Automatic Key Word and Key Sentence Extraction

Yongxin Huang, Chunlan Ma

Agenda

- Background
- Rank-1 Approximation

  saliency score

  mutual reinforcement principle

  weekness
- Rank-k Approximation

  extraction of 1st key sentence

  extraction of 2nd key sentence

  extraction of 2nd key sentence
- Summary

Automatic Key Word and Key Sentence Extraction

Why:

- the explosion of the amount of textual information
- a need to develop automatic procedures for text summarization

How:

- Saliency Score
- Rank-k Approximation

Goal :

- Extract the most important content of a text(key words and key sentences)

Preprocessing

1. Stemming

    eats, eating, ate --> eat

2. Stop-words removing
3. Special symbolic removing

    mathematics or mark-up language tags (HTML, LATEX)

Text Presentation

- consider each sentence as a separate document
- a term is a stemmed word
- term-sentence Matrix A ∈ $\mathbb{R}^{m*n}$

  m: the number of different terms

  n: the number of sentences

	sentence 1	sentence 2	sentence 3	…	sentence n
apple	5.25	3.18	0	…	0
bill	1.21	0	2.54	…	0.88
…	…	…	…	…	…
search	0	1.37	2.28	…	1.59

Term-Sentence Matrix

- Term-Document(term-sentence) Matrix A ∈ $\mathbb{R}^{m*n}$

  m: the number of different terms

  n: the number of sentences
- a_{ij}: the weighted frequency of term i in sentence j
- column vector $(a_{1j}, \ a_{2j}, \cdots\cdots a_{mj})^T$: the terms occurring in sentence j
- row vector $(a_{i1}, \ a_{i2}, \cdots\cdots a_{in})$: sentences containing term i

	sentence 1	…	sentence j	…	sentence n
term 1	5.25	…	$\boldsymbol{a_{1j}}$	…	$\boldsymbol{a_{1n}}$
…	1.21	…	$\boldsymbol{a_{2j}}$	…	$\boldsymbol{a_{2n}}$
term i	$\boldsymbol{a_{i1}}$	…	$\boldsymbol{a_{ij}}$	…	$\boldsymbol{a_{in}}$
…	…	…	…	…	…
term m	0	…	$\boldsymbol{a_{mj}}$	…	$\boldsymbol{a_{mn}}$

Matrix SVD Recap

- A = $\mathbf{U} \sum V^T$

□columns of U and V^T are singular vectors

□diagonal elements of $\sum$ are singular values σ_i

□Av_i=$\sigma_i u_i$

□$A^T u_i$=$\sigma_i v_i$

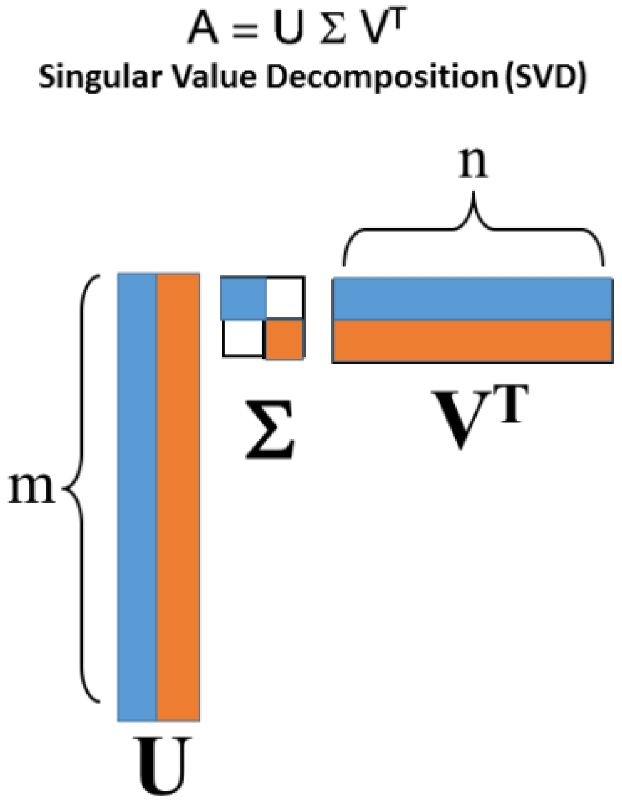
□A $A^T u_i$ = $\sigma_i \sigma_i u_i$

□A^T Av_i = σ_i $\sigma_i v_i$

Eigenvectors of A A^T are the singular vectors u_i

Eigenvectors of A^T A are the singular vectors v_i

AAT and ATA have the same eigenvalues of the form σ_i^2



$\boxed{\mathbf{A}_{[m \times n]} = \mathbf{U}_{[m \times r]}\, \Sigma_{[r \times r]}\, (\mathbf{V}_{[n \times r]})^T}$

Mutual reinforcement principle

☐ Saliency Scores

- u_i : saliency score of term i
- v_j : saliency score of sentence j

☐ mutual reinforcement principle

- A term should have a high saliency score if it appears in many sentences with high saliency scores.

$u_i \propto \sum_{j=1}^{n} a_{ij} v_j,\quad i = 1, 2, 3, \cdots\cdots m.$

- A sentence should have a high saliency score if it contains many words with high saliency scores.

