Text Segmentation and Summarization

Siavash Kazemian

Based on
Tutorial Notes by Cosmin Munteanu

C. Manning and H. Schutze Foundations of Statistical Natural Language Processing
M. A. Hearst, TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages

Text Segmentation and Summarization

- Overview of Assignment 3
  - Purpose
  - Tasks
  - Corpus
- Text Segmentation
  - Applications and motivation
  - TextTiling Algorithm
    - Tokenization
    - Similarity determination
    - Segment breakpoint identification
- Text Summarization
  - Paragraph Salience
  - Singular Value Decomposition (SVD)

Overview of Assignment 3

- Your Data:
  - Text files containing several multi-paragraph articles concatenated together. Each article is about a certain topic
  - In this assignment your articles are news articles from the British National Corpus

- Your Purpose:
  - Break the text files into segments such that each segment contains one article
  - Write code that will generate a summary of each segment that you have identified in the previous step

Task Breakdown in Assignment 3

- Implement the TextTilling Algorithm that breaks each file into segments
- Test it on 10 files (containing 185 news articles)
- Evaluate your segmentation algorithm by comparing your results with the true article break points (labeled in your data)
- For each segment, determine the most salient paragraph (that will stand as its summary)
- Evaluation: Examine your generated summaries, comment on their quality and explain why your algorithm came up with them

Text Segmentation: Motivations and Applications

– Information Retrieval:
  - Performing query similarity measures against a sections of document as supposed to the whole document
  - When displaying the search result, you can display the most relevant portion of the document to the query. Similar to the way Google displays its search results

– Processing News Streams: Segmenting Stream into Stories

Text Segmentation: Motivations and Applications

- Discourse Analysis: Detecting Topic Shifts (changes)

- Text summarization: Breaking a document into sections before summarizing. This will ensure that the summary includes all the topics that were covered in the document.

Text Segmentation using TextTiling

- M. A. Hearst, TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages
- TextTiling is an algorithm for breaking documents into topically coherent multi-paragraph subparts
- Identifies parts of text where the vocabulary shifts – Degree of terms repetition is low in each side of the boundary
- Three phases:
  - Tokenization
  - Similarity Determination
  - Segment breakpoint identification

Tokenization

- Sentence Boundaries are not relevant
- Tiles: Equal sized areas of text : sequences of words that contain the same number of words
- Phases:
  1. Convert text into lower-cases
  2. Remove punctuation (including end of sentence boundaries)
  3. Remove numbers and non-alphabetic symbols
  4. Keep the paragraph breaks
  5. Mark stop-words (no-deletion)
  6. Remaining tokens are stemmed
  7. Divide the text into token sequences of n tokens: Tiles. Usually n=20
- Example: "He **Drives** a **car** or a **taxi**." → "stopword **driv** stopword stopword **car** stopword stopword **taxi**"

Similarity Measures

- Measure how similar two blocks around a potential segment break (gap) are

- One block: k token sequences

- Three possible measures:

  Vector Space Model: create two artificial documents from the sequence of tokens at the left and the right of the gap; compute correlation coefficients for the documents (Using their term vectors)

  Vocabulary Introduction: Similarity is measured as the negative of the number of new terms introduced on either side of the gap

  Block Comparison: compute correlation coefficients between left and right blocks based on within-block term frequency (without inverse document frequency) (We will use this method)

Similarity Measure – Block Comparison

- A "bag of words" approach
- For a gap *i*:

$$sim(i) = \frac{\sum_{t \in T} C(t, b_l) \cdot C(t, b_r)}{\sqrt{\sum_{t \in T} C(t, b_l)^2 \sum_{t \in T} C(t, b_r)^2}}$$

