TEXT SUMMARIZATION

Prof. Marco Viviani

marco.viviani@unimib.it

Text Summarization

"*Text summarization is the process of distilling the most important information from a source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks)*"*

- Goal: produce an abridged version of a text that contains information that is important or relevant to a user.

*Mani, Inderjeet. *Advances in automatic text summarization*. MIT press, 1999

Summarization Applications

- We are all familiar with summaries such as:
  - **headlines** (from around the world)
  - summaries (of e-mail threads)
  - minutes (of a meeting)
  - previews (of movies)
  - synopses (soap opera listings)
  - **reviews** (of a book, CD, movie, etc.)
  - digests (TV guide)
  - biography (resumes, obituaries)
  - abridgments (Shakespeare for children)
  - bulletins (weather forecasts/stock market reports)
  - sound bites (politicians on a current issue)
  - histories (chronologies of salient events)

Advantages

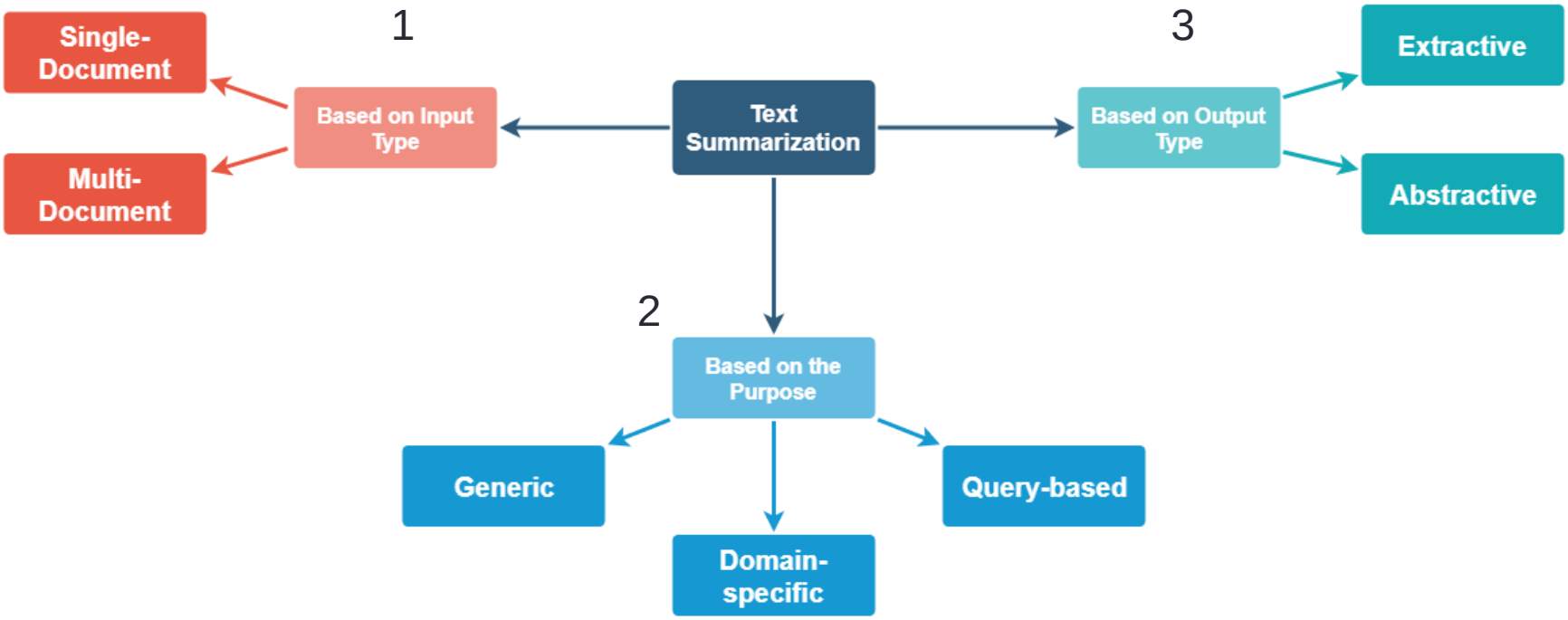
- Summaries reduce reading time.
- When searching for documents, summaries make the selection process easier.
- Automatic summarization improves the **effectiveness** of indexing (IR) and other TM tasks (e.g., classification, clustering…)
- Automatic summarization algorithms are less biased than human summarizers.

Advantages

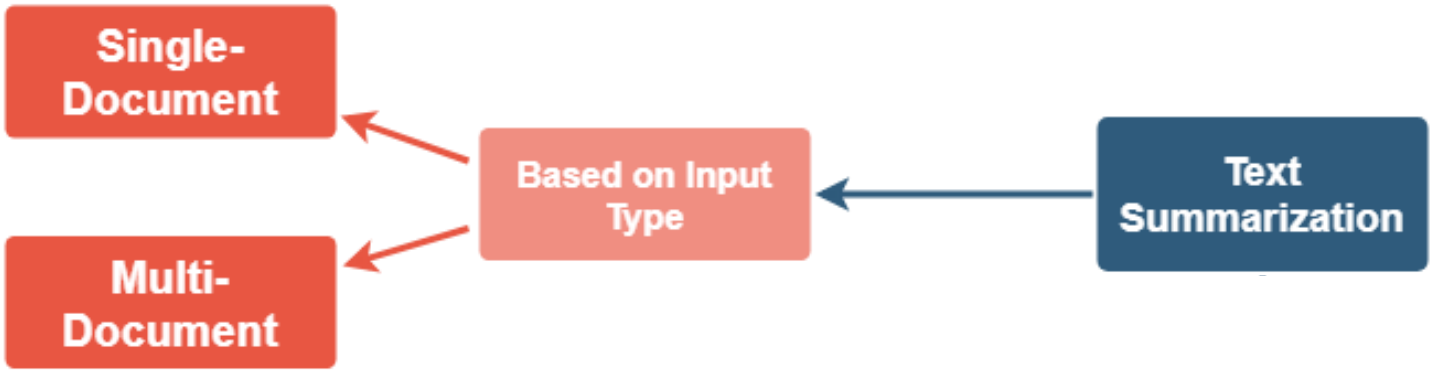
- Personalized summaries are useful in question-answering systems as they provide personalized information.

