Wordnet based summarization.

~~Finds~~ Idea:

(1) select sentences based on their semantic content
↳ meaning.

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

(3) Reduce the ~~text~~ pieces of ~~same~~ corresponding to the same semantic content.
↳ reduction of redundancy.

Key steps.

1. Preprocessing
2. Subgraph construction from the Hotset
3. Synset Ranking
4. Sentence selection
5. PCA
6. Final pruning.

① Preprocessing.

(a) Split the text into sentences.

(b) 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)

(c) Identifying collocations.

words that typically appear together in a sentence — 4miles per hour
↳ all idiomatic phrases.

(d) Remove stopwords.
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.

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

---

(3) Synset Ranking:

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

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

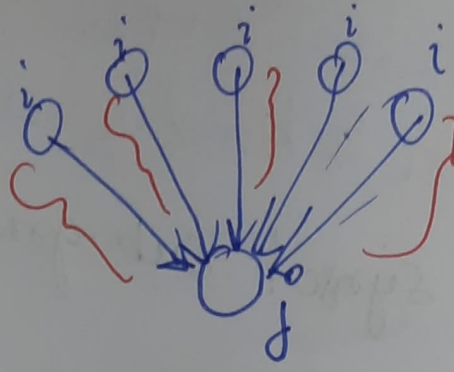
Each entry $= \frac{1}{\sqrt{n}}$

n is the no. of nodes / ~~synsets~~ synsets in the graph.

(b) Authority matrix

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

$= 0$ Otherwise.



hyponymy edges.

↳ how many nodes does j ~~[illegible]~~ draw its meaning from.

(c) 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 & n columns
- n → no. of nodes in the subgraph
- m → no. of sentences in the text to be summarized.

Sentences (m) \ words/synsets (n)	w_1	w_2	w_3	. . .	w_n

(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 principal~~

(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 eigenvectors
& compute projection on each sentence. $Pr(\vec{e})_{\vec{s}_1} = \frac{\vec{e} \cdot \vec{s}_1}{|\vec{s}_1|}$

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.

$$K \propto \frac{\lambda_{ik}}{\sum_j \lambda_j}$$

$$K \simeq \left(\frac{\lambda_i}{\sum \lambda_i}\right) N$$ ← N is no. of sentences in the target summary (budget).

(b) Final pruning.

Removal of undefined references.

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

(b) Remove sentences within quotes.