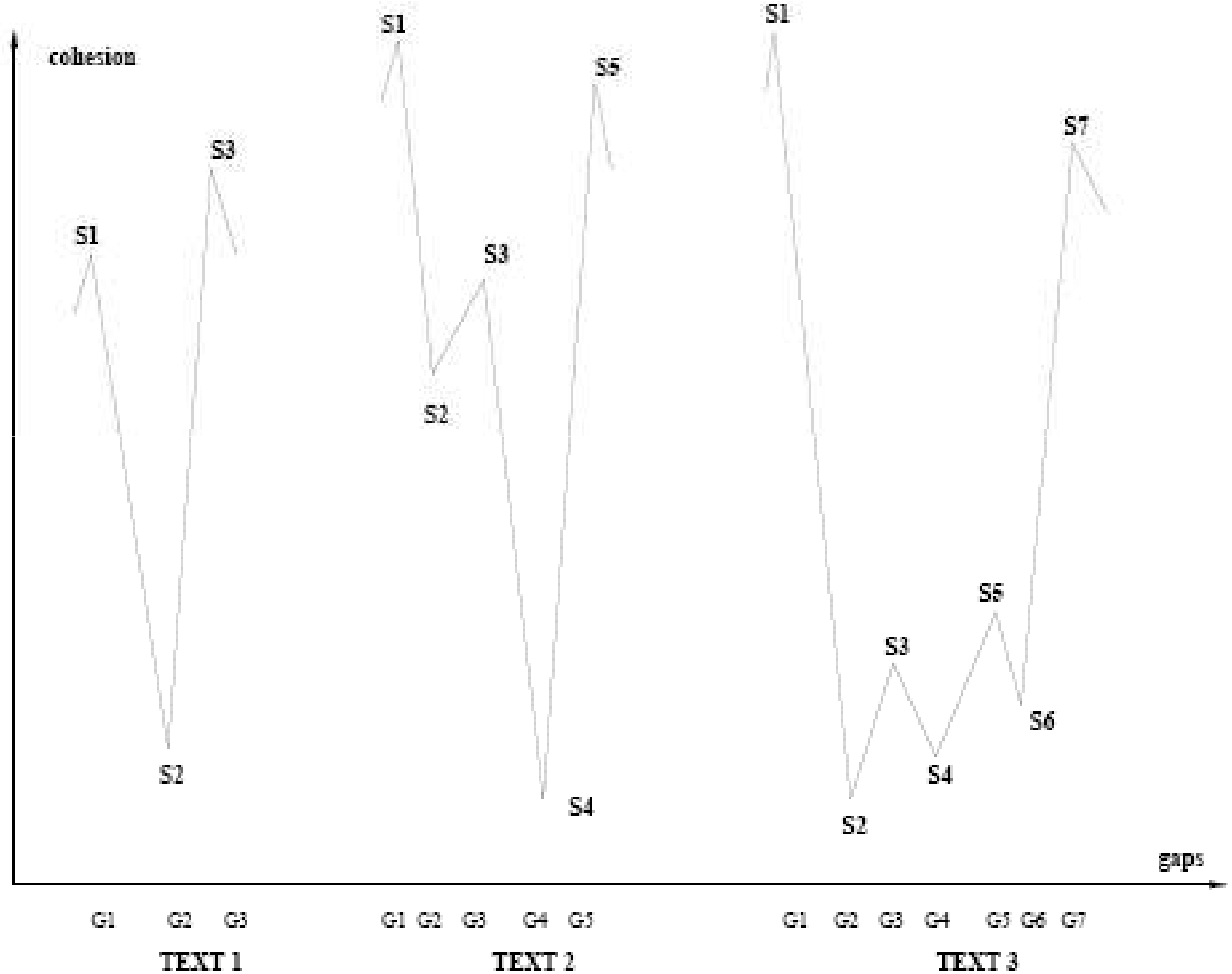
- Where:
  - T is the set of non-stop list terms in both blocks b_l and b_r
  - For a token $t \in T$, $C(a,b)$ is the number of occurrences of t in b
- If $sim(i) = 0/0$, then assign:
  - $sim(i) = 1$ (highly similar) if both $C(t, b_l) = C(t, b_r) = 0$
  - $sim(i) = 0$ if only one of $C(t, b_l)$ or $C(t, b_r)$ is 0

