Automated Summarization System

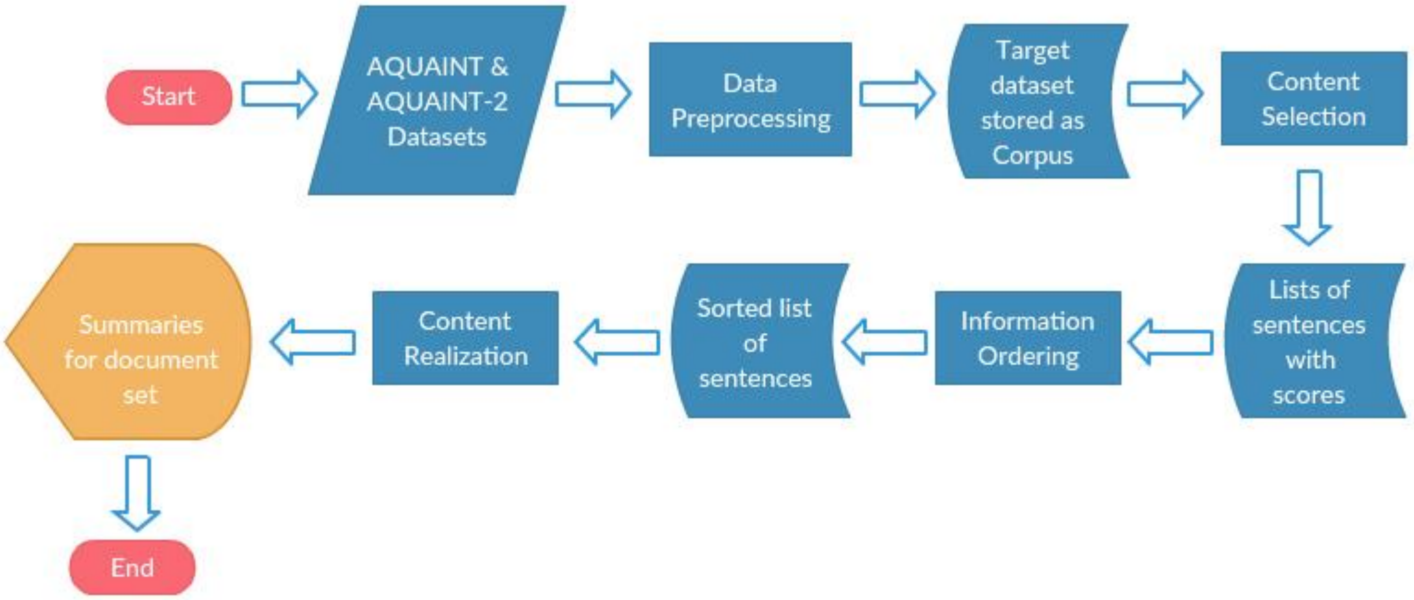
Haobo Gu, Yuanhe Tian, Weifeng Jin, Haotian Zhu

May 29th, 2018

Overview: Improved System

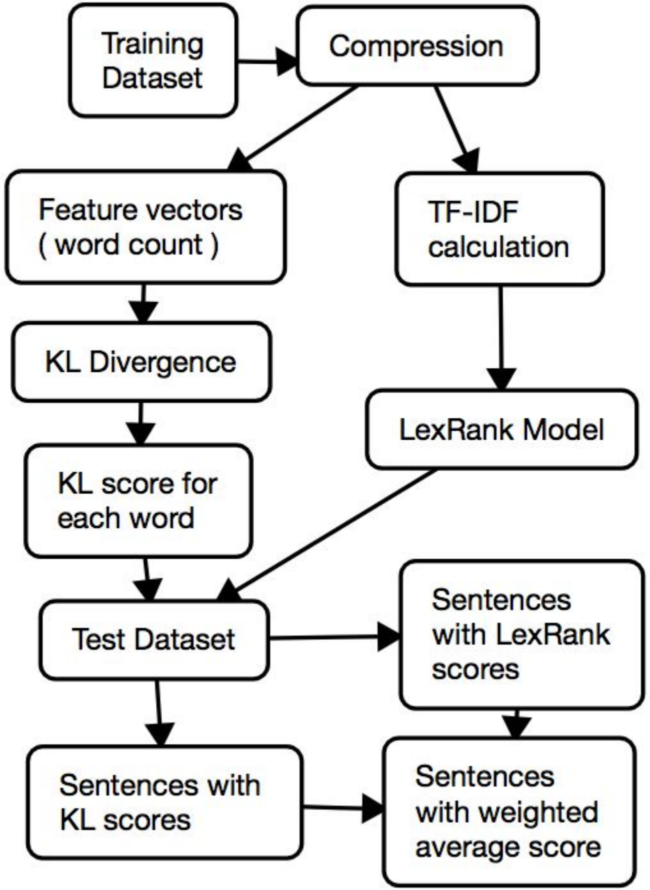
- Combination of both unsupervised and supervised methods
- Extraction-based approach with ***compression***
- Three parts:
    - Improvement of Content Selection
    - Improvement of Information Ordering
    - Improvement of Content Realization

