Applications

Doc. Summarization

Doc | Lengthy one!

L

reduce by

orders of magnitude.

L → L/10 ; L/100 ....

Summary ① L/2 ✓

② $\sqrt{L}$

---

① Genres of summaries

② Algorithmic approaches to summarization

③ Evaluating summaries.

"InShorts"

→ KAP incubation.

CSE Alumni

Deepit Puralayasthya.

---

① Indicative vs. informative.

- Indicative: headlines. very precise
- informative: has some content drawn from the main doc.

② Extractive vs. abstractive.

- Extractive: ✓ Extracts parts from the main document to build the summary
- abstractive: generative — rewrites the info in the document in its own "supposedly coherent" version.

③ Generic vs query oriented.

↓ provides the author's view

↓ provides the summary as per the user query / demand.

④ Background vs just-in-the-news

News articles

↓ assume that the reader has no prior knowledge

↓ assume that the reader is well aware of the situation.

⑤ Single vs Multi-document

↓ summary is generated from a single doc.

↓ multiple docs are collated to generate the summary.

- Abstracts in ~~articles~~ Scientific papers.

- Section headings
- Conclusion.

Meta-linguistic cues hidden in general documents.

"In summary, ..."

"In conclusion, ..."

→ font (height, type etc.)

Graph based summarization.

- nodes & edges (somehow represents the documents)
- Measure certain properties of this graph → build your summary.

↓

→ "Prestige in social networks"

→ Dragomir Radev
ACL, AAAI, ACM fellow.

Network/graph → nodes are sentences.

edges are similarity between sentence pairs.

Vectorization

tf-idf of the sentences.

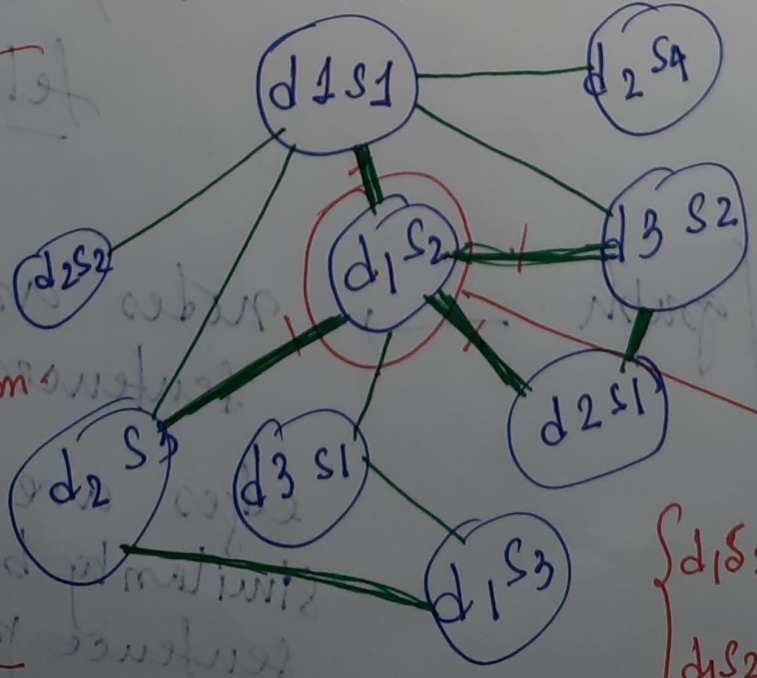
Cosine similarity between tf-idf vectors.

Sentence: ID : dXsY

document X sentenceY

D2S3.

3rd sentence from document 2.



Degree based summarization

— [0 - 0.5]

— [0.5 - 1].

→ degree

d1s2 → similar to most other sentences

d1s2 → highly similar to many sentences

Noisy links.

↳ get rid of the noisy links

$\tau \leftarrow$ threshold.

$\tau > 0.1$? (accept all the edges else drop)

$\tau > 0.01$?

$\tau > 0.5$? $\longrightarrow$ Empirically.

High weight edges $\longrightarrow$ graph becomes sparse.

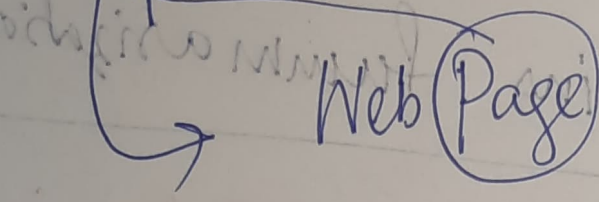
Low weight edges only $\longrightarrow$ too much of redundant & spurious info.

Binarization of a weighted graph.

Degree centrality based summarization.

1. Compute the degree of each node.
2. Pick the node with highest degree (u).
3. Remove all neighbors of u
4. Repeat the above process till the allocated budget expires.

PageRank ← Google:



Recursive importance.

a nodes importance is determined by how important its neighbors are.

Run PageRank on the sentence-sentence graph. — TextRank / LexRank.

1. Compute PageRank of nodes
2. Put the highest PageRank node into the summary
3. Remove neighbors.
4. Repeat process.

— DO —

— DO —

— DO —

(3.5) Recompute PageRank.

— DO —

Linguistic Method of summarization.

Lexical Chains Summarization.

↳ ~~collin~~ Collecting information from correlated chain of words.