Depth Scoring

- Similarity measure is relative and so, can not be used to determine the breakpoints (segment boundries)
- Successive token sequences can have:
  - Low similarity (also called cohesion scores) ex: Introduction
  - High cohesion score (only slight shift in vocabulary over large areas of text)
- Need to observe how similarity scores change (i.e. compare the difference in similarity scores not their absolute value)
- Depth(i) = sim(i-1) – sim(i) + sim(i+1) – sim(i)
- Smoothing (applied twice) to filter out the noise:

$$depth(i) = \frac{depth(i-1) + depth(i) + depth(i+1)}{3}$$

Depth Scoring Example



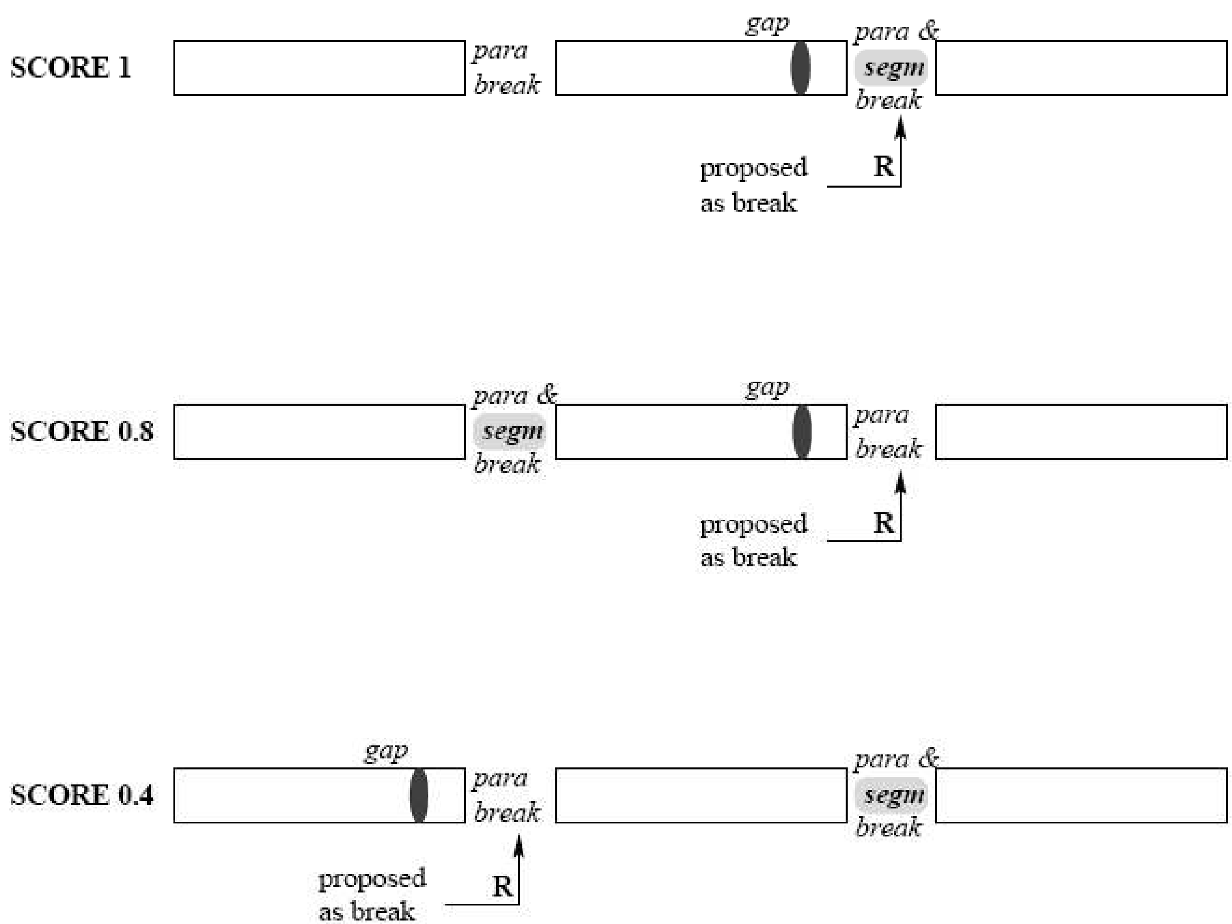
Boundary Identification

- Two methods to choose gaps with big depth scores as breakpoints
  - Compute the mean and standard deviation of depth scores and select gaps with depth scores biger than $\mu - c\sigma$ (depending on the data, usually we pick c=0.5 or c=1.0)
  - Estimate the number of breakpoints from data: D and pick the D gaps with biggest depths (We'll use this method)
- Token sequences are of predefined length → proposed break points could end up in the middle of a paragraph. In this case, mark the closest paragraph break as a breakpoint (mark **L** or **R** if paragraph break is to the left or right of the gap). Note: If a paragraph break is marked both with **R** and **L**, you are more confidant that it's a breakpoint.
- Discard the breakpoints at occur in consecutive paragraph breaks. But do ensure that at the end there are D breakpoints

Evaluation

- Use precision and Recall (gold standard: the original breakpoint marks on your data)
- Methods:
  - **Strict**: If breakpoint is correctly marked with R or L
  - **Relaxed**:
    - Score of 1 if the breakpoint is marked properly
    - Score 0.8 if the breakpoint is within one paragraph: The proposed breakpoint is marked with L and is to the left of the true breakpoint or marked with R and is to the right of the breakpoint
