Text Summarization

A V Jagannathan
Intern- Suvidha Foundation
Contact: jgan2k03@gmail.com

Contents

1. A word to the readers 3
2. Acknowledgement 4
3. Introduction 5
   - 3.1. What is Text Summarization?
   - 3.2. Need for Text Summarization
   - 3.3. Real life examples for Automatic Text Summarization
   - 3.4. Types of Text Summarization
   - 3.5. Techniques used to implement Extractive Text Summarization
   - 3.6. Techniques used to implement Abstractive Text Summarization
   - 3.7. Commonly used datasets for creating text summarization models
   - 3.8. Scoring methods
4. Techniques used to perform Extractive Text Summarization 8
   - 4.1. History
   - 4.2. How models differ : Intermediate Representation
   - 4.3. Topic Representation based models
   - 4.4. Indicator Representation based models
5. Techniques used to perform Abstractive Text Summarization 13
   - 5.1. History
   - 5.2. Graph based approach & Opinosis
   - 5.3. Rule based approach
   - 5.4. Semantics based approach
   - 5.5. Semantic Graph based approach
   - 5.6. Deep learning based approach
6. Scoring method- ROUGE scores 23
7. My Project 24

Word to the readers

This text was an attempt to compile and give an overview / map of Text Summarization methods. It is scalable in the sense that every topic discussed is accompanied with links to further study and research .Most of the concepts involve various different algorithms and methods ,it is hard to fully cover everything within a few textbook pages and hence is beyond the scope of this book.

At the end of this reading , you will be able to:

- Know what is text summarization , different ways to achieve text summarization, and usefulness and need of the same.
- Understand various techniques used to achieve abstractive and extractive text summarization, with the ability to explain surface-level working of different models
- Understand the scoring and evaluation process
- Understand the working of my personal project

Acknowledgement

I would like to thank my mentor Dr. Manish M Motghare for his help and advice with this project. I Am also tremendously grateful for this learning opportunity that I received during my intern time at Suvidha foundation.

Introduction

What is Text Summarization?

Text summarization is the process of producing a concise , crisp and clear summary from an original text, all the while preserving key information and essence of the original source text. Automatic text summarization is the process of using computers to automate summarization tasks.

Need for Text Summarization

Text summarization is used in various fields constantly, examples include journalism ,academic publications, review systems and so on. A powerful automatic text summarizer will save enormous time for both the reader and writer .

Let's consider a company trying to assess its customer feedback on a product they recently launched. Imagine being in the management team and having to face the trouble of manually going through lines and lines of feedback and drawing little insight from each of them.

It is hard to go through the entire contents of every feedback and draw insights as it is time consuming. The smart thing to do would be to employ a text summarizer to summarize key points in each review, and perform sentiment analysis to categorize it as positive or negative feedback and then take a few random reviews from both classes and draw insights from them.

Automatic Text summarization real-life examples

News platform Inshorts , generates short news articles of size 60 words based on news published that day. They employ AI based solutions for text summarization to produce more than 100K short articles per month.
Source: :
inshorts Introduces AI-based News Summarization on its App, Aims to Generate Over 100K Shorts Per Month

So instead of having to read pages and pages of news, one can keep up with current world happenings just by going through a few 60 word articles.

Types of text summarization

There are 2 types of text summarization techniques, namely extractive and abstractive.

Extractive summarization produces a summary only using words and phrases from the original data, that is, it extracts data from the source and puts it together as a summary, it will not add any new phrases or modify it. Basic techniques and rules can be employed to count the frequency, provide weights to certain words and nouns etc. and produce an extracted summary.

Abstractive summarization produces a summary which consists of words and phrases from original text as well as new engineered phrases which consists of words which may or may not be present in original text. In essence, abstractive summarization generates new summary by taking in insights from original text data and adding on modified data based on original data, meanwhile extractive summarization generates new summary by extracting words and phrases from original data and using them to form meaningful sentences.

Techniques used to implement extraction based text summarization

Extractive text summarization extracts phrases and words from input text to make sentences. An algorithm is developed which in turn ranks these extracted sentences based on their importance . Then as per required size, most important sentences are selected and are made into a paragraph. Approaches to Extractive text summarization differ in how the models extract sentences and rank them.

