

Applications

Doc. Summarization

Doc. Lengthy one!

summary

① $\frac{L}{2}$

② \sqrt{L}

reduce by

"orders of magnitude".

$L \rightarrow L/10, L/100, \dots$

-
- ① Genres of summaries
 - ② Algorithmic approaches to summarization
 - ③ Evaluating summaries

"InShots"

20/11/2021

KGP incubation.

CSE Alum.

Deepit Purakayasthya.

① Indicative vs. informative.

↓
headlines.
very precise

↓
has some content
drawn from the main
doc.

② Extractive vs. Abstractive.

↓
✓ Extracts parts
from the
main document
to build the summary

↓
rewrites the
info in the
document in its
own "supposedly
coherent" ~~document~~
version.

generative

③ Generic vs query oriented.

↓
~~provides~~ 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 ~~some~~ 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:

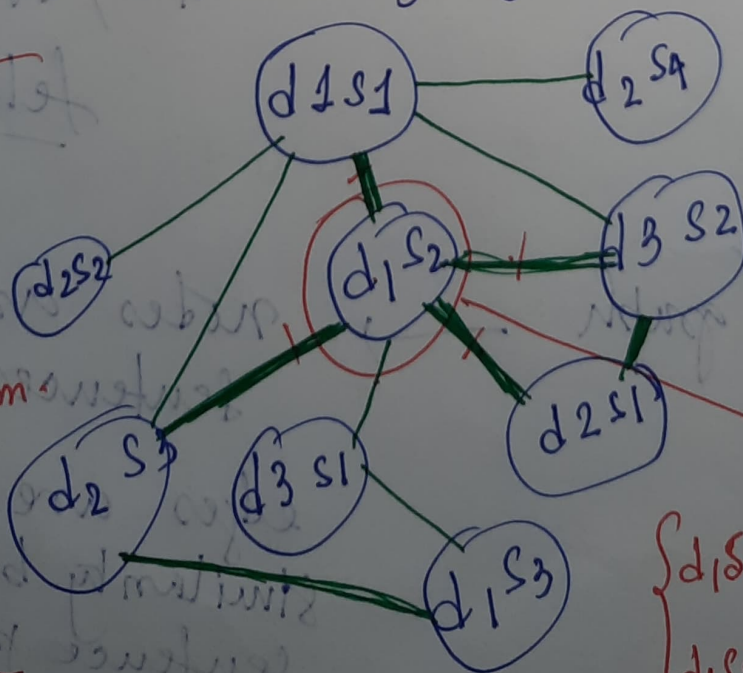
Sentence ID: $d \times s$
document \times sentence y

D2S3.

3rd sentence from document 2.

Degree based

Summarization



— $[0 - 0.5]$
— $[0.5 - 1]$

degree

$d1s2 \rightarrow$ similar to most other sentences
 $d1s2 \rightarrow$ highly similar to many sentences

Noisy links.

→ get rid of the noisy links

τ ← threshold.

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

$\tau > 0.01$?

$\tau > 0.5$?

→ Empirically.

High weight edges → graph becomes sparse.

Low weight edges only → too much of
redundant &
obvious 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

↻
Web (Page)

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.

→ ~~Collecting~~ Collecting information
from correlated chain of words.

Build chains:

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

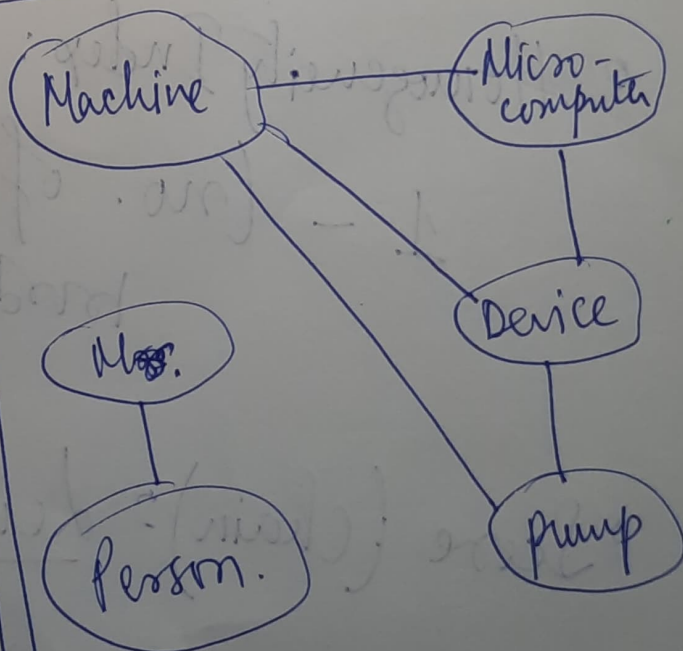
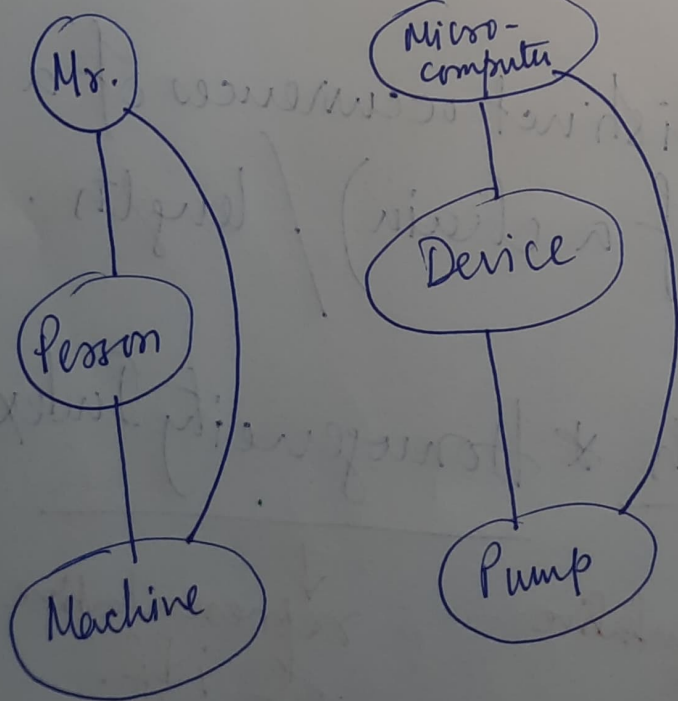
(b) For each candidate word build
a chain based on its relatedness
with other words.

(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 LISA hyponym



"MAN IS MACHINE"

Strong chains:

→ Scoring.

Heuristics:

- Length: The number of occurrences of a particular word of the chain.
- Homogeneity Index:
$$1 - (\text{no. of distinct occurrences of a word of a chain}) / \text{lengths}.$$
- Score (chain) = $\frac{\text{Length}}{\text{Homogeneity Index}}$
↓ representative of tf ↓ representative of idf.

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

$$\text{Length} = 20 \quad (10 + 1 + 6 + 1 + 1 + 1)$$

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

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

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

Strong Chain:

$$\text{Score}(\text{chain}) \rightarrow \text{Average}(\text{score}) + 2 \text{ stdev}(\text{score})$$

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.

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