- Using automatic or semi-automatic summarization systems enables commercial abstract services to increase the number of text documents they can process.

Text Summarization Dimensions



1. Based on input type

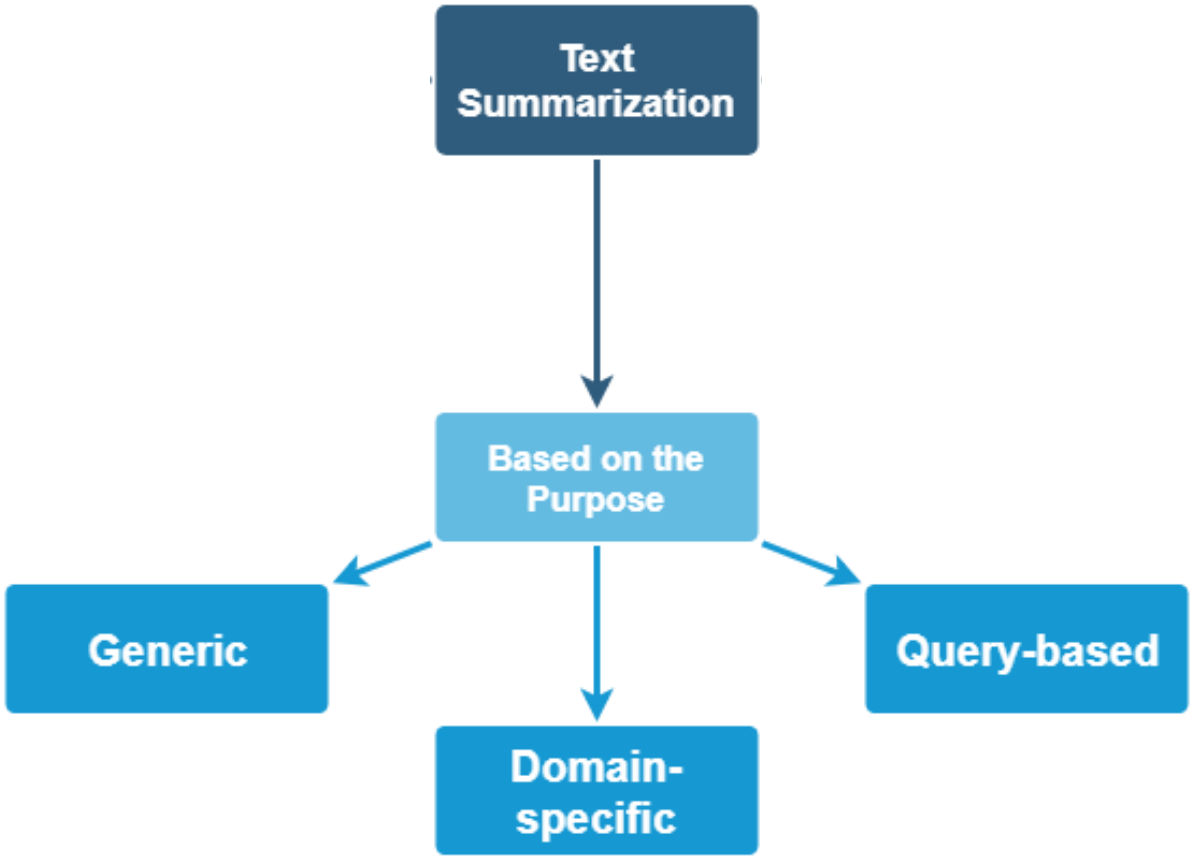


1. Based on input type

- Single-document summarization
  - Given a single document, produce:
    - Abstract
    - Outline
    - Headline
  - The input length is short. Many of the early summarization systems dealt with single document summarization.

- Multiple-document summarization
  - Given a group of documents, produce a gist of the content, e.g.:
    - A series of news stories on the same event.
    - A set of Web pages about some topic or question.
  - The input can be arbitrarily long.

2. Based on the purpose



2. Based on the purpose

- Generic
  - The model makes no assumptions about the domain or content of the text to be summarized and treats all inputs as homogeneous.
  - The majority of the work that has been done revolves around generic summarization.
- Domain-specific
  - The model uses domain-specific knowledge to form a more accurate summary.
    - For example, summarizing research papers of a specific domain, biomedical documents, etc.
- Query-based
  - The summary only contains information which answers natural language questions about the input text.

Query-based summarization

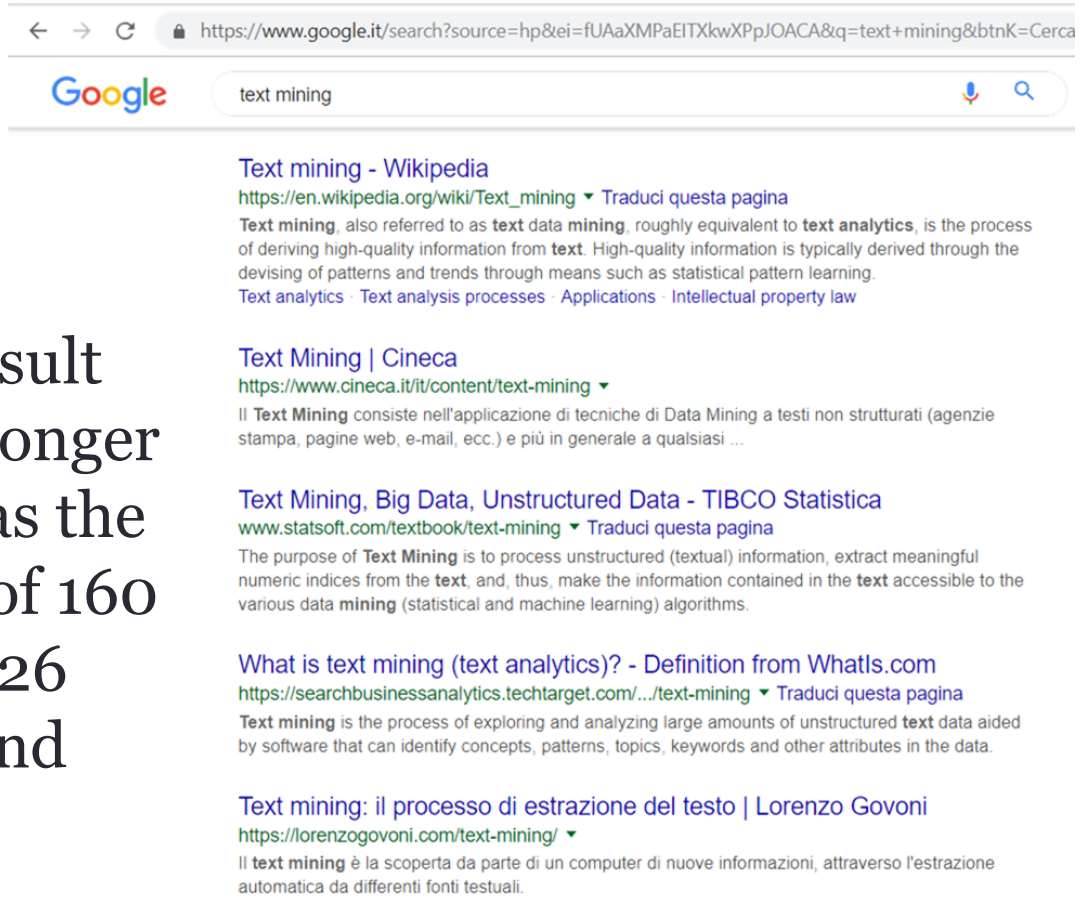
- Summarizes a document with respect to an information need expressed in a user query.