System Flowchart



Improvements of Content Selection

- Compress sentences before calculate their salient score
- Use tf * idf to represent sentence feature vector when calculating its LexRank salient score



Content Selection Flow chart

Compression

A **sequence-labeling** task

O : word to be compressed

B: beginning of the phrase (uncompressed)

I: inside the phrase (uncompressed)

Features: word itself, word last 3 letters, word last 2 letters, if stopword, if negative word, pos tag, previous word, if previous word stopword, if previous word negative, previous word pos tag, next word, if next word stopword, if next word negative, next word pos tag

Training Method: CRF, trained on the compression corpus on Patas, eliminate punctuation-only sentences in the end.

Information Ordering

- **Preference Learning Approach** (Bollegala et al.)
  - Used multiple experts that model different factors
  - Combined experts linearly to determine ordering
  - Built experts for each preference:
    - Chronological
    - Probabilistic (discarded)
    - Topical-closeness
    - Precedence
    - Succession

Information Ordering

- **Chronological Expert**
    - Define a preference function as follows:
        - T(u) is the publication time of sentence u.
        - D(u) is the unique id code of sentence u's document.
        - N(u) is the in-text index of sentence u in it document.

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

Information Ordering

- **Topical-closeness Expert**
  - Use cosine similarity of two sentences
  - Define topic-closeness *topic(u)* as the maximum similarity between *u* and ordered sentences
  - Preference is defined as follows:

$$\mathrm{PREF}_{topic}(u, v, Q) = \begin{cases} 0.5 & [Q = \emptyset v] \vee [\mathrm{topic}(u) = \mathrm{topic}(v)] \\ 1 & [Q \neq \emptyset] \wedge [\mathrm{topic}(u) > \mathrm{topic}(v)] \\ 0 & \text{otherwise} \end{cases}$$

Information Ordering

- **Precedence** and **Succession Experts**
  - Use cosine similarity between two sentences
  - Look at sentences that precede or succeed already ordered sentences in original document
  - Find the average of maximal similarity between sentence *u* and sentences preceding or succeeding ordered sentences to calculate *pre(u)* and *succ(u)*
  - Preference function is defined as follows:

$$\mathrm{PREF}_{pre}(u, v, Q) = \begin{cases} 0.5 & [Q = \emptyset] \vee [\mathrm{pre}(u) = \mathrm{pre}(v)] \\ 1 & [Q \neq \emptyset] \wedge [\mathrm{pre}(u) > \mathrm{pre}(v)] \\ 0 & \text{otherwise} \end{cases}.$$

Information Ordering

- **Sentence Ordering Algorithm**
    - Used learned weights on experts since there is no training data available.
    - Chronological, precedence and succession experts are weighted relatively heavily.

Algorithm 1 Sentence Ordering Algorithm.

Input: A set $\mathcal{X}$ of the extracted (unordered) sentences and a total preference function $\mathrm{PREF}_{total}(u, v, Q)$.

Output: Ranking score $\hat{\rho}(t)$ of each sentence $t \in \mathcal{X}$.

$\mathcal{V} = \mathcal{X}$
$Q = \oslash$
for each $v \in \mathcal{V}$ **do**
    $\pi(v) = \sum_{u \in \mathcal{V}} \mathrm{PREF}_{total}(v, u, Q) - \sum_{u \in V} \mathrm{PREF}_{total}(u, v, Q)$
end for
while $\mathcal{V} \neq \oslash$ **do**
    $t = \arg\max_{u \in \mathcal{V}} \pi(u)$
    $\hat{\rho}(t) = |\mathcal{V}|$
    $\mathcal{V} = \mathcal{V} - \{t\}$
    $Q = Q + \{t\}$
    for each $v \in \mathcal{V}$ **do**
        $\pi(v) = \pi(v) + \mathrm{PREF}_{total}(t, v, Q) - \mathrm{PREF}_{total}(v, t, Q)$
    end for
end while
return $\hat{\rho}$

Information Ordering

Input: a list of selected unordered sentences.

Do:

- Compute total preference of each sentence pair.
- Order sentences using Algorithm 1

Output: a sorted list of sentences.