$v_j \propto \sum_{i=1}^{m} a_{ij} u_i,\quad j = 1, 2, 3, \cdots\cdots n.$

reformed as:

$$\sigma_u \, u = Av$$

$$\sigma_v v = A^T u$$

σ_u and σ_v are proportionality constants

Equation Derivation

σ_u u = Av

σ_vv = A^Tu

insert one equation to the other, we get:

σ_u u = $\frac{1}{\sigma_v}$AA^Tu

σ_v v = $\frac{1}{\sigma_u}$ A^TAv

these shows that u and v are eigenvectors of A A^T and A^T A respectively, with the same eigenvalue σ_u σ_v

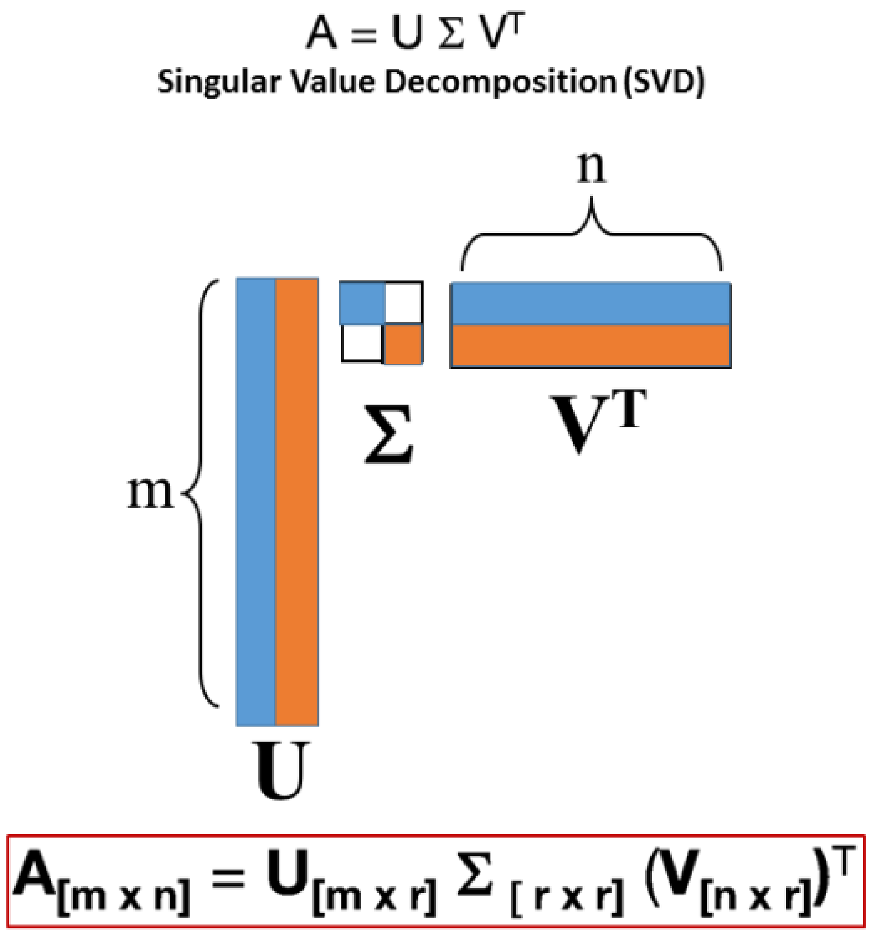
Matrix SVD Recap

- A = $\mathbf{U} \sum V^T$
  - columns of U and V^T are singular vectors
  - diagonal elements of $\sum$ are singular values σ_i
  - Av_i=$\sigma_i u_i$
  - $A^T u_i$=$\sigma_i v_i$
  - A $A^T u_i = \sigma_i \sigma_i u_i$
  - A^T A$\mathrm{v}_i = \sigma_i\ \sigma_i v_i$

Eigenvectors of A A^T are the singular vectors u_i

Eigenvectors of A^T Aare the singular vectors v_i

A A^T and A^T A have the same eigenvalues of the form σ_i^2



Equation Derivation

σ_u u = Av

σ_vv = A^Tu

insert one equation to the other, we get:

σ_u u = $\frac{1}{\sigma_v}$AA^Tu

σ_v v = $\frac{1}{\sigma_u}$$A^T$Av

these shows that u and v are eigenvectors of A A^T and A^T A respectively, with the same eigenvalue σ_u σ_v

