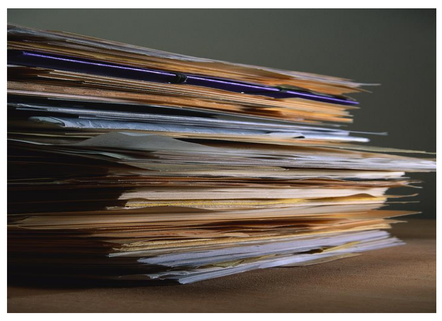
Multi-Document Summarization by Maximizing Informative Content-Words

Scott Wen-tau Yih Joshua Goodman Lucy Vanderwende Hisami Suzuki
Microsoft Research

Best result using a simple and principled method

Task: Multi-Document Summarization

Input: A set of similar documents



Example:
News articles about Hussein's death

Output: A short (e.g., ≤ 100 words) summary

> Former Iraq dictator, Saddam Hussein, was hanged for crimes during his rule. Official said his body would be buried in the "next few hours." After his execution, pictures of Saddam Hussein's body has been shown by Iraqi TV…

What is a good summary?

- The quality of a machine-generated summary is judged by its ***similarity*** to human summaries
- Automatic evaluation method – ROUGE scores
  - Unigram or bigram co-occurrences between machine summary and human summary
- **Principle:**
  - Find words from the original documents that are likely to appear in human summaries
  - Generate a summary that covers as many different ***important*** words as possible

Need to handle the (word) redundancy issue!

Estimate the scores of words

- Important information
  - Term frequency [Nenkova et al. SIGIR-06]
  - ***Word position***
- Incorporate different kinds of information
  - Discriminative scoring
    1. Encode information into *features*
    2. Predict the probability that the word will appear in human summaries using machine learning (e.g., logistic regression)
  - Generative scoring
    1. Estimate probabilities using frequencies of words in the beginning of the documents

Generate the "best" summary

- Objective:
  - Maximize the sum of word scores
    - We can't pick a bag of important words…
    - ***Select sentences instead!***
- Solve the knapsack problem
  (Ignore the redundancy issue)

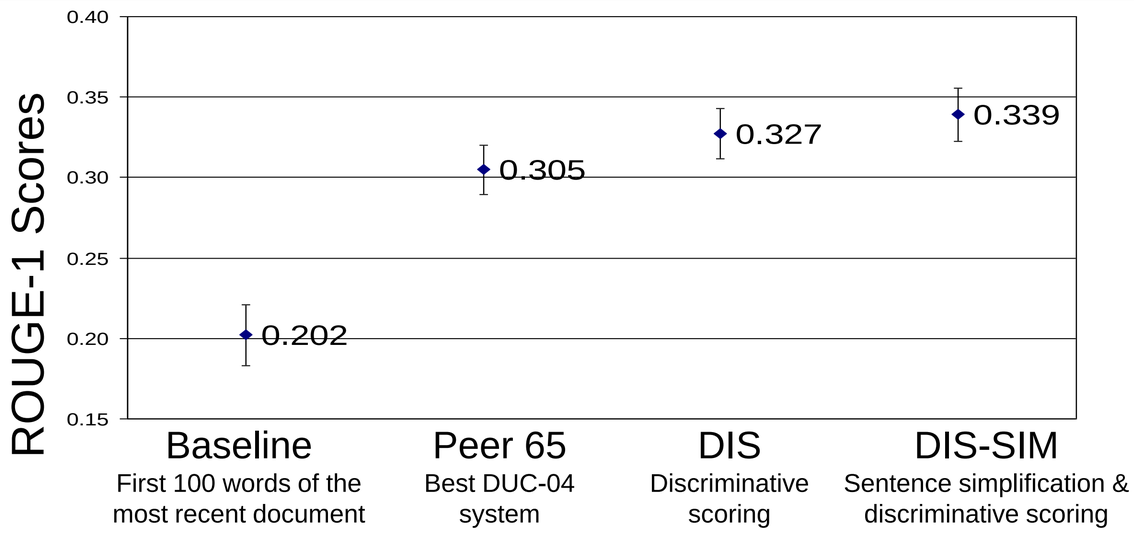
Sent. #1 (#words = 10, Σscores = 0.5)
Sent. #2 (#words = 8, Σscores = 0.45)
Sent. #3 (#words = 5, Σscores = 0.35)
Sent. #4 (#words = 12, Σscores = 0.6)
Sent. #5 (#words = 20, Σscores = 0.8)
…

- Maximize Σscores
- Cannot take more than 100 words

Handle the redundancy issue

- Use stack-decoder [Jelinek '69]
  - A dynamic algorithm
  - Penalize words that occur more than once in the search
  - Equivalent to solving the knapsack problem if ignoring the redundancy issue

Experimental results (DUC-04)



Sentence simplification can help

- Provide *simplified* sentences to select in addition to the original sentences
  - Eliminate syntactic units based on predefined heuristic templates
  - Goal: shorten the length but still keep the important words (information)

Conclusions

- Excellent results
  - Best ROUGE-1 for DUC-04 & MSE-05
  - Very simple method (~600 lines of code)
- Future Work
  - Task-focused summarization