In this article we will use ETS as an abbreviation of Extractive Text Summarization

Techniques used to implement abstraction based text summarization

Abstractive text summarization techniques use the semantic information of the text to understand the meaning of the document and generate the summary using existing and new phrases.

Commonly used approaches include

1. Structure based approach (using trees,graphs and rules)
2. Semantic analysis based approach (semantic similarity is used to create new text summary)
3. Deep learning based models.

In this article we will use ATS as an abbreviation to Abstractive Text Summarization and ETS as an abbreviation to Extractive Text summarization

Commonly used dataset to evaluate models trained to perform Text Summarization

A huge problem that arises usually is the need for training data. Most models in text summarization are specifically designed to perform certain tasks (scientific journal summarisation etc) . But a commonly used dataset for evaluating the performance of general use models is the CNN/Daily mail dataset.

The CNN/Daily mail dataset has around 300 thousand news articles written in english by journalists at CNN and daily mail. It is free and publicly available on kaggle.com supported by google.

CNN-DailyMail News Text Summarization | Kaggle

Scoring

The type of scoring used to determine accuracy and performance of the model is different from normal methods . The method is called **ROUGE** scoring or **ROUGE** scores. **ROUGE** stands for **R**ecall-**O**riented **U**nderstudy for **G**isting **E**valuation. It is further branched to 3 types of scores namely ROUGE N, ROUGE L, and ROUGE-S scores.

Techniques used to perform Extractive Text Summarization

4.1 History

Extractive Summarization models are older(1980s onward) and much more researched on, than their counterpart ATS. Earlier models exclusively used rule based IF-THEN type algorithms to rank importance of sentences and then use important sentences to form summaries. It was called the "importance evaluator" model. As time progressed more research was put on to developing better ETS models.
Modern ETS models perform the 3 following tasks to generate a summary:

1. **Sentence Extraction** ,a process which extracts sentences from text by observing starting and ending words.
2. **Intermediate Representation ,**a process which aims to construct an intermediate representation of the input text in the form of different sentences and phrases, along with their scores.
    a. **Sentence Scoring ,**a process which aims to score the sentences extracted from input text .
3. **Sentence Selection ,**a process which aims to select and combine k most important sentences to create a summary.

ETS models differ from each other ,from how they achieve step 2 .
We will look at various approaches to Intermediate representation .

4.2 How models differ : Intermediate Representation

Intermediate Representation is the process of representing sentences or words with a score which is usually the importance . There are 2 major categories of intermediate representation .Topic representation and Indicator representation. They are briefly defined below, as are their sub-categories.
Topic representation approaches transform the text into an intermediate representation and interpret the topic(s) discussed in the text.
Topic representation-based summarization techniques are divided into

- frequency-driven approaches
- topic word approaches
- Latent Semantic Analysis (LSA) and
- Bayesian topic models

Indicator representation approaches try to describe the sentences as a function of its features -paragraph position, etc.. rather than describing it as a function of how well they cover the topics. They are divided into

- Machine learning based approaches
- Graph based approaches

4.3 Topic Representation

Frequency driven approach

Sentences are denoted along with their frequency , average word probability or another famous method called TFIDF. This is a form of intermediate representation. There are models designed entirely based on this type of intermediate representation. One such example is the SumBasic model .

SumBasic Model:

The probability of a word w is determined as the number of occurrences of the word, divided by the number of all words in the input.
$p(w_i)$=number of appearances of w_i / total words
Step 1 , Assign a weight to each word w_i , which is its frequency .
Step 2 , Separate the input text into sentences based on starting words and ending words (. , etc)
Step 3 , For each sentence Sj , assign a weight equal to the average probability of the words in the sentence.
Step 4 , Pick the best scoring sentence that contains the highest probability word. Then update the weight for each word in the chosen sentence, Where new weight is the square of its old weight.
Step 5 , Pick the best scoring sentence to be added to the summary.
Step 6 , If the required summary length is reached, stop the process.