reform as：

σ_v σ_u u = AA^Tu

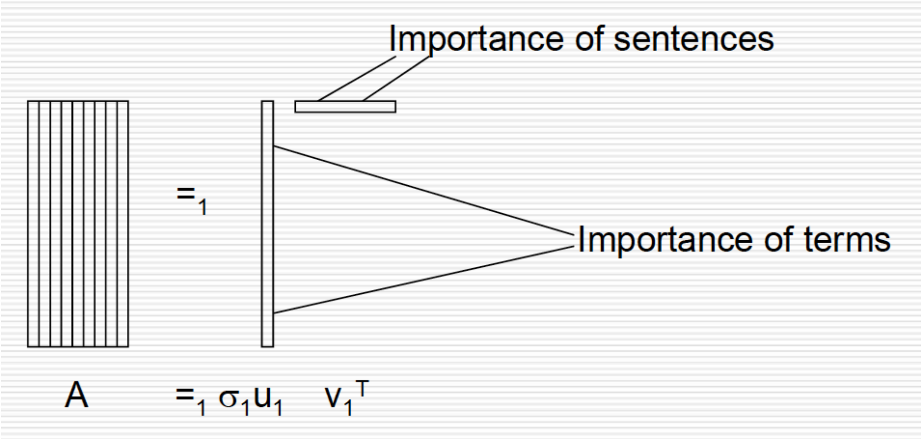
σ_u σ_v v = A^TAv

→ u and v are singular vectors corresponding to the same singular value σ_v σ_u

Saliency Score-Rank 1 Approximation

For some singular value si of SVD of A

- vector u of saliency scores for terms is the singular vector u_i
- vector v of saliency scores for sentences is the singular vector v_i
- $\sigma_\mathbf{v}\ \sigma_\mathbf{u} = \sigma_i\ \sigma_i$
- We select the largest singular value σ_1



The saliency scores for

- terms are the entries of the singular vector u_1
- sentences are the entries of the singular vector v_1
- for the largest singular value σ_1
- $A \approx \sigma_1 u_1 v_1^T$

From Rank-1 Approximation to Rank-k Approximation

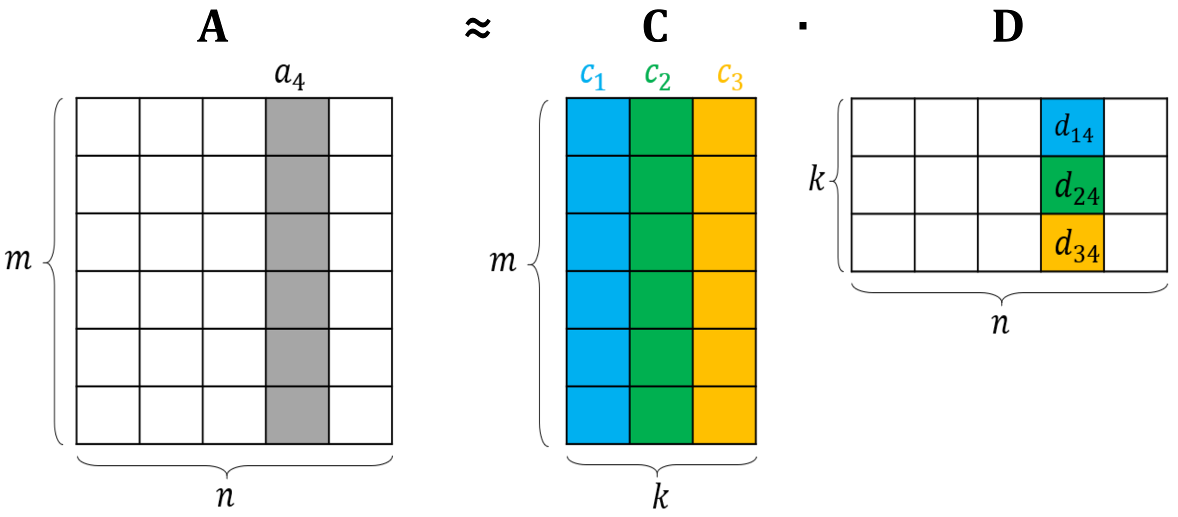
□Problems of rank-1 approximation:

- mutual reinforcement principle prefers long sentences
- consider only the most important (semantic) direction in “sentence space”
- cannot avoid extracting similar sentences

➢Solution: use rank-k approximation to extract key sentences

Key Sentence Extraction from a Rank-k Approximation

- Assume we have computed a rank-k approximation of $\boldsymbol{A}$: $\boldsymbol{A} \approx \boldsymbol{CD}$ (using SVD, clustering, or nonnegative matrix factorization)
- $\boldsymbol{C}$ can be seen as **k basis vectors** representing the most important (semantic) directions in the “sentence space”. (k ≥ number of key sentences)
- $\boldsymbol{D}$ holds **coordinates** of corresponding columns in A with regards to the basis vectors. （weights of semantic directions）
- So each column (sentence) in $\boldsymbol{A}$ is a linear combination of basis vectors in $\boldsymbol{C}$.