    - Score 0.4 if the breakpoint is within two paragraphs. The proposed breakpoint is marked with R and is to the left of the true breakpoint or marked with L and is to the right of the true breakpoint
  - **Very Relaxed**: mark all three cases above with 1.0

Evaluation Example



SEGMENT SUMMARIZATION

Paragraph Salience

- Motivation: Automatic summarization by sentence/paragraph extraction
- Goal: Identify the most significant (salient) paragraph in the segment
- Task: Compute the Salient score for each paragraph for each segment through SVD

SVD for Salient Scores

- SVD: Method for dimensionality reduction
- Dimensions of your data matrix A– terms by paragraphs
- Matrix A: rows → non-stoplist terms
  columns → paragraphs
- a_{ij} = $\text{count}_{\text{paragraph } j}$ (term i)

- Matrix A:
  - **Total method**: counts collected over all files S01 … S10
  - **Tile method**: represent term frequencies of terms in paragraphs from a single segment. (number of paragraphs and terms much smaller than in the total method) Note: In Tile method, number of rows = number of terms in a segment

- Decompose A through SVD: $A = U . S . V^T$

Voting Protocols

- use matrices S and V
- choose s = 1 . . . n
- $\tilde{S}$ = first s rows of S
- $\tilde{V}$ = first s columns of V
- for a paragraph p:

summing:

$$sum_p = \sum_i |(\tilde{S} \cdot \tilde{V}_p^T)_i|$$

maxing:

$$max_p = \sum_{j:p=\underset{i}{argmax}|V_{ij}|} S_{jj}$$

Voting Protocols – Summing

$$S = \begin{pmatrix} 2.16 & 0 & 0 & 0 & 0 \\ 0 & 1.59 & 0 & 0 & 0 \\ 0 & 0 & 1.28 & 0 & 0 \\ 0 & 0 & 0 & 1.00 & 0 \\ 0 & 0 & 0 & 0 & 0.39 \end{pmatrix}$$

$$V = \begin{pmatrix} -0.75 & -0.29 & 0.28 & 0 & -0.53 \\ -0.28 & -0.53 & -0.75 & 0 & 0.29 \\ -0.20 & -0.19 & 0.45 & 0.58 & 0.63 \\ -0.45 & 0.63 & -0.20 & 0 & 0.19 \\ -0.33 & 0.22 & 0.12 & -0.58 & 0.41 \end{pmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{matrix}$$