Further reading
Beyond SumBasic: Task-Focused Summarization with Sentence Simplification and Lexical Expansion
On TF IDF
tf–idf - Wikipedia

Topic Words Approach
Topic words are usually the most important words in the input text, they are detected by a variety of methods. After extraction of sentences , there are two ways to compute the importance of a sentence, as a function of the number of topic signatures it contains, or as the proportion of the topic signatures in the sentence. Both sentence scoring functions relate to the same topic representation, however, they might assign different scores to sentences. The first method may assign higher scores to longer sentences, The second approach measures the density of the topic words. Then these sentences are represented along with this score, which are then made into summaries.
Further reading
Accurate Methods for the Statistics of Surprise and Coincidence

Latent Semantic Analysis (LSA)

Latent semantic analysis (LSA) is an unsupervised method for creating a representation of text semantics based on words present. LSA has several variations and models built solely from it.

Step 1 , build a matrix A (n x m matrix), where each row represents a word w_i from the input (n words) and each column represents a sentence S_j (m sentences). Element a_{ij} of the matrix A denotes the weight of the word w_i in sentence S_j. The weights of the words are computed by TF IDF technique and if a sentence does not have a word the weight of that word in the sentence is zero.

Step 2 , Singular value decomposition (SVD) is used on the matrix which transforms the matrix A into three matrices: $A = U \Sigma V^T$.

- MatrixU (n ×m) represents a word-topic matrix having weights of words.
- Matrix Σ is a diagonal matrix (m × m) where each row corresponds to the weight of a topic i.
- Matrix V^T is the topic-sentence matrix.
- The matrix $D = \Sigma V^T$ describes how much a sentence represents a topic, thus, d_{ij} shows the weight of the topic i in sentence j.

This form of representation is called LSA representation, it is very useful as it conveys a lot more information than other representations.

Models built using LSA

1. After creating the matrices ,choose one sentence per topic. Therefore, based on the length of summary in terms of sentences, retain the number of topics.The major drawback of this model is that , to convey very important topics, we might need more than 1 sentence. Generic text summarization using relevance measure and latent semantic analysis | Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval

2. Alternative solutions are present to improve the performance of LSA-based techniques for summarization. One such example is the use of weights for each topic, so that the more important topic might use more sentences to express itself when compared to another topic which has less importance.

3. Many other methods are presents , Links to further reading:
    a. Dimensionality Reduction Aids Term Co-Occurrence Based Multi-Document Summarization - ACL Anthology
    b. Text Summarization of Turkish Texts using Latent Semantic Analysis
    c. Two uses of anaphora resolution in summarization - ScienceDirect

Bayesian Topic models

The motivation behind the creation of bayesian topic models involve the following:

1. Pre-existing models consider sentences to be independent of each other which might not always be the case.
2. Sentence scores do not have clear probabilistic interpretations but rather they are greedy and heuristic.

Bayesian topic models solve these issues to an extent. They have better topic representations when compared to other models.

Bayesian models use a distinct type of sentence scoring method called Kullback-Liebler (KL) or relative entropy or I-divergence. It is a measure of how one probability distribution is different from the other.

A simple interpretation of the KL divergence of P from Q is the expected excess surprise from using Q as a model when the actual distribution is P.

$$D_{KL}(P||Q) = \sum_{w} P(w) \log \frac{P(w)}{Q(w)}$$

Where P(w) and Q(w) are probabilities of w in Sentences P and Q respectively. We can represent this in a matrix D where element d_{ij} denotes $D(S_i||S_j)$ where S_k denotes Sentence k. When d_{ij} is 0, it means S_i and S_j are identical , or convey identical info .

This form of representation is used by many models to create an ETS model, one of the most famous one among them is Latent Dirichlet Allocation (LDA).

Links to further reading:

1. (PDF) Discovering Coherent Topics with Entity Topic Models
2. Automatic Summarization of Events From Social Media
3. Latent Dirichlet Allocation
4. Bayesian Query-Focused Summarization - ACL Anthology

4.4 Indicator Representation

Indicator based approaches aim to create the summary by representing sentences as a function of its features ,and not as a function of how well they cover topics and topic words.