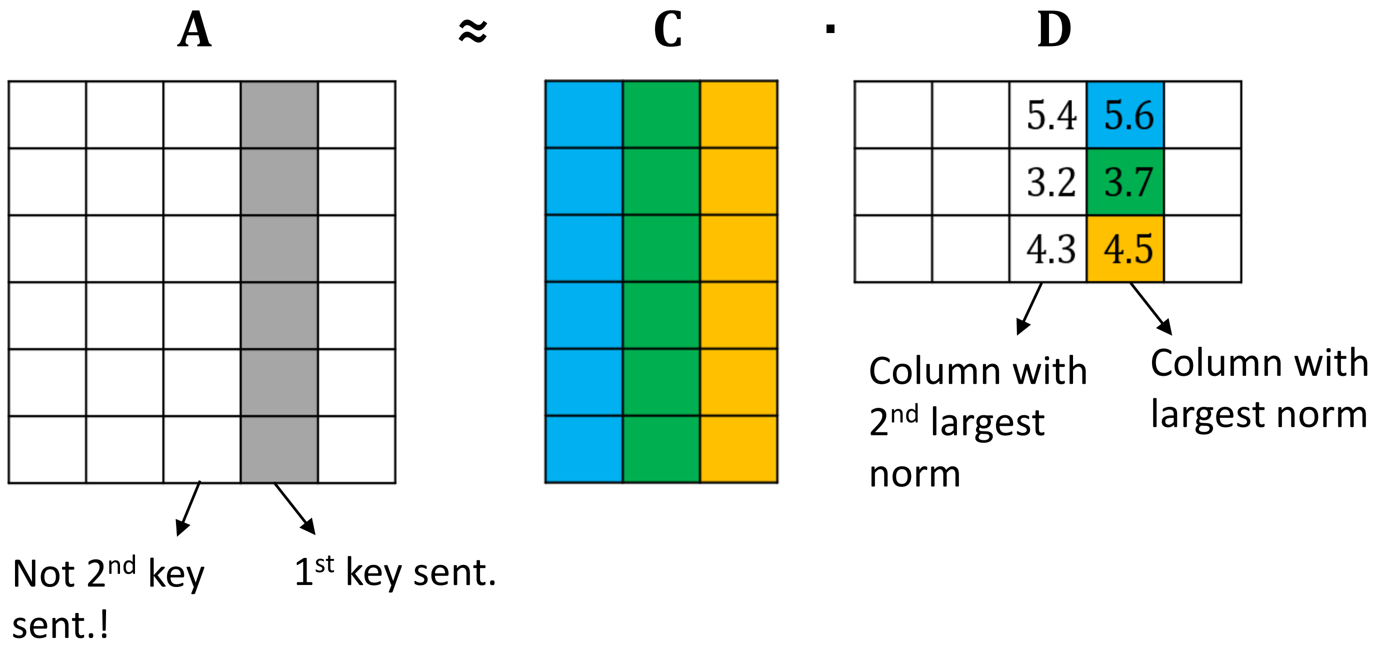


$$m = 6, n = 5, k = 3$$

$$a_4 = d_{14}c_1 + d_{24}c_2 + d_{34}c_3$$

Key Sentence Extraction from a Rank-k Approximation

- Find the most important column in $\boldsymbol{A}$ by looking for the column in $\boldsymbol{D}$ with largest 2-norm. (largest "weights" with regards to basis $\boldsymbol{C}$)
- But! To find the second key sentence, we cannot simply find the second largest column in $\boldsymbol{D}$. Because it may be similar to the first one.
- To avoid similarity, we cannot extract k sentences directly. we must apply some transformation before we determine each key sentence.



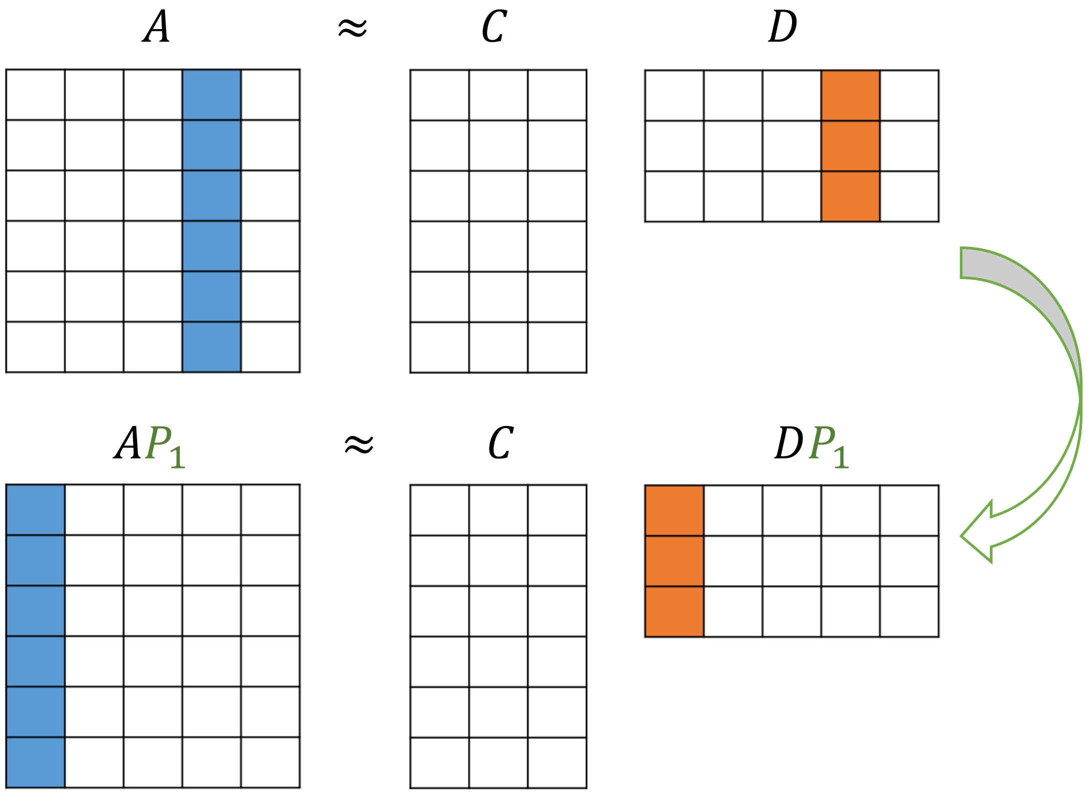
2-norm of a vector:

