight the words in the chosen sentence by squaring their probability
   - This serves to de-prioritize already-chosen words.
5. Repeat until summary length is reached.

Word Weight Example

1. Bombing Pan Am…
2. Libya Gadafhi supports…
3. Trail suspects…
4. UK and USA…

Word	Weight
Pan	0.0798
Am	0.0825
Libya	0.0096
Supports	0.0341
Gadafhi	0.0911

Libya refuses to surrender
two Pan Am bombing suspects

Nenkova, 2011

Limitations of Frequency

- Basic approach actually works fairly well
- However, misses some key information:
  - No notion of foreground/background contrast
    - Is a word that's frequent everywhere a good choice?
  - Surface form match only
    - Want concept frequency, not just word frequency
      - WordNet, LSA, LDA, etc

Modeling Background

- Want to contrast between document(s) being summarized, other content
- Words/phrases indicative of the topic = "topic signatures"
  - (Compared to general background words/phrases)
- Several solutions:
  - TF*IDF
  - LLR
  - Sentence Clustering

Modeling Background: TF*IDF Refresher

- word count = $f_{t,\ d}$
- Term Frequency = # of occurrences in document (set) = $\frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$
- Inverse Document Frequency:
  - DF = # docs with term
  - $IDF = \log(\frac{N}{\mathrm{df}_w})$
  - where N is total number of documents in corpus
  - (And using log space to prevent underflow)

Log Likelihood Ratio

- Log Likelihood Ratio (LLR) $\lambda(w)$
  - Unlike TF*IDF, sets threshold for dividing words in input to descriptive or not
  - Compare likelihood of input I to background B
  - Two hypotheses:
    - H1 — Probability of word in the input is same as in background
    - H2 — Word has a different, higher probability in input than background
  - This is the ratio between the two hypotheses

Log Likelihood Ratio

- k_1 = count of w in topic cluster
- k_2 = count of w in background corpus
- n_1 = # docs in topic cluster; n_2 = # in background
- $p_1 = k_1/n_1 \qquad p_2 = k_2/n_2 \qquad p = (k_1+k_2)/(n_1+n_2)$
- $-2log\lambda = 2[\log L(p_1,k_1,n_1) + \log L(p_2, k_2, n_2) - \log L(p, k_1, n_1) - \log L(p, k_2, n_2)]$

Using LLR for Weighting

- Compute weight for all cluster terms
  - weight(w_i) = 1 of -2log λ > 10, 0 o.w.
- Use that to compute sentence weights $$weight(s_i) = \sum_{w \in s_i} \frac{weight(w)}{|\{w|w \in s_i\}|}$$
- How do we use the weights?
  - One option: directly rank sentences for extraction
- LLR-based systems historically perform well
  - Better than tf*idf generally

Next Time:

- Other content-selection approaches
- Sentence Clustering
- Supervised/ML Approaches
- Graph-Based Approaches

And Now… This

- Let's walk through a git merge conflict resolution tutorial
  - https://github.com/nruth/git-conflict-resolution-demo