summing: $(s = 5)$

$$\tilde{S} \cdot \begin{pmatrix} -0.75 & -0.29 & 0.28 & 0 & -0.53 \end{pmatrix}^T = \begin{pmatrix} -1.62 & -0.46 & 0.35 & 0 & 0.2 \end{pmatrix}^T$$

$$sum_{p_1} = 1.62 + 0.46 + 0.35 + 0 + 0.2 = 2.6462$$

Voting Protocols - maxing

$$S = \begin{pmatrix} 2.16 & 0 & 0 & 0 & 0 \\ 0 & 1.59 & 0 & 0 & 0 \\ 0 & 0 & 1.28 & 0 & 0 \\ 0 & 0 & 0 & 1.00 & 0 \\ 0 & 0 & 0 & 0 & 0.39 \end{pmatrix}$$

$$V = \begin{pmatrix} \boxed{-0.75} & -0.29 & 0.28 & 0 & -0.53 \\ -0.28 & -0.53 & \boxed{-0.75} & 0 & 0.29 \\ -0.20 & -0.19 & 0.45 & \boxed{0.58} & \boxed{0.63} \\ -0.45 & \boxed{0.63} & -0.20 & 0 & 0.19 \\ -0.33 & 0.22 & 0.12 & -0.58 & 0.41 \end{pmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \end{matrix}$$

maxing: $(s = 5)$

$$max_{p_1} = 2.16$$
$$max_{p_2} = 1.28$$
$$max_{p_3} = 1.00 + 0.39 = 1.39$$
$$max_{p_4} = 1.59$$

Summing vs. Maxing

$$S = \begin{pmatrix} 2.1 & 0 & 0 \\ 0 & 1.7 & 0 \\ 0 & 0 & 1.3 \end{pmatrix} \qquad V = \begin{pmatrix} 1.1 & 2.2 & 1.6 \\ 0.7 & 3.1 & 2.5 \\ 0 & 0.2 & 0 \end{pmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \end{matrix}$$

$$s = 2$$

$$sum_{p_1} =$$
$$sum_{p_2} =$$

$\Rightarrow$ Winner:

$$max_{p_1} =$$
$$max_{p_2} =$$

$\Rightarrow$ Winner:

Summing vs. Maxing

$$S = \begin{pmatrix} 2.1 & 0 & 0 \\ 0 & 1.7 & 0 \\ 0 & 0 & 1.3 \end{pmatrix} \qquad V = \begin{pmatrix} 1.1 & 2.2 & 1.6 \\ 0.7 & 3.1 & 2.5 \\ 0 & 0.2 & 0 \end{pmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \end{matrix}$$

$$s = 2$$

$$\begin{aligned} sum_{p_1} &= 1.1 \cdot 2.1 + 2.2 \cdot 1.7 = 6.05 \\ sum_{p_2} &= 0.7 \cdot 2.1 + 3.1 \cdot 1.7 = 6.74 \end{aligned}$$

$\Rightarrow$ Winner: p_2

$$\begin{aligned} max_{p_1} &= \\ max_{p_2} &= \end{aligned}$$

$\Rightarrow$ Winner:

Summing vs. Maxing

$$S = \begin{pmatrix} 2.1 & 0 & 0 \\ 0 & 1.7 & 0 \\ 0 & 0 & 1.3 \end{pmatrix} \qquad V = \begin{pmatrix} 1.1 & 2.2 & 1.6 \\ 0.7 & 3.1 & 2.5 \\ 0 & 0.2 & 0 \end{pmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \end{matrix}$$

$$s = 2$$

$$\begin{aligned} sum_{p_1} &= 1.1 \cdot 2.1 + 2.2 \cdot 1.7 = 6.05 \\ sum_{p_2} &= 0.7 \cdot 2.1 + 3.1 \cdot 1.7 = 6.74 \end{aligned}$$

$\Rightarrow$ Winner: p_2

$$\begin{aligned} max_{p_1} &= 2.1 \\ max_{p_2} &= 1.7 \end{aligned}$$

$\Rightarrow$ Winner: p_1

Changing s

$$S = \begin{pmatrix} 2.1 & 0 & 0 \\ 0 & 1.7 & 0 \\ 0 & 0 & 1.3 \end{pmatrix} \qquad V = \begin{pmatrix} 1.1 & 2.2 & 1.6 \\ 0.7 & 3.1 & 2.5 \\ 0 & 0.2 & 0 \end{pmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \end{matrix}$$

$s = 1 \Rightarrow$ winner is

$$\begin{aligned} sum_{p_1} &= \\ sum_{p_2} &= \\ max_{p_1} &= \\ max_{p_2} &= \end{aligned}$$

$s = 3 \Rightarrow$ winner is

$$\begin{aligned} sum_{p_1} &= \\ sum_{p_2} &= \\ sum_{p_3} &= \\ max_{p_1} &= \\ max_{p_2} &= \\ max_{p_1} &= \end{aligned}$$

Changing s

$$S = \begin{pmatrix} 2.1 & 0 & 0 \\ 0 & 1.7 & 0 \\ 0 & 0 & 1.3 \end{pmatrix} \qquad V = \begin{pmatrix} 1.1 & 2.2 & 1.6 \\ 0.7 & 3.1 & 2.5 \\ 0 & 0.2 & 0 \end{pmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \end{matrix}$$

$s = 1 \Rightarrow$ winner is p_1

$$\begin{aligned} sum_{p_1} &= 1.1 \cdot 2.1 = 2.31 \\ sum_{p_2} &= 0.7 \cdot 2.1 = 1.47 \\ max_{p_1} &= 2.1 \\ max_{p_2} &= 0 \end{aligned}$$

$s = 3 \Rightarrow$ winner is

$$\begin{aligned} sum_{p_1} &= \\ sum_{p_2} &= \\ sum_{p_3} &= \\ max_{p_1} &= \\ max_{p_2} &= \\ max_{p_1} &= \end{aligned}$$

Changing s

$$S = \begin{pmatrix} 2.1 & 0 & 0 \\ 0 & 1.7 & 0 \\ 0 & 0 & 1.3 \end{pmatrix} \qquad V = \begin{pmatrix} 1.1 & 2.2 & 1.6 \\ 0.7 & 3.1 & 2.5 \\ 0 & 0.2 & 0 \end{pmatrix} \begin{matrix} p_1 \\ p_2 \\ p_3 \end{matrix}$$

$s = 1 \Rightarrow$ winner is p_1

$$\begin{aligned} sum_{p_1} &= 1.1 \cdot 2.1 = 2.31 \\ sum_{p_2} &= 0.7 \cdot 2.1 = 1.47 \\ max_{p_1} &= 2.1 \\ max_{p_2} &= 0 \end{aligned}$$

$s = 3 \Rightarrow$ winner is p_2

$$\begin{aligned} sum_{p_1} &= 1.1 \cdot 2.1 + 2.2 \cdot 1.7 + 1.6 \cdot 1.3 = 8.13 \\ sum_{p_2} &= 0.7 \cdot 2.1 + 3.1 \cdot 1.7 + 2.5 \cdot 1.3 = 9.99 \\ sum_{p_3} &= 0 \cdot 2.1 + 0.2 \cdot 1.7 + 0 \cdot 1.3 = 0.34 \\ max_{p_1} &= 2.1 \\ max_{p_2} &= 1.7 + 1.3 = 3.0 \\ max_{p_1} &= 0 \end{aligned}$$

Your Tasks

- Total matrix with summing
- Total matrix with maxing
- Tile matrices with summing
- Tile matrices with maxing
- Each for your best choice of s

Evaluation

- Automatic summarization – no gold standard evaluation
- Choose the best s for each voting protocol and matrix dimension
- Decision: based on your own judgment
- Important information to include:
  - Your methodology for choosing s
  - Arguments supporting your decision
  - Discussion