$$\|d_4\|_2 = \sqrt{d_{14}^2 + d_{24}^2 + d_{34}^2}$$

Extraction of 1st key sentence

Step 1: Permutation

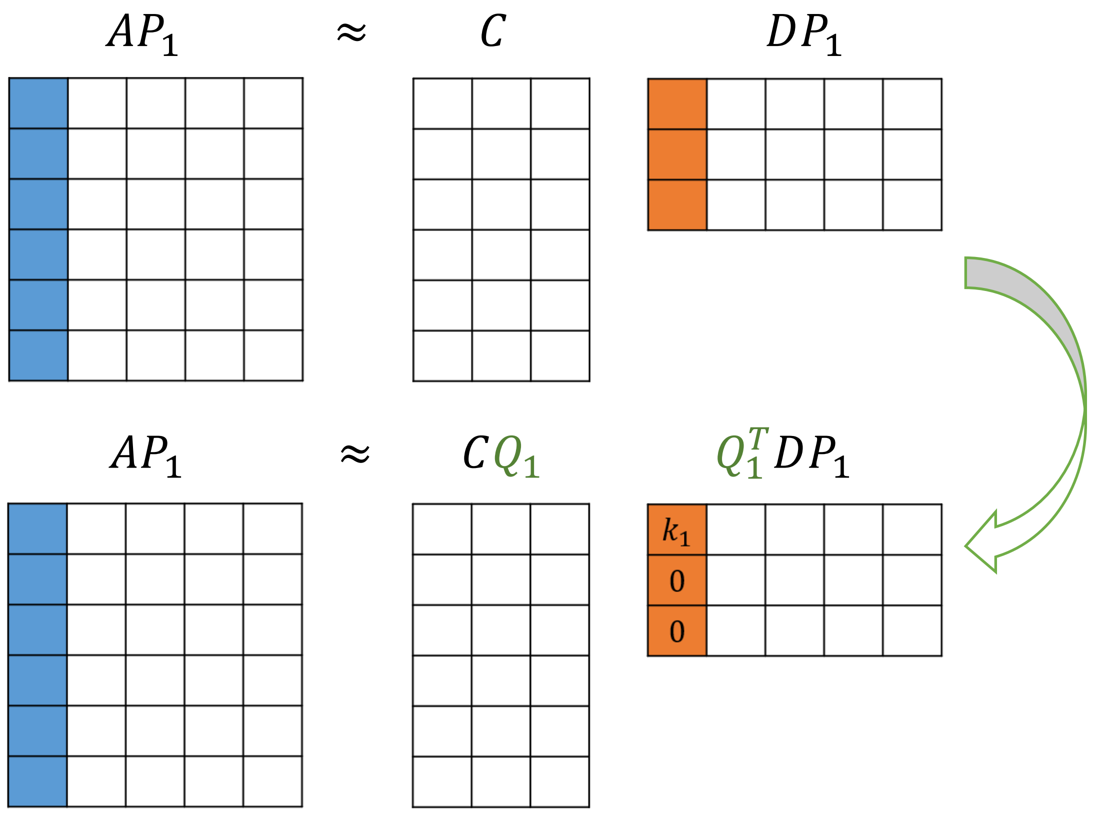
- Permute column in $\boldsymbol{D}$ with largest 2-norm to first position
- By applying a permutation $\boldsymbol{P_1}$ to $\boldsymbol{D}$ (multiply $\boldsymbol{D}$ by $\boldsymbol{P_1}$)
- To keep the equation balanced, $\boldsymbol{P_1}$ is also applied to $\boldsymbol{A}$.
- The corresponding column in $\boldsymbol{A}$ is also moved to first position.



Extraction of 1st key sentence

Step 2: Householder Transformation

- Zeros the elements in the first column below the element in position *(1,1)* in $\boldsymbol{DP_1}$
- By applying a Householder Transformation $\boldsymbol{Q_1^T}$ to $\boldsymbol{DP_1}$
- Then multiply $\boldsymbol{C}$ by $\boldsymbol{Q_1}$, so that the equation $AP_1 \approx CQ_1Q_1^T DP_1$ remains true.
- Because $Q_1Q_1^T = I$ for a householder transformation

