D2 Summary

Sentence Selection Solution

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

Architecture: Implementation

Reader:

- Topic parser reads topics and generates filenames
- Document parser reads documents and makes document descriptors

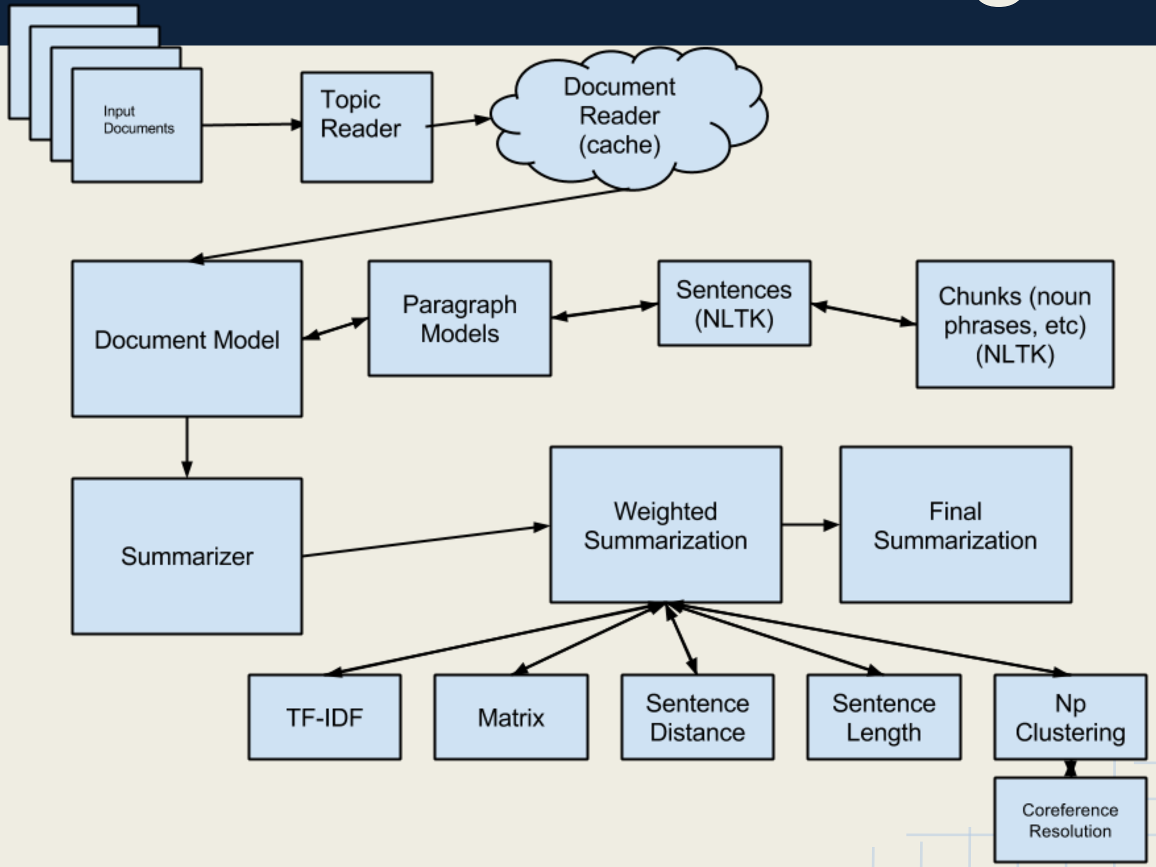
Document Model:

- Sentence Segmentation and “cleaning”
- Tokenization
- NP Chunker

Summarizer - creates summaries

Evaluator - uses pyrouge to call ROUGE-1.5.5.pl

Architecture: Block Diagram



Summarizer

Employed Several Techniques:

Each Technique:

- Computes rank for all sentences normalized from 0 to 1
- Is given a weight from 0 to 1

Weighted sentence rank scores are added together
Overall best sentences are selected from the summary sum

Summary Techniques

- Simple Graph Similarity Measure
- NP Clustering
- Sentence Location
- Sentence Length
- tf*idf

Trivial Techniques

- Sentence Position Ranking - Highest sentences get highest rank
- Sentence Length Ranking - Longest sentences get best rank
- tf*idf - All non-stop words get tf*idf computed and the total is divided by sentence length. Sentences with the highest sum of tf*idf get best rank.
  - We use the Reuters-21578, Distribution 1.0 Corpus of news articles as a background corpus.
  - Scores are scaled so the best score is 1.0

Simple Graph Technique

Iterate:

- Build a fully connected graph of the cosine similarity (non-stopword raw counts) of the sentences
- Compute the most connected sentence
- Give that sentence the highest score
- Change the weights of its edges to negative to discourage redundancy
- recompute

NP-Clustering Technique

Compute the most connected sentences:

- Use coreference resolution:
  - Find all the pronouns, and replace them with their antecedent
- Compare just the noun phrases of each sentence with every other sentence.
  - Use edit distance for minor forgiveness
  - Normalize casing
- Similarity metric is the count of shared noun phrases
- Rank every sentence with between 0-1, with the highest being 1

Technique Weighting

It is difficult to tell how important each technique is in contributing to the overall score. Because of this, we established a **weight generator** which did the following:

for each technique:

- compute unweighted sentence ranks.

- Iterate weights of each technique from 0 to 1 at intervals of 0.1
    - for each weight set:
        - rank sentences based on new weights
        - generate rouge scores

At the end, the best set of weights is the one with the optimal score!

Optimal Weights at Time of Submission

AAANNND... the optimal set of weights turns out to be:

Disappointing!

It looked like none of our fancy techniques were able to even slightly improve the performance of **tf*idf** by itself.

Results?

Average ROUGE scores for our tf*idf-only solution:

ROUGE Technique	Recall	Precision	F-Score
ROUGE1	0.55024	0.52418	0.53571
ROUGE2	0.44809	0.42604	0.43580
ROUGE3	0.38723	0.36788	0.37643
ROUGE4	0.33438	0.31742	0.32490

Results?

Obviously, we had done something wrong.   It's pretty unlikely that we got three times better than the best summarizers!   We figured out pretty quickly that it was our method of calling rouge, and reran our weight generator.

Optimal Weights Revisited

Hurray! Upon running again, discovered that our hard work had paid off after all! The NP-Clustering technique proved to be the best, followed closely by “equal weight” for every technique.

Optimal Weights

Optimal Technique Weights:

Technique	Weight
tf*idf	0.0
Simple Graph	0.0
NP-Clustering	1.0
Sentence Position	0.0
Sentence Length	0.0

NP-Clustering Results

Average ROUGE scores for the NP-Clustering-only solution:

ROUGE Technique	Recall	Precision	F-Score
ROUGE1	0.23391	0.28553	0.25522
ROUGE2	0.05736	0.07053	0.06272
ROUGE3	0.01612	0.01969	0.01758
ROUGE4	0.00533	0.00657	0.00584

Equal Weight Results

Average ROUGE scores for our “equal weight” solution:

ROUGE Technique	**Recall**	**Precision**	**F-Score**
ROUGE1	0.23336	0.28628	0.25516
ROUGE2	0.05708	0.07044	0.06251
ROUGE3	0.01612	0.01969	0.01758
ROUGE4	0.00533	0.00657	0.00584

Simple Graph Results

Average ROUGE scores for the Simple Graph-only solution:

ROUGE Technique	Recall	Precision	F-Score
ROUGE1	0.19379	0.25550	0.21845
ROUGE2	0.04473	0.05859	0.05033
ROUGE3	0.01170	0.01505	0.01305
ROUGE4	0.00362	0.00453	0.00400

tf*idf Only Results

Average ROUGE scores for our (tf*idf-only) solution:

ROUGE Technique	Recall	Precision	F-Score
ROUGE1	0.15341	0.20846	0.17522
ROUGE2	0.03014	0.04037	0.03426
ROUGE3	0.00746	0.01038	0.00863
ROUGE4	0.00242	0.00329	0.00278

Room for Improvement

- Our individual content selection techniques are simple, and much tuning and improvement remains to be done
  - Implement LLR and compare with tf*idf
  - Test other vector weighting schemes for cosine similarity in Simple Graph technique
  - Merge the Simple Graph style of redundancy reduction into NP Clustering technique
- Move coreference into document model so all content selection techniques and future ordering/realization techniques can take advantage of it

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