- A kind of complex question-answering:
  - Answer a question by summarizing a document that has the information to construct the answer.

https://pdfs.semanticscholar.org/a9c7/5fccaeaf4ace3cec1255f1dd33065c0cc813.pdf

Query-based summarization (Snippets)

- Create snippets summarizing a Web page for a query.
  - Google's search result snippets are now longer than what once was the maximum length of 160 characters (about 26 words) plus title and link.



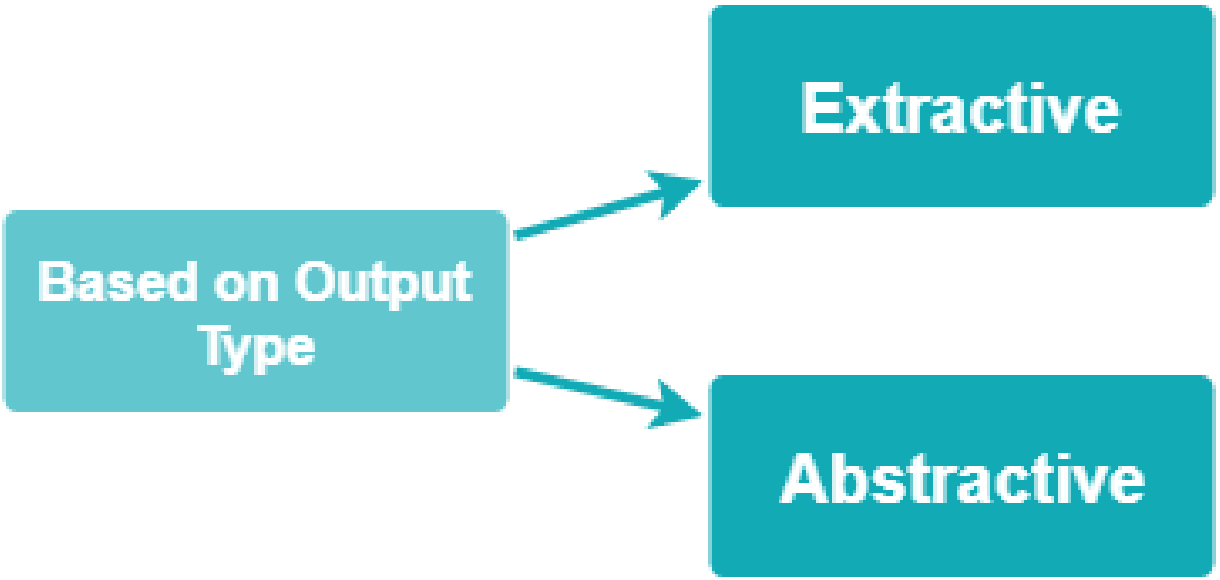
Query-based summarization (QA systems)

- Typical of Question-Answering (QA) systems (but not only).

- Create answers to complex questions summarizing multiple documents.
  - Instead of giving a snippet for each document.
  - Create a cohesive answer that combines information from each document.

https://elearning.unimib.it/pluginfile.php/688560/mod_resource/content/1/querybased_summarization_of_discussion_threads.pdf