There are 2 commonly used Indicator based methods:

1. Graph based methods
2. Machine Learning based methods

Graph based methods

Graph methods use the graph data structure to represent relationships between different sentences, where each vertex is a sentence and each edge between 2 vertices (sentences) denotes the similarity of those two sentences. if similarity between two sentences is greater than a threshold, then they are connected.By similarity we refer to the cosine similarity, with TF IDF weights for words.

- Due to edge creation criteria, a graph is composed of several sub graphs which are not connected to other sub graphs, hence all these sub graphs denote different topics that the text is supposed to convey. The most important message tried to convey could thus be identified by the message behind the subgraph containing the maximum number of vertices.
- Sentences that are connected to many other sentences thus contain the most amount of information in a single sentence as they are similar to many other sentences. Hence are more likely to be included in the summary.
- Graph based methods are versatile and are not limited linguistically since they do not need language specific linguistic processing other than sentence and word boundary detection.

The drawback is using TFIDF weighting scheme for similarity measure, because it only takes frequency of words into account and does not take the syntactic and semantic information into account. Thus, similarity measures based on syntactic and semantic information enhance the performance of the summarization system, like scores based on KL divergence .
Further reading:

1. Chapter 3 A SURVEY OF TEXT SUMMARIZATION TECHNIQUES

Machine Learning Based approaches

Machine learning based approaches turn the extraction problem to a classification problem.For training set, under each text, each sentence is accompanied with several scores which are its feature values and information whether they are included in summary or not. The classification algorithm trains on this and tries to predict the probability that a sentence is included in the final summary or not. This probability serves as a sentence score .Classification algorithms which do not assume sentence independence (Conditional Random field and hidden markov models) perform better than other classification algorithms.
One drawback to using ML based approach is the need for a labeled training set specifically created for that particular task which might not be easily available. Which is countered by methods such as Annotated corpora creation and usage of semi supervised approaches.
Further reading: 1.A trainable document summarizer | Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval
2.Extractive Summarization Using Supervised and Semi-Supervised Learning - ACL Anthology

Techniques used to perform Abstractive Text Summarization

5.1 History

Historical approaches include **sentence compression** , which aims to create a grammatical summary of a given sentence ,**sentence revision**, a method that generates sentences not found in the input document and synthesizes information across sentences.

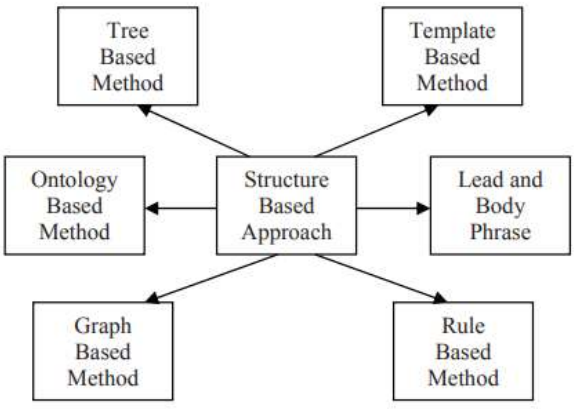
These methods did not perform well compared to their counterpart **Extractive Text Summarization** , which motivated the creation of better & more powerful ATS models.

Modern ATS models perform these 3 tasks to achieve a result:

1. **Information Extraction,** a process that aims to extract important information from the original document.
2. **Content Selection,** a process that aims to select a subset of phrases from the set of phrases extracted in the previous step, to include in the final summary. This step might be performed several times in succession , based on requirements of final summary.
3. **Surface Realization,** a process that aims to combine the candidates from the previous step using grammatical/ syntactic rules to generate a summary.

Modern ATS models can be subdivided into 2 parts:

1. Structure based models
2. Semantic based models



In the structure based models, Graph and rule based approaches are the best performing.

Image source : [(PDF) A survey on abstractive text summarization]

5.2 Graph based models

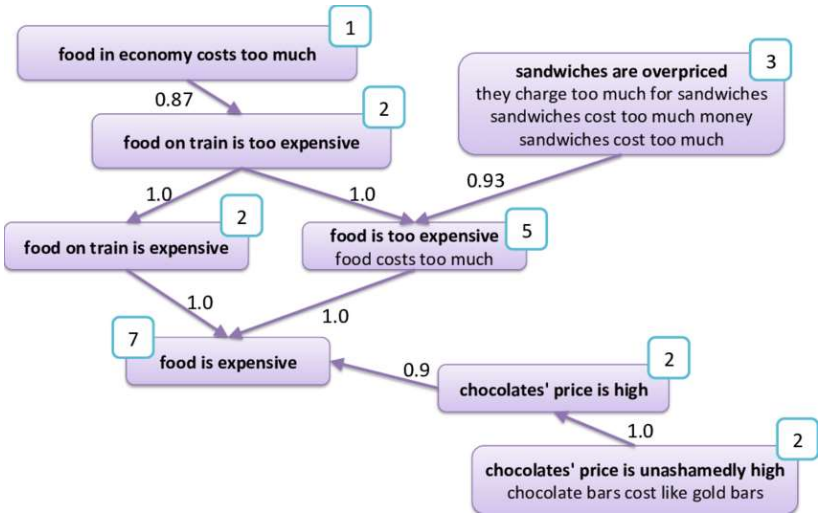
Graph based models make use of the inter-connectivity that a graph data structure provides to create a relation based model between words and phrases , where each node represents a phrase/ event and each edge between 2 nodes represents the semantic relation between them encoded as edge weight.
They offer some advantages over other methods since they have comparatively less weight and are computationally less expensive compared to their deep learning counterparts.
Graph based models are trivial when compared to deep learning models and perform poor when compared to state of the art deep learning models.

Graph based models achieve the 3 steps to ATS, in the following way:

- Step 1,Information Extraction is achieved using methods already available in Extraction based summarization models. Extracted Information , usually in the form of words/phrases are stored in graphs.

- Step 2 ,Content Selection is achieved by making use of entailment graphs, where redundant sentences are removed. In more detail, if two sentences have the same meaning, the less informative of them will be removed and if both have some parts that do not overlap with the other, none of them will be removed

- Step 3, Surface Realization is achieved by traversing through the graph and finding the best path to take. The graph is designed in such a way that if you traverse along the best path , you get meaningful sentences that just need Articles (a , an , the) and Conjunctions (and) etc to create new meaningful sentences.



An example of entailment graph
https://www.researchgate.net/publication/280531093/figure/fig2/AS:614084961640453@1523420672330/Example-of-entailment-graph-for-the-text-exploration-use-case.png
For further study, refer
https://www.researchgate.net/publication/280531093_Entailment_Graphs_for_Text_Analytics_in_the_Excitement_Project

Case study on a graph based model- Opinosis

Opinosis is a model designed by Kavita Ganesan, ChengXiang Zhai, Jiawei Han from the University of Illinois , designed to create a single summary from a set of opinions, hence the name "**Opin**osis".

Procedure:
Step 1: Create the opinosis graph
Each node has 2 values ,

1. Data: a unique word .
2. A list of SIDs and PIDs
    a. SID: Sentence ID , id of the sentence at which the word occurs .
    b. PID: Position ID, the position at which the word occurs in sentence SID.

Eg :
Consider these are two opinions to be represented in an opinosis graph
1.My phone calls drop frequently with the iPhone.
2. Great device, but the calls drop too frequently.

the word **My** as a node will have [SID,PID] as [1: 1] (first sentence, first word).
Completed graph:

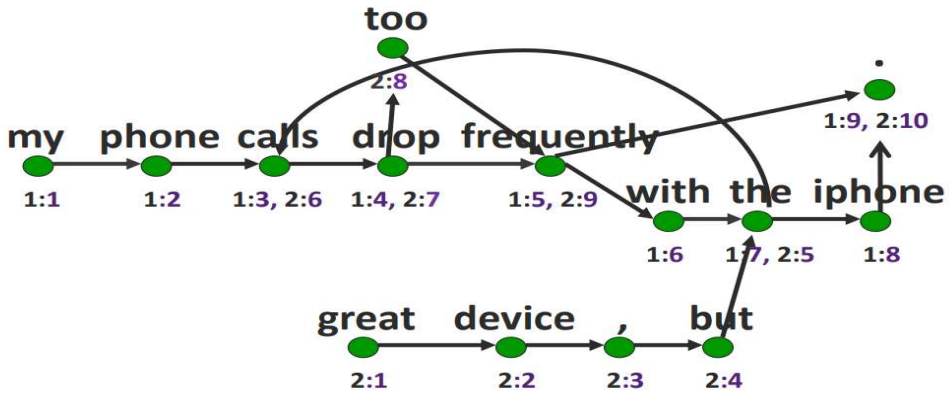


Image source:
Opinosis A Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions

Step 2: Creating a pool of candidate summaries.
Find valid starting points (nouns , pronouns) and ending points (full stop , etc,) , and thus Create a pool of candidate summaries, by traversing through these paths and storing them.

Step 2a: Find collapsible nodes while creating the pool of summaries to collapse 2 different sentences into one, using conjunctions (and, for etc) . Collapsible nodes are denoted in opinosis by observing linking verbs (is, are..)

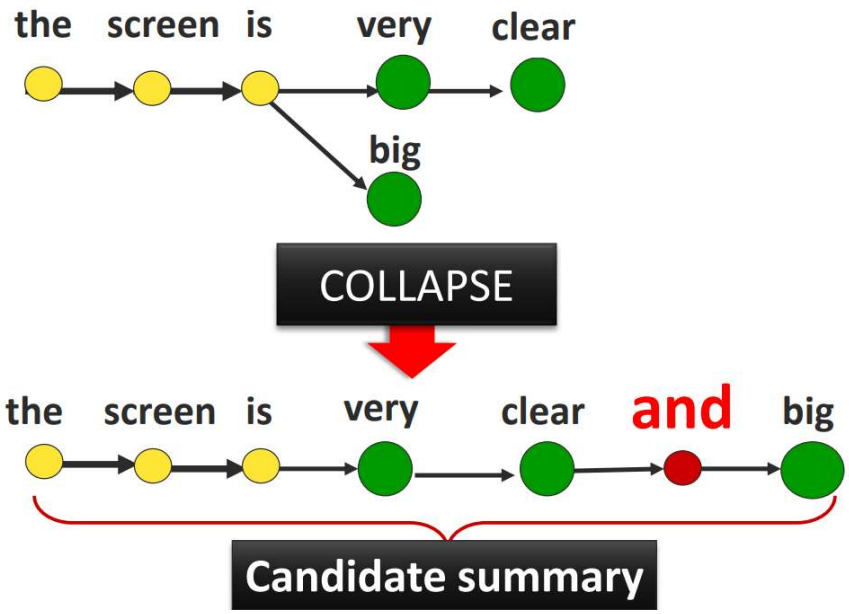


Image source:
Opinosis A Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions

Step 3: Selection of top 2 candidates
Candidates in opinosis are scored by the following criteria:
Redundancy level * length
Redundancy level measurement:

1. # of sentences sharing same path
2. controlled by gap threshold, σgap

Why redundancy level ? a high level of redundancy implies that the path is frequently visited in several sentences hence it carries a level of importance.
Why length? a better length means more coverage hence better summary.
Immediate problem that comes to mind:
If a path covers all nodes then it has the highest redundancy and length , thus it is automatically selected as the best summary which is not optimal.
This is controlled and resolved by taking another variable into factor while calculating redundancy: gap threshold(σgap)
Select top 2 scoring candidates that are most dissimilar. Thus summary is completed.

For Further reading on opinosis:
Opinosis: A Graph Based Approach to Abstractive Summarization of Highly Redundant Opinions - Kavita Ganesan, PhD
For Further reading on graph based models:
Abstractive Text Summarization based on Improved Semantic Graph Approach | SpringerLink

5.3 Rule based approach:

Rule based approaches usually work in the following fashion:

1. Tokenize a paragraph into sentences by observing ending symbols (. , ! ?)
2. Tokenize sentences into words , preprocess them(word stemming etc) and store them along with their position in the original text.
3. After doing this for all sentences, calculate similarity scores across all sentences.
4. Find out rules between words
5. Pick out the important sentences by the help of an importance score
6. Add words from smaller less important sentences which are found to be heavily related to words from similar but more important sentences (found in step 4) and remove their smaller parent sentence .
7. Create meaningful sentences from stemmed word sentences using NLG
8. Arrange the sentences based on their positions in the original text to generate a summary.

