D3 Summary

Sentence Selection + Ordering 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

textrazor for entity extraction

attensity for entity and semantic information extraction

Architecture: Implementation

Reader - Extracts data from topic-focused document clusters

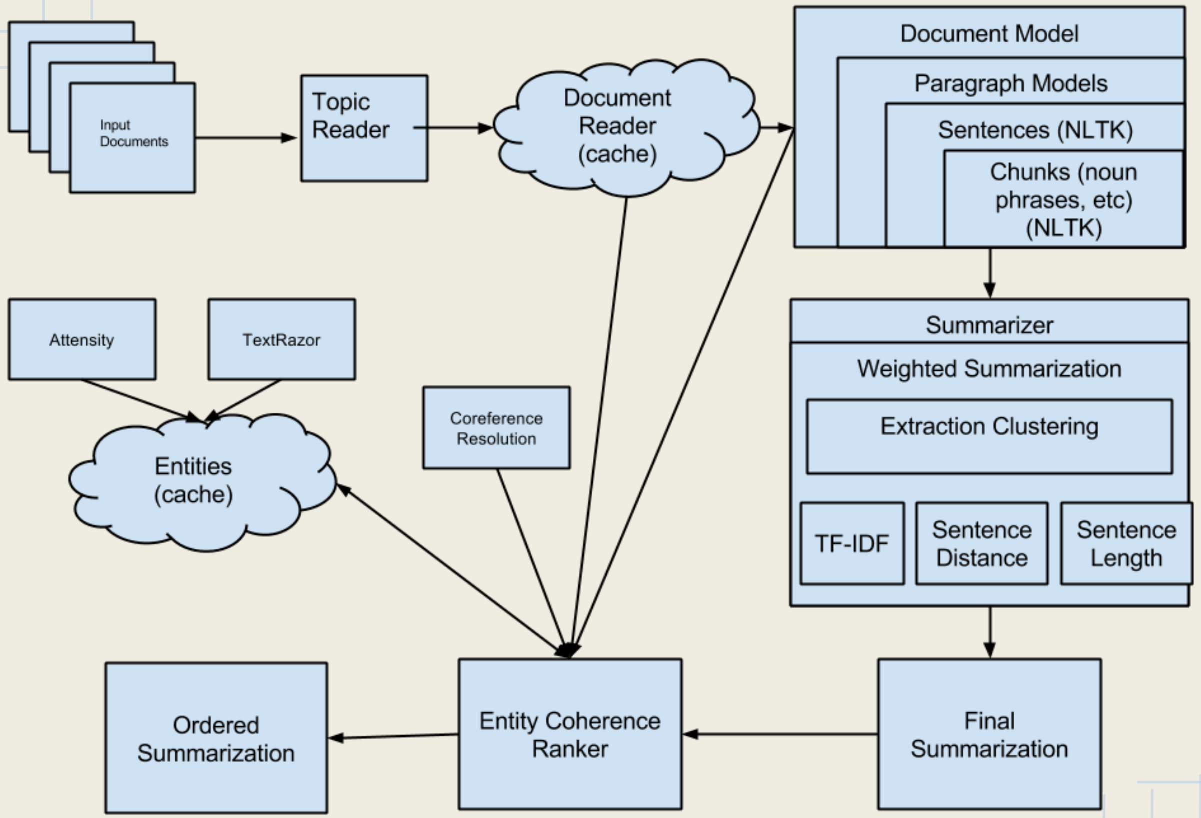
Document Model - Sentence Segmentation, Tokenization, NP Chunking

Summarizer - Creates summaries (using various techniques)

Evaluator - Uses pyrouge to call ROUGE-1.5.5.pl

Reorderer - Uses entity-coherence ranking to reorder

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 added.   Sentences with the highest sum of tf-idf get best rank.
  - We use the gigaword corpus as a background corpus.

Simple Graph Technique

Iterate:

- Find the common words of all sentences
- Compute the most connected sentence
- Mark that sentence as highest rank
- Change its value to negative
- 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
- Highest score sentences win
- Rank every sentence with normalized metric 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

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.

Optimal Weights

Optimal Technique Weights:

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

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

Optimal Weights Revisited

However: we found a bug in our weight generator, and, upon running again, discovered that our hard work had paid off after all! The Simple Graph technique DOES enhance our tf-idf only solution!

Optimal Weights Revisited

Optimal Technique Weights:

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

Best Results

Average ROUGE scores for our final (tf-idf + matrix) solution:

ROUGE Technique	**Recall**	**Precision**	**F-Score**
ROUGE1	0.54107	0.57388	0.55580
ROUGE2	0.42822	0.45443	0.43997
ROUGE3	0.36791	0.39088	0.37819
ROUGE4	0.31867	0.33882	0.32767

Simple Graph Results

Average ROUGE scores for the Simple Graph-only solution:

ROUGE Technique	**Recall**	**Precision**	**F-Score**
ROUGE1	0.48228	0.56860	0.52048
ROUGE2	0.36821	0.43541	0.39787
ROUGE3	0.31484	0.37348	0.34065
ROUGE4	0.27465	0.32683	0.29757

NP-Clustering Results

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

ROUGE Technique	Recall	Precision	F-Score
ROUGE1	0.45691	0.53378	0.49056
ROUGE2	0.33306	0.39053	0.35813
ROUGE3	0.28221	0.33196	0.30386
ROUGE4	0.24758	0.29237	0.26700

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