3. Based on output type

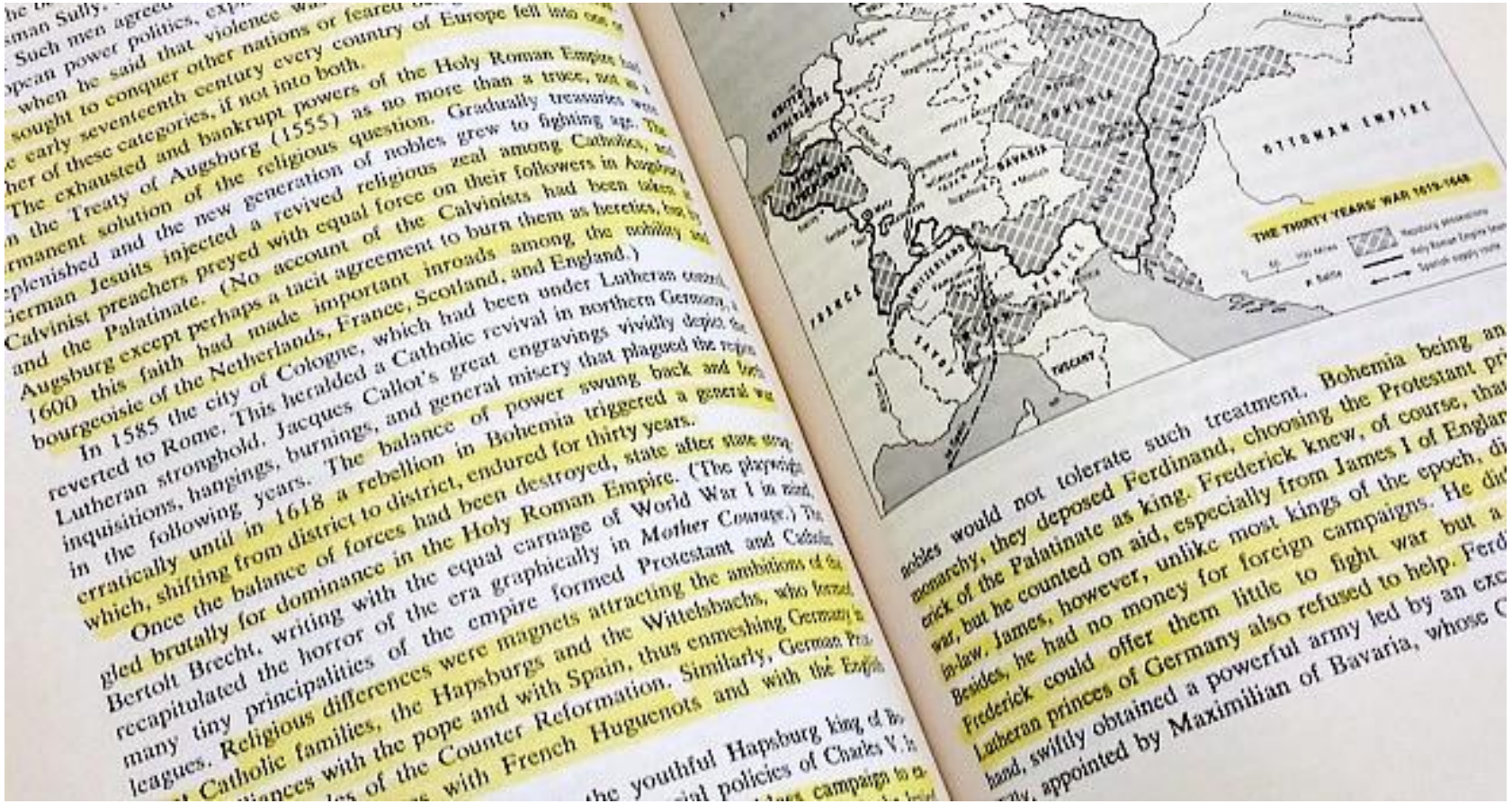


3. Based on output type (E)

- Extractive summarization
  - Important phrases or sentences are selected from the input text.
  - The summary is created from these phrases or sentences in the source document(s).

- Several summarization approaches today are extractive in nature.

Extractive summarization

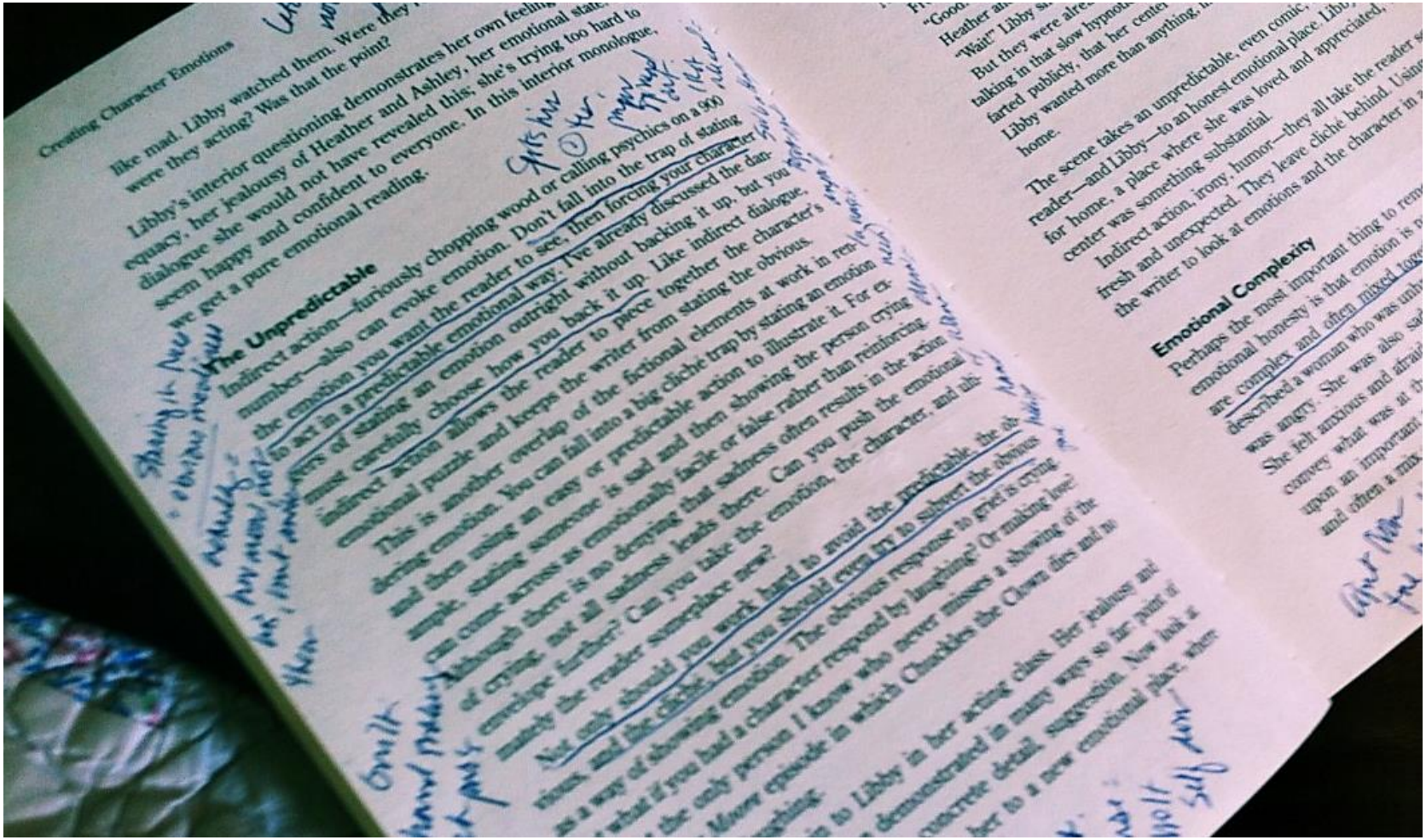


3. Based on output type (A)

- Abstractive summarization
  - Expresses the ideas in the source document(s) using (at least in part) different words.
  - The model forms its own phrases and sentences to offer a more coherent summary, like what a human would generate.

