Wordnet based summarization.

Idea :

1. select sentences based on their $\boxed{\text{semantic}}$ content

   ↳ meaning.

2. Given relative importance w.r.t ~~in conjunction~~ to the semantics of the whole text.

3. Reduce the pieces of text ~~cous~~ corresponding to the same semantic content.

   ↳ reduction of redundancy.

Key steps.

1. Pre processing
2. Subgraph construction from the WordNet
3. Synset Ranking
4. Sentence selection
5. PCA
6. Final pruning.

① **Preprocessing.**

ⓐ Split the text into sentences.

ⓑ POS tagging. (every word in the sentence is tagged with its most relevant POS) → NLTK.

↳ detect the correct sense of the word.

pant → noun (clothing).

pant → verb (fast breathing)

ⓒ. Identifying collocations.

words that typically appear together in a sentence — 4 miles per hours

↳ all idiomatic phrases.

ⓓ Remove stop words.

it, of etc.

→ The sequence is very important. "take off"

(2) Sub-graph Construction.

(a) Mark all the words and collocations that appear in the text (to be summarized) in the WordNet

hypernymy.

(b) Traverse the generalization edges upto a fixed depth & mark the synsets you visit

↳ groupings of synonymous words that express the same concept.

book.n.02 → [book.n.01, collocation.n.02, impression.n.06, magazine.n.01, ... volume.n.04]

↑ noun.

→ physical objects of a number of pages bound together.

ⓒ Construct a graph containing only the marked synsets as nodes & the generalization relationships as edges —— synset sub-graph.

---

③ **Synset Ranking.**

Rank the synsets ~~bas~~ based on their relevance in the text (to be summarized)

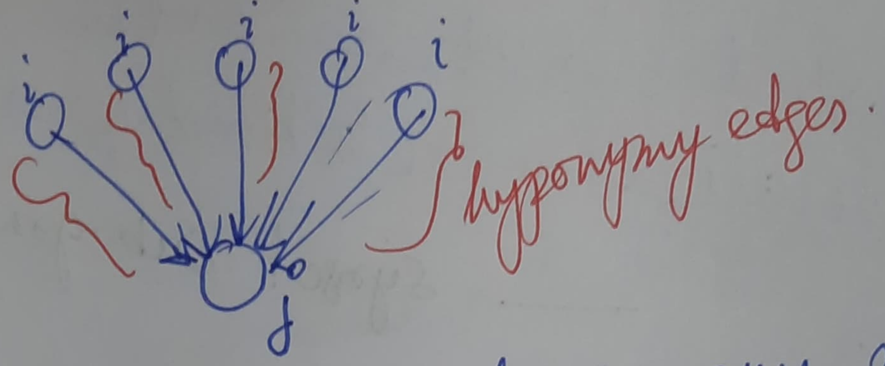
ⓐ Construct a rank vector R corresponding to each node of the graph. R is of dimension n.

Each entry $= \frac{1}{\sqrt{n}}$ n is the no. of nodes / synsets in the graph.

ⓑ Authority matrix

$$A(i,j) = \frac{1}{(\text{numba\_of\_predecessors}(j))} \quad \text{if } j \text{ is a child of } i$$

$$= 0 \quad \text{otherwise.}$$



↳ how many nodes does j draw its meaning from.

Ⓒ Update Rank vector:

$$R_{new} = \frac{R_{old} * A}{|R_{old} * A|}$$

Analogous to PageRank.

unit normal in that direction.

till $|R_{new}|$ changes less than a predefined threshold.

Higher values correspond to better ranked nodes.

(4) Sentence selection.

(a) Construct a matrix M with m rows ↳ no. of sentences in the text to be summarized.
& n columns ↳ no. of nodes in the subgraph

words/synsets (n)

$$\begin{array}{c|c} & w_1 \;\; w_2 \;\; w_3 \; \cdots \cdots \; w_n \\ \text{sentences } (m) & \begin{bmatrix} \; & \; & \; \\ \; & \; & \; \\ \; & \; & \; \end{bmatrix} \end{array}$$

(b) For each sentence S_i

Traverse the subgraph following the gen. edges with the words present in S_i

Find all the reachable synsets SY_i

For each $sy_j \in SY_i$ set $M[S_i][sy_j] = R[sy_j]$.

(5) PCA.

~~The princi~~

(a) Compute the principal components of M.

↳ eigenvectors of M with the largest few eigenvalues.

↳ Important "meaning" directions.

(b) Sort the eigen vector based on their eigen values. take the top few eigen vectors & compute projection on each sentence.

$$Pr(\vec{e})_{\vec{s}_1} = \frac{\vec{e}\,\vec{s}_1}{|\vec{s}_1|}$$

Max

λ_1 e_1 ← on how many sentences?

Second max

λ_2 e_2 ← on how many sentence?

→ K sentences.

through the eigenvector i.

$$\underline{\underline{K}} \propto \frac{\lambda_i}{\sum_j \lambda_j}$$

$$K \simeq \left(\frac{\lambda_i}{\sum \lambda_i}\right) N$$

← N is no. of sentences in the target summary (budget).

(6) <u>Final pruning</u>:

<u>Removal of undefined references</u>.

(a) Remove sentences that start with pronouns he/she/it.

(b) Remove sentences within quotes.