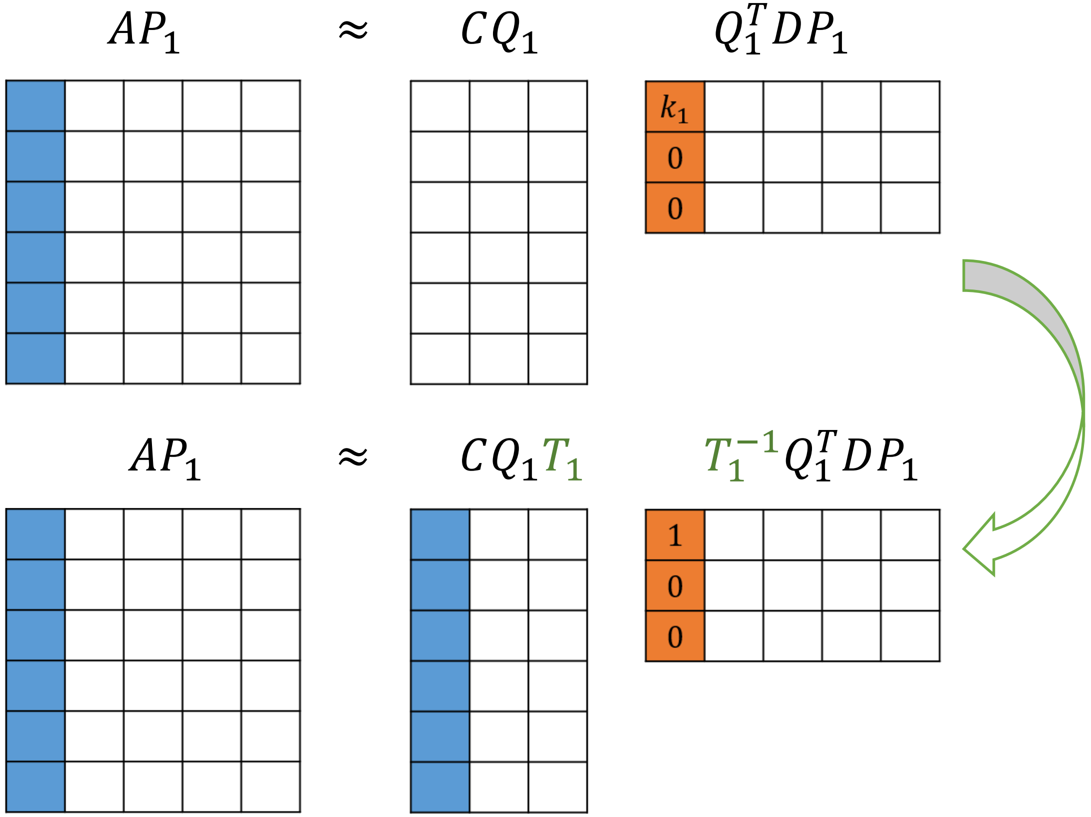


Extraction of 1st key sentence

Step 3: Scaling

- Normalize the first column in $\boldsymbol{Q_1^T D P_1}$ by applying transformation $\boldsymbol{T_1^{-1}}$
- The multiply $\boldsymbol{CQ_1}$ by $\boldsymbol{T_1}$, so that the equation remains true. $(T_1 T_1^{-1} = I)$

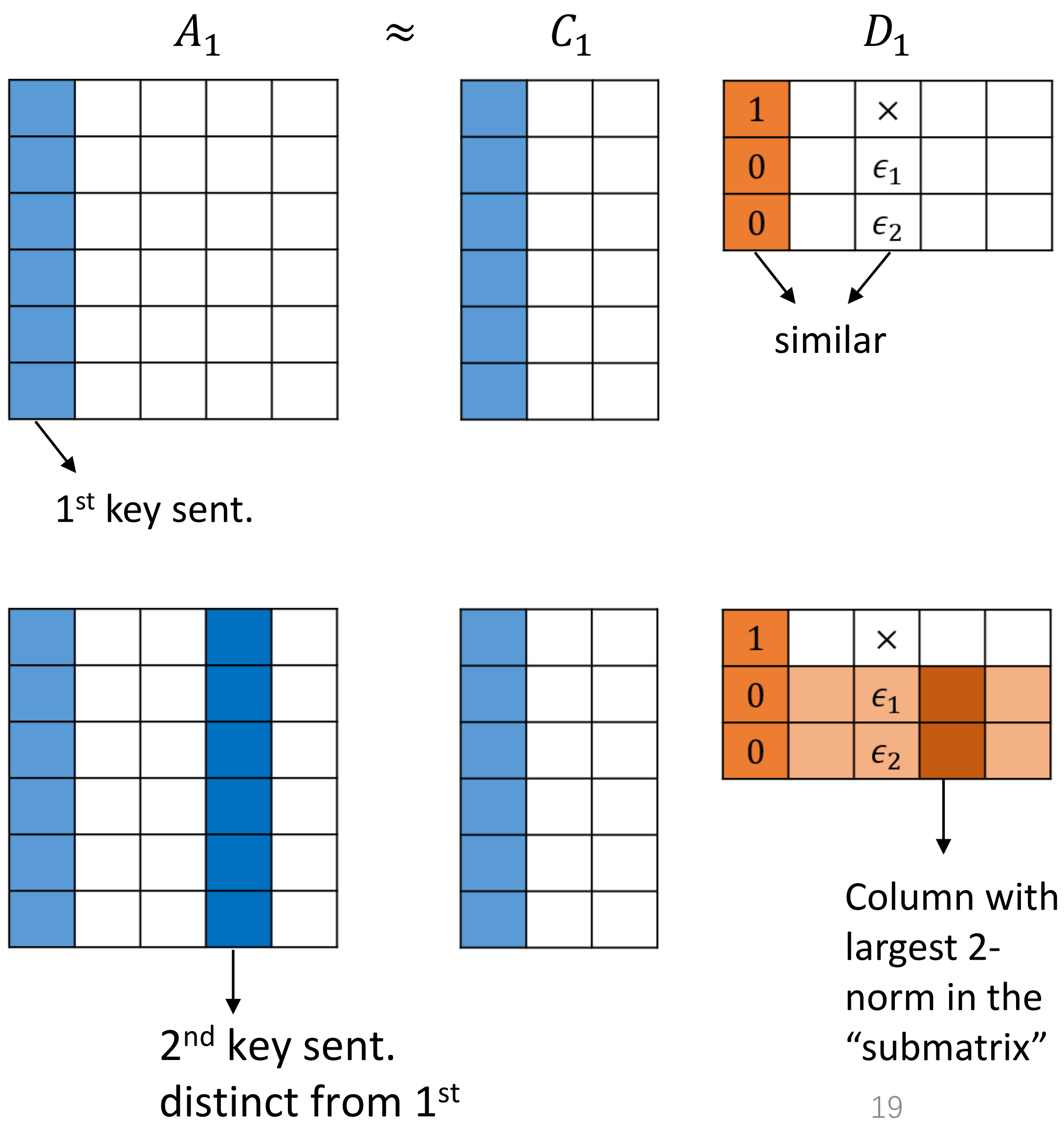
- Now the first column in $\boldsymbol{AP_1}$ is approximately equal to the first column in new basis $\boldsymbol{CQ_1 T_1}$
- We have found the first key sentence