- This approach is definitely a more appealing, but much more complex and computationally expensive than extractive summarization.

Abstractive summarization



Abstractive summarization: Basics

- Sequence-to-sequence models
  - Use Recurrent Neural Networks (RNNs) or more advanced architectures like Long Short-Term Memory networks (LSTMs) or Transformers.
  - These models are trained to map input sequences (source document) to output sequences (summary).
- Attention mechanisms
  - Attention mechanisms enable the model to focus on different parts of the input sequence when generating each part of the summary. This helps capture the relevant information and context for creating a coherent summary.
- Encoder-decoder architecture
  - The encoder processes the input sequence, and the decoder generates the summary.
- Reinforcement learning
  - These models are trained using a combination of supervised learning and reinforcement learning, with the latter providing rewards or penalties based on the quality of generated summaries.
- Transfer learning
  - Pre-trained language models like BERT (Bidirectional Encoder Representations from Transformers) can be applied to abstractive summarization tasks.
  - Fine-tuning these models for summarization tasks can leverage the broad linguistic knowledge encoded in the pre-trained models.

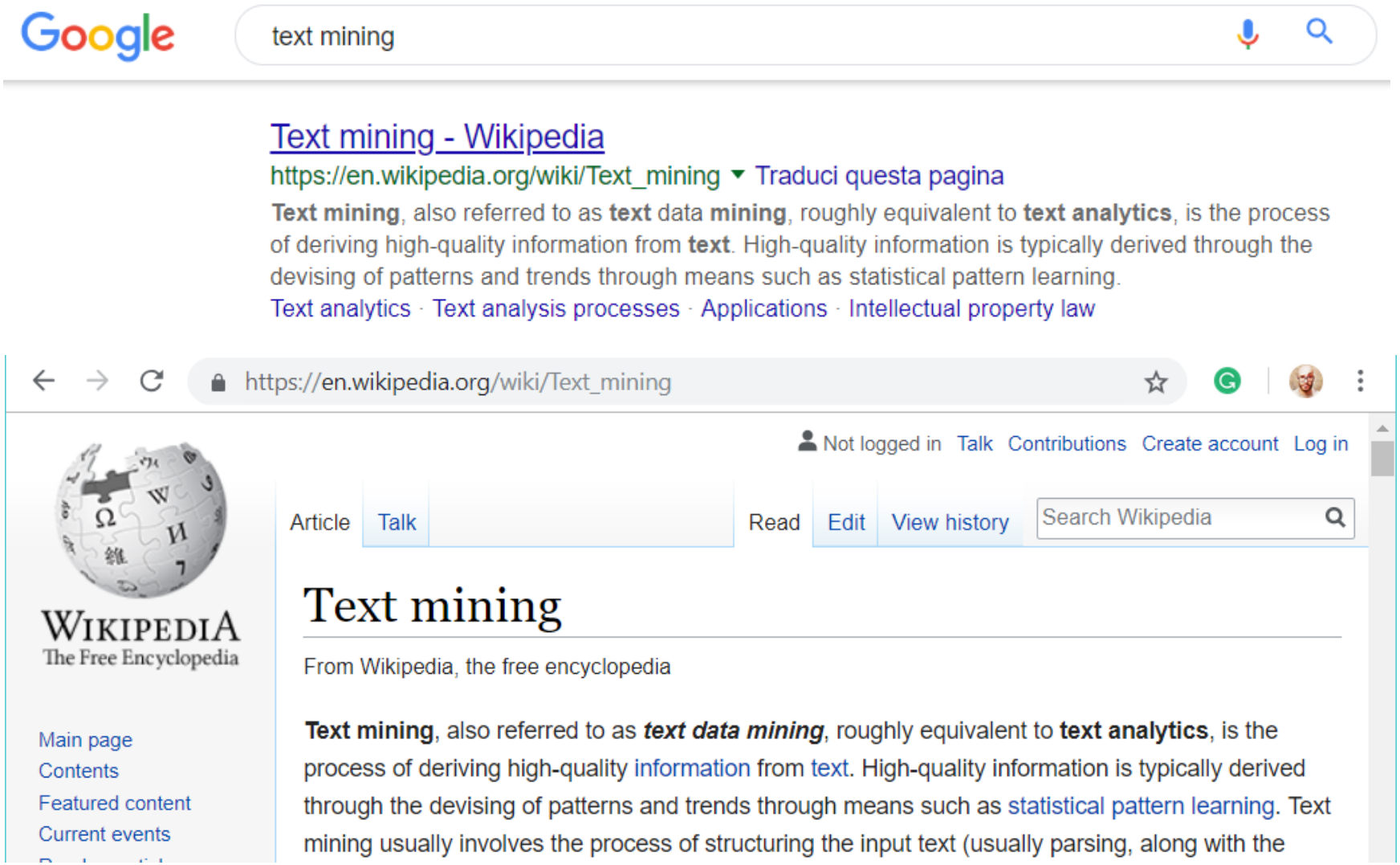
EXTRACTIVE SUMMARIZATION

Why extractive summarization?

- Extractive summarization is still used today and remains a relevant and widely employed approach in Natural Language Processing and Text Mining.

- Extractive summarization has some advantages:
  - Preservation of source content: Since it selects sentences directly from the source document, extractive summarization inherently preserves the language and content used in the original text.
  - Reduced risk of generating incoherent text: Extractive summarization avoids the challenge of generating new sentences, reducing the risk of generating grammatically incorrect or semantically inconsistent text.
  - Interpretability: The extracted sentences are often directly interpretable and traceable back to the source material, making it clear which parts of the document contributed to the summary.

Simplest strategy: take the first sentence



Summarization phases

1. Creating an intermediate representation of the input which captures only the key aspects of the text.

2. Scoring sentences based on that representation.

3. Selecting a summary consisting of several (scored) sentences.

Approaches

- Topic representation approaches
  - First derive an intermediate representation of the text that captures the topics discussed in the input.
  - Based on these representations of topics, sentences in the input document are scored for importance.

- Indicator representation approaches
  - The text is represented by a diverse set of possible indicators of importance which do not aim at discovering topicality.
  - These indicators are combined, very often using machine learning techniques, to score the importance of each sentence.

- A summary is produced choosing the sentences that will go in the summary one by one, or globally optimizing the selection, choosing the best set of sentences to form a summary.

1. Intermediate representation (1)

- Topic representation approaches:
  - Topic words approaches in which the topic representation consists of a simple table of words and their corresponding weights, with more highly weighted words being more indicative of the topic.
    - Lexical chain approaches in which a thesaurus such as WordNet is used to find topics or concepts of semantically related words and then give weight to the concepts.
  - Latent semantic analysis (LSA) in which patterns of word co-occurrence are identified and roughly construed as topics, as well as weights for each pattern.
  - Bayesian topic models (LDA) in which the input is represented as a mixture of topics and each topic is given as a table of word probabilities (weights) for that topic.

1. Intermediate representation (2)

- Indicator representation approaches:
  - Represent each sentence in the input as a list of indicators of importance such as:
    - Sentence length;
    - Location in the document;
    - Presence of certain words;
    - Etc.
  - In graph models, such as *LexRank* and *TextRank*, the entire document is represented as a network of inter-related sentences.

https://blog.floydhub.com/gentle-introduction-to-text-summarization-in-machine-learning/

2. Scoring sentences

- Once an intermediate representation has been derived, each sentence is assigned a score which indicates its importance.

- For topic representation approaches, the score is commonly related to:
  - how well a sentence expresses some of the most important topics in the document.
  - to what extent it combines information about different topics.

- For the majority of indicator representation methods, the weight of each sentence is determined by combining the evidence from the different indicators.
  - Most commonly by using machine learning techniques to discover indicator weight.

3. Selecting the summary

- The summarizer must select the best combination of important sentences to form a paragraph length summary.
  - Best *n* approaches
    - The top *n* most important sentences which combined have the desired summary length are selected to form the summary.
  - Maximal-marginal relevance approaches
    - Sentences are selected in an iterative greedy procedure:
      - At each step of the procedure the sentence importance score is recomputed as a linear combination between the original importance weight of the sentence and its similarity with already chosen sentences.
      - Sentences that are similar to already chosen sentences are dispreferred.
  - Global selection approaches
    - The optimal collection of sentences is selected subject to constraints that try to maximize overall importance, minimize redundancy, and, for some approaches, maximize coherence.

TOPIC REPRESENTATION APPROACHES

Topic words approaches (1)

- Intuition dating back to Luhn* (1958):
  - Choose sentences that have salient or descriptive words (topic words/signatures).
  - Frequent content words would be indicative of the topic of the article (stopwords are not considered).
  - Frequency thresholds to identify descriptive words in a document to be summarized.

*Luhn, Hans Peter. "The automatic creation of literature abstracts." *IBM Journal of research and development* 2.2 (1958): 159-165

Topic words approaches (2)

- The importance of a sentence is computed as:
  1. The number of topic signatures it contains.
  2. The proportion of topic signatures in the sentence.

- Both sentence scoring functions are based on the same topic representation; despite this, the scores they assign to sentences may be rather different.
  - The first approach is likely to score longer sentences higher, simply because they contain more words.
  - The second approach favors density of topic words.

Weighting words

- When assigning weights of words in topic representations, we can think of binary (0 or 1) or real-value (continuous) weights and decide which words are more correlated to the topic.

- The two most common techniques in this category are:
  - Word probability
  - TF-IDF

Word probability

- The probability of a word w is determined as:

$$P(w) = \frac{f(w)}{N}$$

where

- $f(w)$ number of times the word appears in the input (i.e., its frequency).
- N is the total number of words in the input.

The SumBasic system (1)

- Uses only the word probability approach to determine the sentence importance.

- For each sentence s_j in the input, it assigns a weight equal to the average probability of the words in the sentence:

$$g(s_j) = \frac{\sum_{w_i \in s_j} P(w_i)}{|\{w_i | w_i \in s_j\}|}$$

where $g(s_j)$ is the weight of sentence s_j.

https://www.cs.bgu.ac.il/~elhadad/nlp09/sumbasic.pdf

The SumBasic system (2)

- The algorithm then selects the best scoring sentence that contains the highest probability word.
- This step ensures that the highest probability word (which should represent the topic of the document) is included in the summary.
- After the best sentence is selected, the probability of each word that appears in the chosen sentence is adjusted (set to a smaller value, e.g., $P_{new}(w_i) = P_{old}(w_i) * P_{old}(w_i))$
  - The probability of a word occurring twice in a summary is lower than the probability of the word occurring only once.
- The selection loop is repeated until the desired summary lenght is achieved.

TF-IDF

- This weighting technique assesses the importance of words and identifies very common words (that should be omitted from consideration) in the document(s) by giving low weights to words appearing in most documents.

- The weight of each word w in document d is computed as follows:

$$q(w) = f_d(w) * \log \frac{|D|}{f_D(w)}$$

where $f_d(w)$ is term frequency of word w in the document d, $f_D(w)$ is the number of documents that contain word w, and $|D|$ is the number of documents in the collection D.

Centroid-based summarization (Basics) (1)

- First step:
  - TF-IDF vector representations of the documents are created.
  - A clustering algorithm is run over the TF-IDF vectors, adding documents to clusters and recomputing centroids.
    - Centroids can be considered as pseudo-documents that consist of those words whose TF-IDF scores are higher than a certain threshold and form the cluster.

https://elearning.unimib.it/pluginfile.php/688619/mod_resource/content/1/1-s2.0-S0306457303000955-main.pdf

http://www.summarization.com/mead/

http://www.decodeschool.com/blog/Python/Centroid-based-Text-summarization-in-Python

Centroid-based summarization (Basics) (2)

- Second step:
  - Using centroids to identify sentences in each cluster that are central to the topic of the entire cluster.
    - The sentences which are more similar to the centroid of the cluster are considered as central sentences.

- Two metrics are defined:
  - Cluster-based relative utility (CBRU)
    - Decides how relevant a particular sentence is to the general topic representing the entire cluster.
  - Cross-sentence informational subsumption (CSIS)
    - Measures redundancy among sentences.

- Based on the combination of the two metrics, the final score of each sentence is computed and the selection of sentences is determined.