Build Chains:

(a) pick a set of candidate words (selected words: nouns / noun compounds)

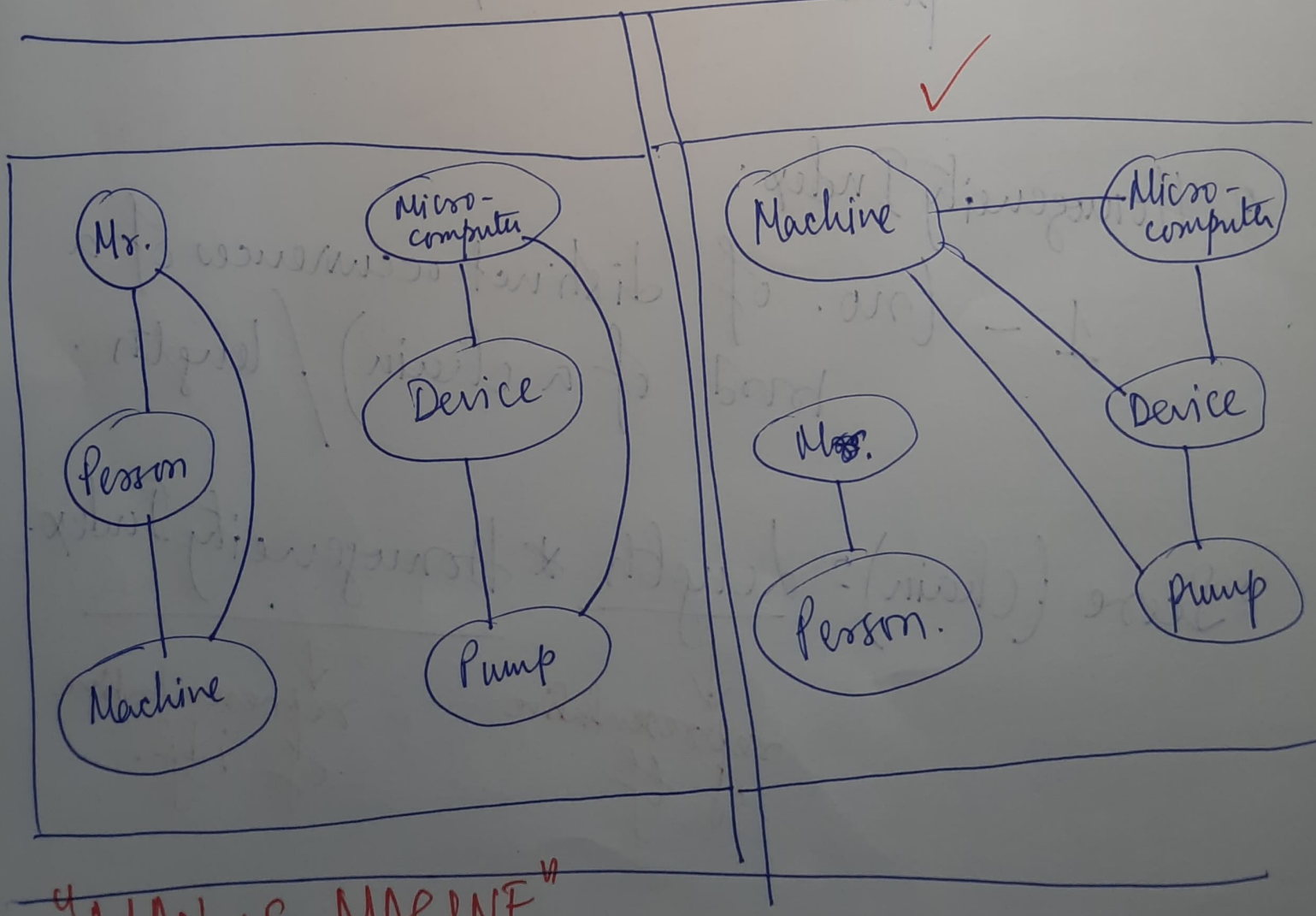
(b) For each candidate word build a chain based on its relatedness with other words.

(c) If the candidate word is already related to words in a particular chain then fuse it with that chain rather than building a new chain for it.

Lexical resources : WordNet.

↳ network of words with the relationships as edges.

hyperonym — I:SA — hyponym



"MAN IS MACHINE"

Strong chains:

→ Scoring.

Heuristics:

- Length: The number of occurrences of a particular word of the chain.

- Homogeneity Index:

  1 − (no. of distinct occurrences of a word of a chain) / length.

- Score (chain) = Length * Homogeneity Index.

  Length → representative of tf

  Homogeneity Index → representative of idf.

Chain: { microsoft (10) ✓ concern (1) ✓ company (6) ✓
entertainment (1) ✓ enterprise (1) ✓
mit (1) ✓ .

in the doc. microsoft has occurred 10 times, concern 1 time ....

Length = 20 (10+1+6+1+1+1).

$HI = 1 - \frac{6}{20} = \frac{14}{20}$

$Score = \frac{20 \times 14}{20} = 14.$

Strong Chain:

Score (chain) > Average (Score) + 2 Stdev (Score).

1. } strong chains.
2.
3.

① Choose the sentence that has the first occurrence of a word from the chain.

② Choose the sentence that has the first occurrence of the key-word of the chain.

better:

③ Choose ~~that~~ sentence that has the first occurrence of a high ~~desi~~ density of words from a strong chain.

more complex: