Text Summarization

Noah Gallant // Jacob House

Computer Science 4750
Fall 2018

Topic

Text Summarization // Automatic Abstracting

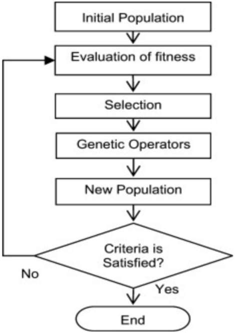
The process of reducing an input text to a smaller, more concise version of itself.

Use cases include: search engine results, academic research, legal contract analysis, and more advanced email inbox filtering.

Different Approaches

Evolutionary Algorithm

1. Assign weights to text features
2. Create population of distinct summaries
3. Assess fitness
4. Choose summaries with highest fitness
5. Create offspring from chosen summaries
6. Repeat steps 3–5 until the summary is concise enough

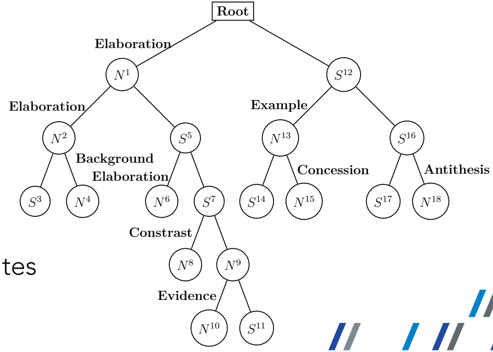


Different Approaches

Nested Tree

The goal in this approach is to repeatedly trim the tree until the desired summary size is reached.

- Tree structure is dependency based
  - Inter-sentence
  - Inter-word
- Tree trimming to reduce size
  - Removes duplicates
  - Removes less important nodes



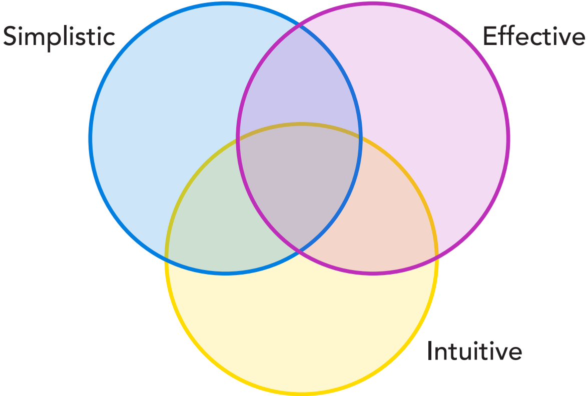
Different Approaches

Graph-Based

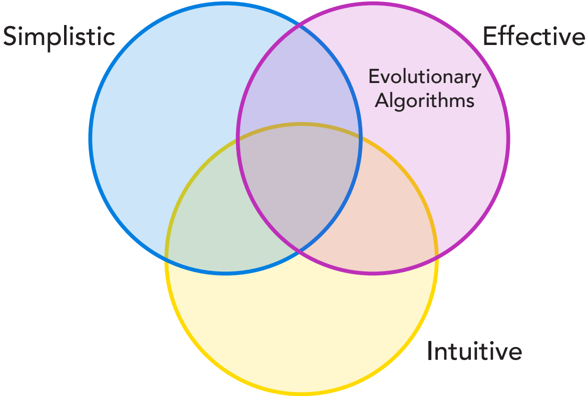
- Nodes created for each sentence
- Edges added between sentence nodes
  - Adjacent sentences in the text
  - Sentences that share common words
- Edges with higher weight denote dependency
- Nodes (sentences) with highest total edge weight should be included in the summary

MEMORIAL UNIVERSITY

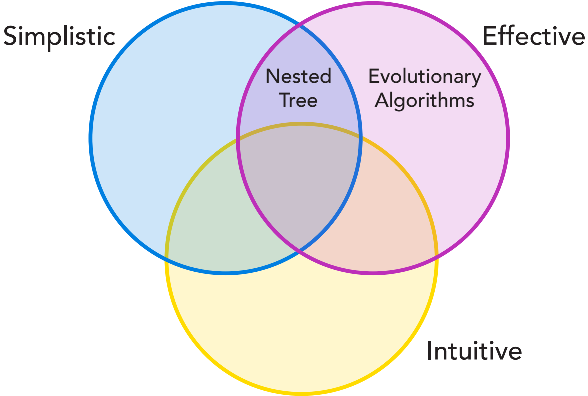
Our Design



Our Design


Simplistic
Effective
Evolutionary Algorithms
Intuitive

Our Design



Simplistic

Effective

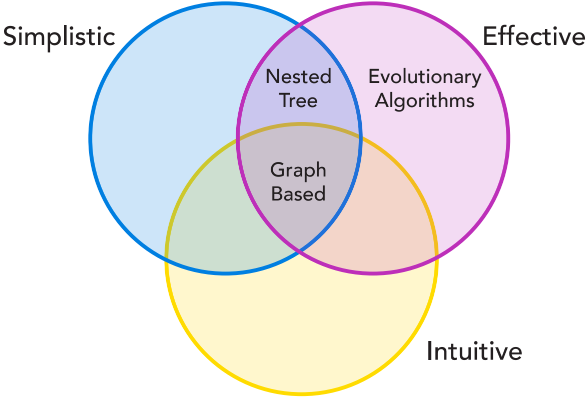
Intuitive

Nested Tree

Evolutionary Algorithms

MEMORIAL UNIVERSITY

Our Design



Simplistic

Effective

Intuitive

Nested Tree

Evolutionary Algorithms

Graph Based

Implementation

Preprocessing

- Sentence separation
- Deals with special/unicode characters

Implementation

The `Graph` Class

The `Graph` class is a singleton which contains all the nodes and edges. It is the responsibility of this class to do the text preprocessing, and to create nodes and the edges between them.

- Contains a list of:
    - Nodes
    - Edges
    - Words in the text

Implementation

The Node Class

A new node object is created for every sentence in the input text. It contains the sentence, a list of words in the sentence, and a list of edges which associate it with other sentence nodes.

- Initialized with 2 edges
    1. Connecting the preceding sentence
    2. Connecting the following sentence
- More edges are added between nodes that share a common word

MEMORIAL
UNIVERSITY

Implementation

The Edge Class

The Edge class contains the two nodes it is connecting, as well as the information about the connection.

This information includes the words connecting the sentences, as well as if they are sentences connected by proximity.

Questions?

H. P. Edmundson. “Problems in Automatic Abstracting”. In: *Commun. ACM* 7.4 (Apr. 1964), pp. 259–263. ISSN: 0001-0782. DOI: 10.1145/364005.364088. URL: http://doi.acm.org/10.1145/364005.364088.

H. P. Edmundson and R. E. Wyllys. “Automatic Abstracting and Indexing — Survey and Recommendations”. In: *Commun. ACM* 4.5 (May 1961), pp. 226–234. ISSN: 0001-0782. DOI: 10.1145/366532.366545. URL: http://doi.acm.org/10.1145/366532.366545.

Mahak Gambhir and Vishal Gupta. “Recent automatic text summarization techniques: a survey”. English. In: *The Artificial Intelligence Review* 47.1 (Jan. 2017). Copyright - Artificial Intelligence Review is a copyright of Springer, 2017; Last updated - 2018-10-06, pp. 1–66. URL: https://search-proquest-com.qe2a-proxy.mun.ca/docview/1857255406?accountid=12378.

MEMORIAL
UNIVERSITY

Jaya Jagadeesh, Prasad Pingali, and Vasudeva Varma. "Sentence Extraction Based Single Document Summarization". In: (Jan. 2005).

Gunnel Källgren. "Automatic Abstracting Content in Text". In: *Nordic Journal of Linguistics* 11.1-2 (1988), pp. 89–110. DOI: `10.1017/S0332586500001761`.

Yuta Kikuchi et al. "Single Document Summarization based on Nested Tree Structure". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, 2014, pp. 315–320. DOI: `10.3115/v1/P14-2052`. URL: `http://aclweb.org/anthology/P14-2052`.

Nandhini Kumaresh and Balasundaram Sadhu Ramakrishnan. “Graph Based Single Document Summarization”. eng. In: *Data Engineering and Management: Second International Conference, ICDEM 2010, Tiruchirappalli, India, July 29-31, 2010. Revised Selected Papers*. Vol. 6411. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 32–35. ISBN: 9783642278716.

Yogesh Kumar Meena and Dinesh Gopalani. “Evolutionary Algorithms for Extractive Automatic Text Summarization”. In: *Procedia Computer Science* 48 (2015). International Conference on Computer, Communication and Convergence (ICCC 2015), pp. 244–249. ISSN: 1877-0509. DOI: `https://doi.org/10.1016/j.procs.2015.04.177`. URL: `http://www.sciencedirect.com/science/article/pii/S1877050915006869`.

- Dragomir R. Radev. “Experiments in single and multidocument summarization using MEAD”. In: *In First Document Understanding Conference*. 2001. URL: `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.119.4643`.

- J. E. Rush, R. Salvador, and A. Zamora. “Automatic abstracting and indexing. II. Production of indicative abstracts by application of contextual inference and syntactic coherence criteria”. In: *Journal of the American Society for Information Science* 22.4 (), pp. 260–274. DOI: `10.1002/asi.4630220405`. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.4630220405`. URL: `https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.4630220405`.

- Prateek Singh. *PSSummary*. 2018. URL: `https://github.com/PrateekKumarSingh/PSSummary`.

Caroline Uyttendaele, Marie-Francine Moens, and Jos Dumortier. “Salomon: Automatic Abstracting of Legal Cases for Effective Access to Court Decisions”. In: *Artificial Intelligence and Law* 6.1 (Mar. 1998), pp. 59–79. ISSN: 1572-8382. DOI: `10.1023/A:1008256030548`. URL: `https://doi.org/10.1023/A:1008256030548`.