Latent Semantic Analysis (LSA) (1)

- Latent semantic analysis (LSA) is an unsupervised technique for deriving an implicit representation of text semantics based on observed co-occurrence of words.
- LSA has been initially proposed for single and multi-document generic summarization of news, as a way of identifying important topics in documents without the use of lexical resources such as WordNet.
- Building the topic representation starts by filling in a n by m matrix A: each row corresponds to a word from the input (n words) and each column corresponds to a sentence in the input (m sentences).
- Entry a_{ij} of the matrix corresponds to the weight of word i in sentence j
  - If the sentence does not contain the word, the weight is zero, otherwise the weight is equal to the TF-IDF weight of the word.

Latent Semantic Analysis (LSA) (2)

- The matrix A can be represented as the product of three matrices: $A = U\Sigma V^T$
  - Matrix U is an n (words) by m (topics) matrix of real numbers. Each column can be interpreted as a topic, i.e., a specific combination of words from the input with the weight of each word in the topic given by the real number.
  - Matrix Σ is diagonal m (topics) by m (topics) matrix. The single entry in row i of the matrix corresponds to the weight of the "topic", which is the ith column of U.
  - Matrix V^T is an m (sentences) by m (topics) matrix, a new representation of the sentences, one sentence per row, each of which is expressed not in terms of words that occur in the sentence but rather in terms of the topics given in U.
  - The matrix $D = \Sigma V^T$ combines the topic weights and the sentence representation to indicate to what extent the sentence conveys the topic, with d_{ij} indicating the weight for topic i in sentence j.

Latent Semantic Analysis (LSA) (3)

- Hypotesis: often sentences that discuss several of the important topics are good candidates for summaries.

- To identify such sentences, the weight of sentence s_i is set equal to:

$$g(s_j) = \sqrt{\sum_{j=1}^{m} d_{ij}^2}$$

Bayesian topic models

- Bayesian topic models are probabilistic models that uncover and represent the topics of documents.

- Latent Dirichlet Allocation (LDA) model is the state-of-the-art unsupervised technique for extracting thematic information (topics) of a collection of documents.
  - Documents are represented as a random mixture of latent topics, where each topic is a probability distribution over words.

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.158.1654&rep=rep1&type=pdf

Bayesian topic models (Example)

- If we apply LDA to the "Romeo and Juliet" tragedy, when searching for three topics, each one represented by 4 keywords, we find:
  - One dominant topic described by "love", "death", "lady", "night";
  - Two minor topics described respectively by "gentlemen", "pretty", "lady", and "quarrel", and by "die", "youth", "villain", and "slaughter".
  - The topic importance is quantified by its keyword's weights.

- Different possibilities:
  - The dominant topic can be considered as the summary of the tragedy;
  - The weights (probabilities) associated to the dominant topic words can be used to weight the sentences in which they appear;
  - Variations of the previous solutions.

INDICATOR REPRESENTATION APPROACHES

Introduction

- Indicator representation approaches aim to model the representation of the text based on a set of features and use them to directly rank the sentences rather than representing the topics of the input text.

- Graph-based methods and machine learning techniques for summarization are often employed to determine the important sentences to be included in the summary.

Graph-based methods (1)

- They represent the documents as a connected graph.
  - Influenced by the PageRank algorithm.
- Sentences form the vertices of the graph and edges between the sentences indicate how similar the two sentences are.
- A common technique employed to connect two vertices is to measure the similarity of two sentences and if it is greater then a threshold, they are connected.
- The most often used method for similarity measure is cosine similarity with TF-IDF weights for words.

Graph-based methods (2)

- The graph representation results in two outcomes.
  - First, the partitions (sub-graphs) included in the graph, create discrete topics covered in the documents.
  - The second outcome is the identification of the important sentences in the document.
    - Sentences that are connected to many other sentences in the partition are possibly the center of the graph and more likely to be included in the summary.

Graph-based methods (3)

- Graph-based methods can be used for single as well as multi-document summarization.

- Since they do not need language-specific linguistic processing other than sentence and word boundary detection, they can also be applied to various languages.

- Nonetheless, using TF-IDF weighting scheme for similarity measure has limitations, because it only preserves frequency of words and does not take the syntactic and semantic information into account.

LexRank and TextRank (1)

- In both TextRank and LexRank, a graph is constructed by creating a vertex for each sentence in the document.
  - https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/
  - https://pypi.org/project/lexrank/

- Origins:
  - TextRank was developed for single-document summarization.
  - LexRank has been applied to multi-document summarization.

- The edges between sentences are based on some form of similarity or content overlap:
  - TextRank uses a measure based on the number of words two sentences have in common (normalized by the sentences' lengths).
  - LexRank uses cosine similarity of TF-IDF vectors.

LexRank and TextRank (2)

- Weighted/Unweighted edges:
  - TextRank uses continuous similarity scores as weights.
  - LexRank uses unweighted edges after applying a threshold to the cosine similarity values.

- In both algorithms, the sentences are ranked by applying a PageRank-like algorithm to the resulting graph.

- Additional features:
  - TextRank does not employ additional features.
  - LexRank score sentences considering other features like sentence position and length using a linear combination with either user-specified or automatically tuned weights.
    - In this case, some training documents might be needed.

Machine learning approaches

- (Supervised) machine learning approaches model the summarization as a (binary) classification problem.

- Sentences are classified as summary sentences and non-summary sentences based on their features.
  - Given a training set of documents and their extractive summaries.

- The likelihood of a sentence to belong to the summary class, or the confidence of the classifier that the sentence should be in the summary, is the score of the sentence.
  - The chosen classifier plays the role of a sentence scoring function.
    - Input ← intermediate representation;
    - Output → score of the sentence.

Some common features

- Position of the sentence in the document.
  - First sentences of news are almost always informative.
- Position in the paragraph.
  - First and last sentences are often important.
- Sentence length.
- Similarity of the sentence with the document title or headings.
- Weights of the words in a sentence determined by any topic representation approach.
- Presence of named-entities or cue phrases from a predetermined list.
- Etc.

Training classifiers

- A problem inherent in the supervised learning paradigm is the necessity of labeled data on which classifiers can be trained.
  - Asking annotators to select summary-worthy sentences.
    - Time consuming.
    - Annotator agreement is low.
  - Abstracts written by people (often professional writers).
    - Used also in abstractive methods.
    - One could compute similarity between sentences in human abstracts and those in the input in order to find very similar sentences, not necessarily doing full alignment.