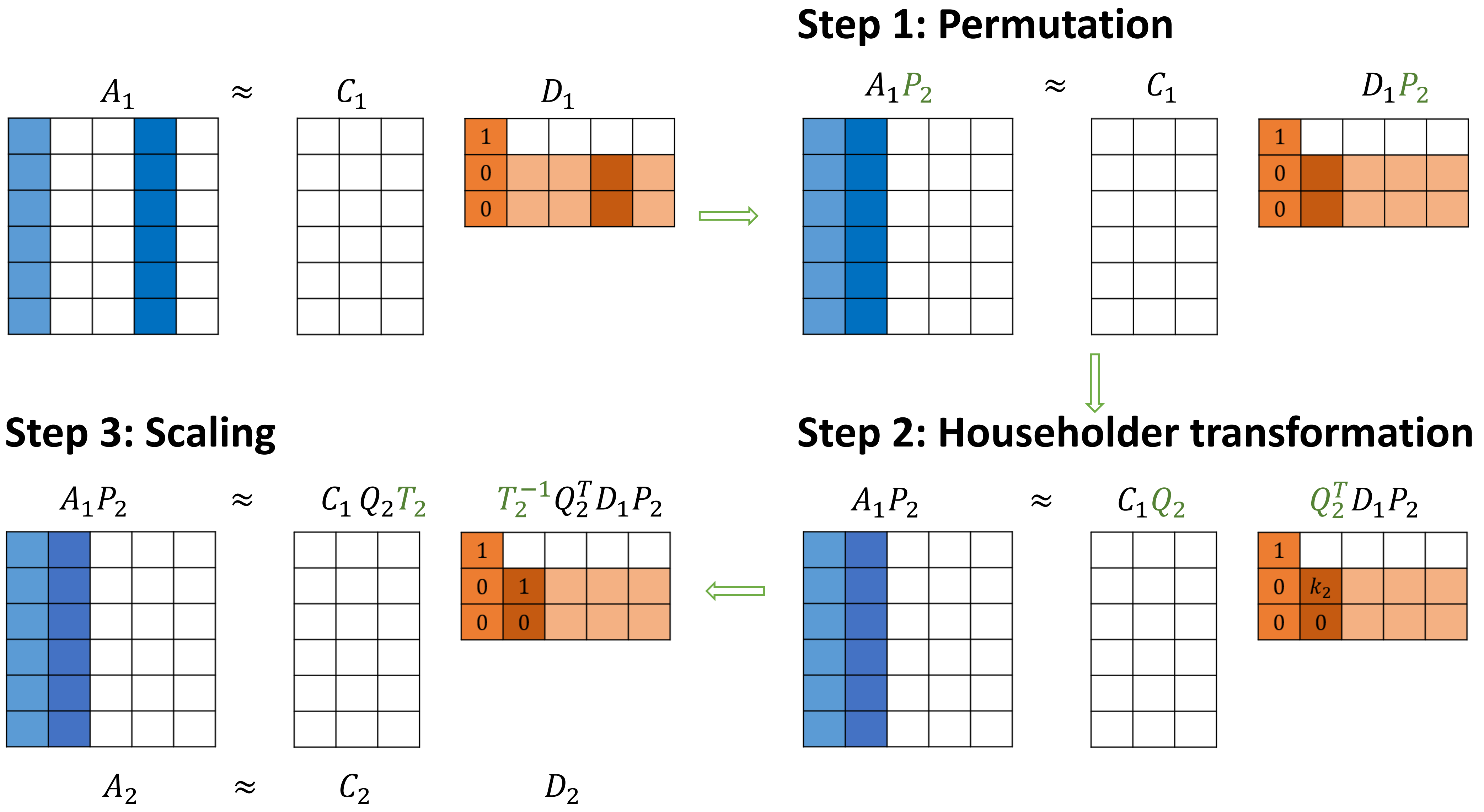
$$T_1 = \begin{bmatrix} k_1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad T_1^{-1} = \begin{bmatrix} \frac{1}{k_1} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Extraction of 2nd key sentence

- If another column in D was similar to the one that is now in position 1, it has small entries in rows 2 and 3.
- To avoid that column, we ignore the first row in D_1 while computing 2-norm in the next iteration
- Do the transformation again to extract second key sentence