To learn more about information scores, similarity scores and to see an actual algorithm , refer
A Rule Based Extractive Text Summarization Technique for Bangla News Documents

PTO

5.4 Semantic approaches :

Semantic means logic or meaning in a language. Semantic based approaches make use of Natural Language Generators to generate a summary based on the logic extracted. The logic extracted is in the form of nouns and verbs.
Steps involved:

1. Extract semantic information from data in the form of verbs and nouns or structures(graphs)
2. Feed to a NLG model which combines these verbs and nouns to make sentences
3. Compile sentences together to form summaries

Semantic approaches can be subdivided into:

- Graph based semantic approach
- Deep learning based semantic approach

NLG

NLG refers to Natural Language Generation . NLG models produce a sentence if provided with nouns ,verbs and the syntactical information between them. Most common NLG models at use are:

- Markov Models
- LSTM based models
- Transformers
    - Generative Pre Trained Transformer (GPT)
    - Bidirectional Encoder Representative Transformer (BERT)
    - XLNet

Explaining about the NLG models is beyond the scope of this project.
Further reading on NLG:
natural language generation (NLG)

5.5 Graph based approach

Graph based approaches make use of the graph data structure to create a map of semantic information of the text , where each vertex is either a verb or a noun and edges between vertices denote the semantic and topological relationship between them.
Graph based approaches usually follow a 4 step process

1. Data Preprocessing
2. Rich Semantic Graph(RSG) generation
3. Rich Semantic Graph(RSG) reduction
4. Summary generation

- Preprocessing module converts the input text into preprocessed sentences which can then be fed into the RSG creation algorithm to create a RSG.

- The RSG generation step is divided into 3 parts
    1. creates a set of RSG subgraphs ranked by their importance.
    2. Selection of set of RSG subgraphs to create a graph
    3. Repeating process 2 over some iterations to create a set of RSGs

  Now the RSGs are ranked based on their importance, and the highest ranked RSG is selected for further processing.

- RSG reduction step involves reducing the selected RSG by deleting, merging or consolidating graph nodes, via heuristic algorithms.
  Suppose S1={SN1,MV1,ON1} and S2={SN2,MV2,ON2} ,
  (SN denotes Subject Noun, MV denotes Main Verb and ON denotes Object Noun)

  Table 5.1 denotes how we can merge S1 and S2 together, under different scenarios only using heuristic rule based algorithms.

- Final step Summary generation, aims to generate a summary from the reduced RSG, it is once again comprised of 4 steps
    1. Text Planning, to convert the given graph into set of sentence objects
    2. Sentence Planning, to convert the sentence objects into semi paragraphs
    3. Surface Realization, to convert semi paragraphs into paragraphs
    4. Evaluation, to select best paragraph out of all to produce a summary

  Domain ontology and WordNet is used to perform summary generation.

	Rule 1	
IF	***SN1 is instance of noun N***	***And***
	SN2 is instance of noun N	***And***
	MV1 is similar to MV2	***And***
	ON1 is similar to ON2	
THEN	***Merge both MV1 and MV2***	***And***
	Merge both ON1 and ON2	
	Rule 2	
IF	*SN1 is instance of subclass of noun N*	*And*
	SN2 is instance of subclass of noun N	*And*
	{ [MV11,ON11], .. [MV1n,ON1n] } is similar to	
	{ [MV21,ON21], .. [MV2n,ON2n] }	
THEN	*Replace SN1 by N1 (instance N)*	*And*
	Replace SN2 by N2 (instance N)	*And*
	Merge both N1 and N2	
	Rule 3	
IF	*SN1 and SN2 are instances of noun N*	*And*
	MV1 is instance of subclass of verb V	*And*
	MV2 is instance of subclass of verb V	*And*
	ON1 is similar to ON2	
THEN	*Replace MV1 by V1 (instance V)*	*And*
	Replace MV2 by V2 (instance V)	*And*
	Merge both V1 and V2	*And*
	Merge both ON1 and ON2	
	Rule 4	
IF	*SN1 and SN2 are instances of noun N*	*And*
	MV1 is similar to MV2	*And*
	ON1 is instance of subclass of noun NN	*And*
	ON2 is instance of subclass of noun NN	
THEN	*Merge both MV1 and MV2*	*And*
	Replace ON1 by NN1 (instance NN)	*And*
	Replace ON2 by NN2 (instance NN)	*And*
	Merge both NN1 and NN2	