SELECTING SUMMARY SENTENCES

Introduction

- Most (extractive) summarization approaches choose content sentence by sentence.
  - They first include the most informative sentence, and then if space constraints permit, the next most informative sentence is included in the summary and so on.

- Some process of checking for similarity between the chosen sentences is also usually employed in order to avoid the inclusion of repetitive sentences.

Maximal Marginal Relevance

- One of the early summarization approaches for both generic and query focused summarization that has been widely adopted is Maximal Marginal Relevance (MMR).

- In this approach, summaries are created using greedy, sentence-by-sentence selection.

- At each selection step, the greedy algorithm is constrained to select the sentence that is:
  - maximally relevant to the user query (or has highest importance score when a query is not available).
  - minimally redundant with sentences already included in the summary.

Maximal Marginal Relevance – Issues

- One typical problematic scenario for greedy sentence selection is when a very long and highly relevant sentence happens to be evaluated as the most informative early on.
- Such a sentence may contain several pieces of relevant information, alongside some not so relevant facts which could be considered noise.
- Including such a sentence in the summary will help maximize content relevance at the time of selection, but at the cost of limiting the amount of space in the summary remaining for other sentences.
- In such cases it is often more desirable to include several shorter sentences, which are individually less informative than the long one, but which taken together do not express any unnecessary information.

Global Summary Selection

- Global optimization algorithms can be used to solve the formulation of the summarization task, in which the best overall summary is selected.

- Given some constraints imposed on the summary, such as maximizing informativeness, minimizing repetition, and conforming to required summary length, the task would be to select the best summary.

- Drawback: computationally expensive.

A useful library for Text Summarization

- https://pypi.org/project/sumy/

- Implemented summarization methods:
  - Luhn heuristic method
  - Edmundson heuristic method
  - Latent Semantic Analysis
  - LexRank
  - TextRank
  - SumBasic
  - KL-Sum
  - Reduction - Graph-based summarization

EVALUATIONS

Open issues (1)

- Evaluation of a summary is a difficult task because there is no ideal summary for a document, or a collection of documents and the definition of a good summary is an open question to large extent.

- It has been found that human summarizers have low agreement for evaluating and producing summaries.

- Additionally, prevalent use of various metrics and the lack of a standard evaluation metric has also caused summary evaluation to be difficult and challenging.

Open issues (2)

- In order to be able to do automatic summary evaluation, we need to conquer three major difficulties:
  1. It is fundamental to decide and specify the most important parts of the original text to preserve.
  2. Evaluators have to automatically identify these pieces of important information in the candidate summary, since this information can be represented using disparate expressions.
  3. The readability of the summary in terms of grammaticality and coherence has to be evaluated.

Human evaluation

- The simplest way to evaluate a summary is to have a human assess its quality.

- The factors that human experts must consider when giving scores to each candidate summary are:
  - Grammaticality
  - Non-redundancy
  - Integration of most important pieces of information
  - Structure
  - Coherence

Automatic evaluation methods

- There has been a set of metrics to automatically evaluate summaries since the early 2000s.

- ROUGE is the most widely used metric for automatic evaluation.

https://kavgan.github.io/ROUGE-2.0/

ROUGE

- Lin* introduced a set of metrics called Recall-Oriented Understudy for Gisting Evaluation (ROUGE) to automatically determine the quality of a summary by comparing it to human (reference) summaries.

- There are several variations of ROUGE. The most broadly used are:
  - ROUGE-*n*
  - ROUGE-*L*
  - ROUGE-S
  - Variants of the above-mentioned measures

*Lin, Chin-Yew. "ROUGE: A package for automatic evaluation of summaries." *Text Summarization Branches Out* (2004)

ROUGE-n

- This metric is recall-based measure and based on comparison of n-grams.
- A series of n-grams (mostly two and three and rarely four) is elicited from the reference summaries and the candidate summary (automatically generated summary).
- Let p be "the number of common n-grams between candidate and reference summary", and q be "the number of n-grams extracted from the reference summary only".
- The score is computed as:

$$\text{ROUGE-}n = \frac{p}{q}$$

ROUGE-*L*

- This measure employs the concept of longest common subsequence (LCS) between the two sequences of text.
- The intuition is that the longer the LCS between two summary sentences, the more similar they are.
- Although this metric is more flexible than the previous one, it has a drawback that all *n*-grams must be consecutive.

ROUGE-*S*

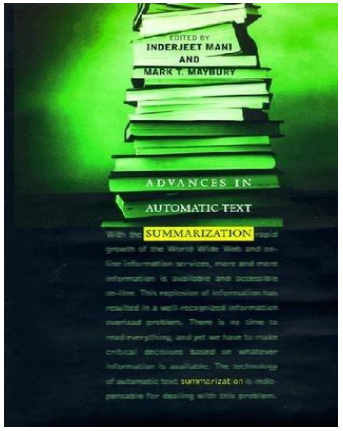
- This metric is also known as skip-gram co-ocurrence ROUGE and considers bi-grams.

- This metric allows insertion of words between the first and the last words of the bi-grams, so they do not need to be consecutive sequences of words.
  - For example, skip-bigram measures the overlap of word pairs that can have a maximum of two gaps in between words.
  - For the phrase "cat in the hat" the skip-bigrams would be "cat in, cat the, cat hat, in the, in hat, the hat".

Useful libraries for TS evaluation

- https://pypi.org/project/sumy/
- https://pypi.org/project/rouge/
- https://pypi.org/project/py-rouge/
- https://pypi.org/project/rouge-score/

Further readings



- Book:
  - Mani, Inderjeet. *Advances in automatic text summarization*. MIT press, (1999)

- Surveys:
  - Nenkova, Ani, and Kathleen McKeown. “A survey of text summarization techniques.” *Mining text data*. Springer, Boston, MA, (2012) 43-76
  - Gambhir, Mahak, and Vishal Gupta. “Recent automatic text summarization techniques: a survey.” *Artificial Intelligence Review* 47.1 (2017) 1-66
  - Lin, Hui, and Vincent Ng. "Abstractive summarization: A survey of the state of the art." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. No. 01. 2019.