Content Realization

We improved our **Integer Linear Programming** content realization approach:

- Use the bigram occurrence of document rather than occurrence of sentence as the weight in target function
- Consider only meaningful bigrams in target function. Remove bigrams consist of stopwords and punctuations
- Add a small weight for each sentence variables in target function to reduce randomness

Compression in Content Realization

We have 3 approaches to compress the sentence:

- Remove clauses starting with an adverb
- Remove appositions
- Remove contents in a pair of parenthesis

For each sentence, three compressed sentences are generated. Together with the original sentence, four sentences are added as candidates for summary.

An extra constraint in ILP is added to ensure that only one sentence in four candidates will be chosen in final summary.

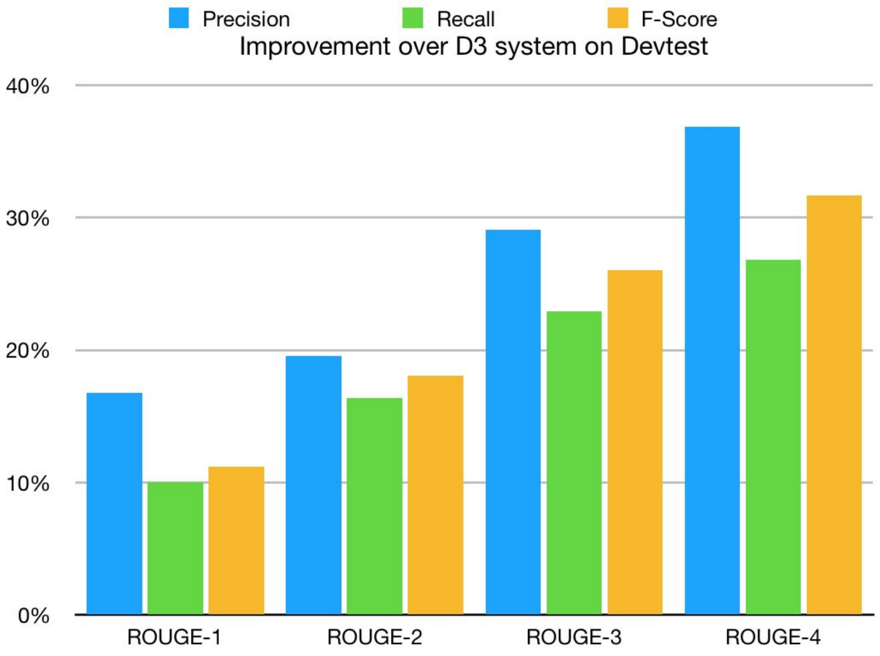
Results and Analysis

Table 1: ROUGE Results for devtest (comp=0.1)

ROUGE	Recall	Precision	F-score
ROUGE-1	0.25985	0.28427	0.27114
ROUGE-2	0.07090	0.07820	0.07428
ROUGE-3	0.02435	0.02695	0.02556
ROUGE-4	0.00946	0.01051	0.00994

Table 2: ROUGE Results for evaltest (comp=0.1)

ROUGE	Recall	Precision	F-score
ROUGE-1	0.28090	0.30808	0.29300
ROUGE-2	0.07803	0.08579	0.08144
ROUGE-3	0.02898	0.03175	0.03019
ROUGE-4	0.01467	0.01591	0.01521



Issues and Successes

Issues:

- Long training time
- Mainly extraction-based method, does not combine several sentences into one.

Successes:

- Big improvement over the baseline system and D3 system
- Combination of both unsupervised and supervised learning methods
- Compression step eliminates unnecessary information from sentences
- Improvement in all areas

References

- Danushka Bollegala, Naoaki Okazaki and Mitsuru Ishizuka 2012. A Preference Learning Approach to Sentence Ordering for Multi document Summarization. Information Sciences 2012, 217:78-95.

- Dimitrios Galanis, Gerasimos Lampouras and Ion Androutsopoulos.2012. Extractive Multi-Document Summarization with Integer Linear Programming and Support Vector Regression. Proceedings of COLING 2012, 2012: 911-926

- Dan Gillick and Benoit Favre. 2009. A Scalable Global Model for Summarization. Proceedings of the NAACL HLT Workshop on Integer Linear Programming for Natural Language Processing, pages 1018, Boulder, Colorado, June 2009.

References

- John M.Conroy, Judith D.Schlesinger, Jade Goldstein and Dianne P.O'Leary 2004. Left-Brain/Right-Brain Multi-Document Summarization. Proceedings of DUC, 2004

- Kai Hong and Ani Nenkova 2014. Improving the Estimation of Word Importance for News Multi-Document Summarization. Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics 2004, 2004:712-721

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

-Ryan A.Georgi 2018. Lecture of Content Selection, Information Ordering and Content Realization. in the course LING 573 Spring 2018 in the University of Washington.