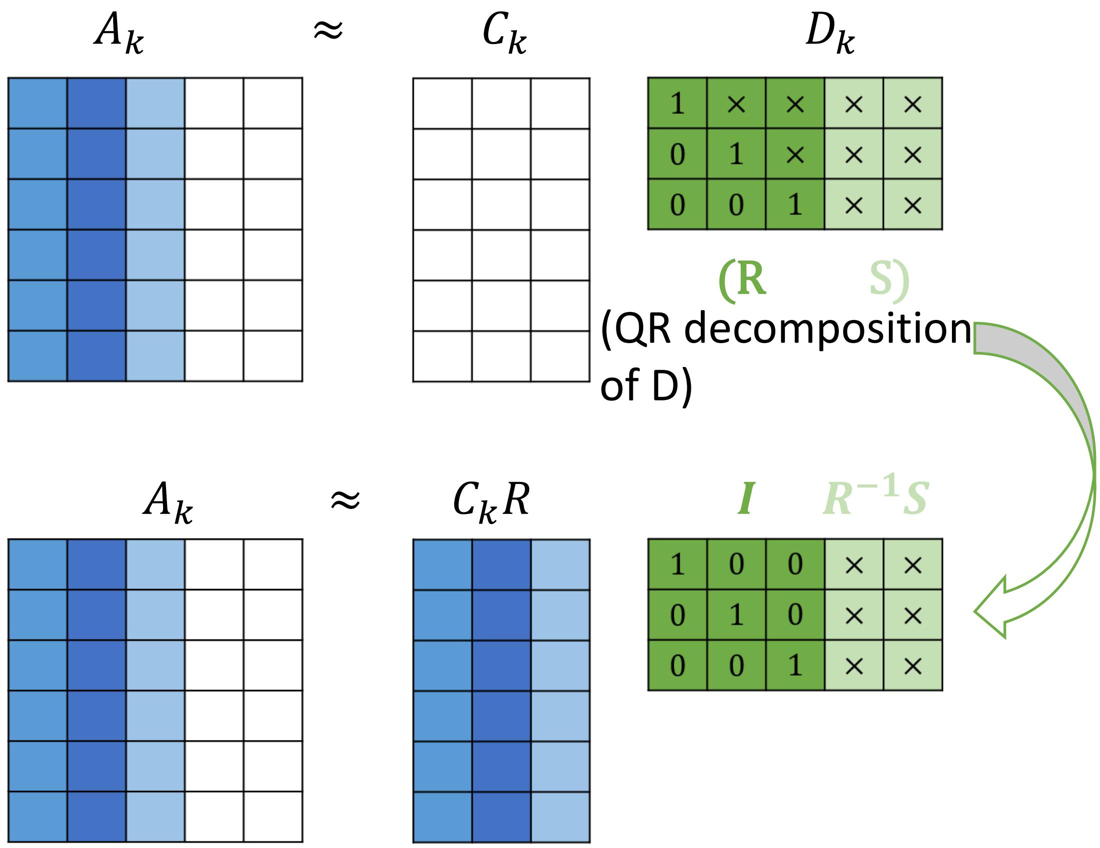


Extraction of 2nd key sentence



Key Sentence Extraction from a Rank-k Approximation

- After k iterations: $\boldsymbol{A_k} \approx \boldsymbol{C_k D_k}$
- The first k columns in $\boldsymbol{A_k}$ are now the k key sentences.
- If we apply $\boldsymbol{R}$ and $\boldsymbol{R^{-1}}$, we get new basis vectors directly corresponding to the key sentences.
- In practice, we just apply QR decomposition to $\boldsymbol{D}$ to get $\boldsymbol{D_k} = (\boldsymbol{R}\ \boldsymbol{S})$, remember the permutation order and find the corresponding columns in $\boldsymbol{A}$.
- If we are only interested in finding the top k sentences, we don't need to apply any transformations to the matrix $\boldsymbol{C}$.

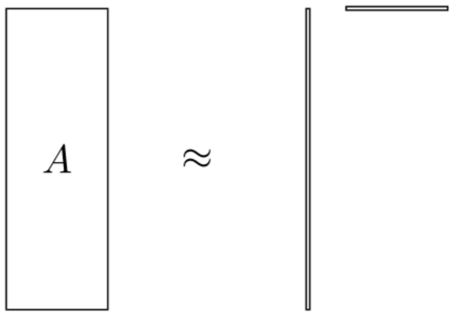


Summary

Automatic key word and key sentence extraction through

- Rank-1 approximation of term-sentence matrix A:

$$A \approx \sigma u v$$



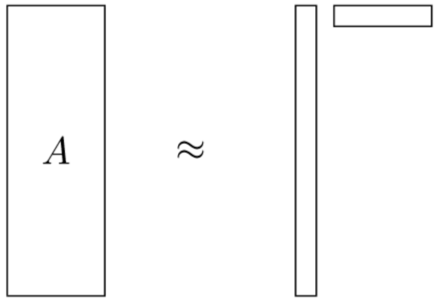
u: saliency scores of terms (1st left singular vector)

v: saliency scores of sentences (1st right singular vector)

Problem: extract similar sentences

- Rank-k approximation of term-sentence matrix A:

$$A \approx CD$$



Transform D and extract k key sentences step by step to avoid similarity

Bibliography

Eldén, L. (2007). Automatic Key Word and Key Sentence Extraction. In *Matrix methods in data mining and pattern recognition*. Philadelphia: SIAM.

Klaus.S. Extraction from a rank-k approximation (Survey).

Zha, H. (2002). Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 113-120).