Table 5.1

Further reading:

1. (PDF) Semantic graph reduction approach for abstractive Text Summarization
2. Abstractive Text Summarization based on Improved Semantic Graph Approach | SpringerLink

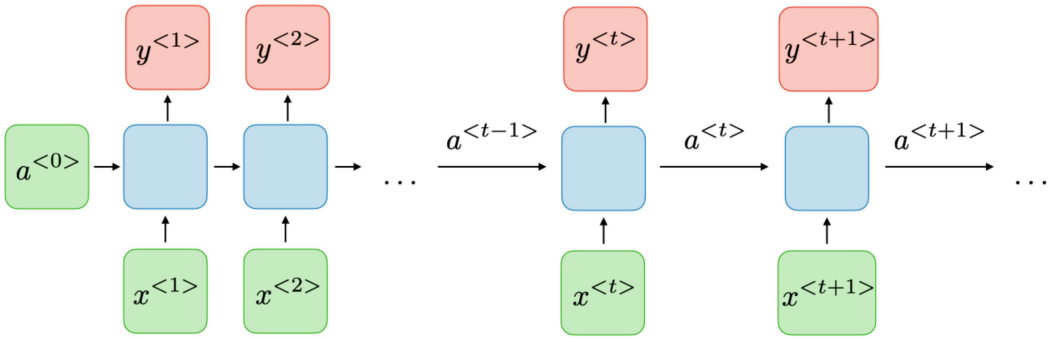
5.6 Deep Learning based approach

Deep learning methods as the name suggests involve variations of ANNs such as RNNs ,and LSTM based encoder-decoder models to achieve abstractive text summarization. We will look at the working of an encoder-decoder based Seq2Seq model as they are used in almost every state of the art text summarizer.

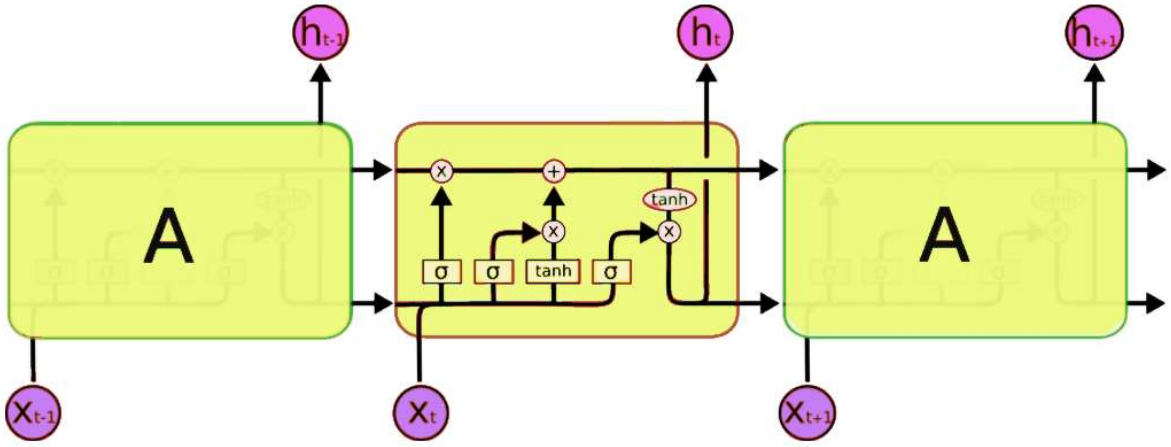
An RNN is a special type of ANN(What are Neural Networks? - United Kingdom | IBM.) which takes in the parameters:

Input for state i : $<x_i>$

memory parameter for state i from the previous state : $<a_{i-1}>$



RNN architecture. Image source:CS 230 - Recurrent Neural Networks Cheatsheet



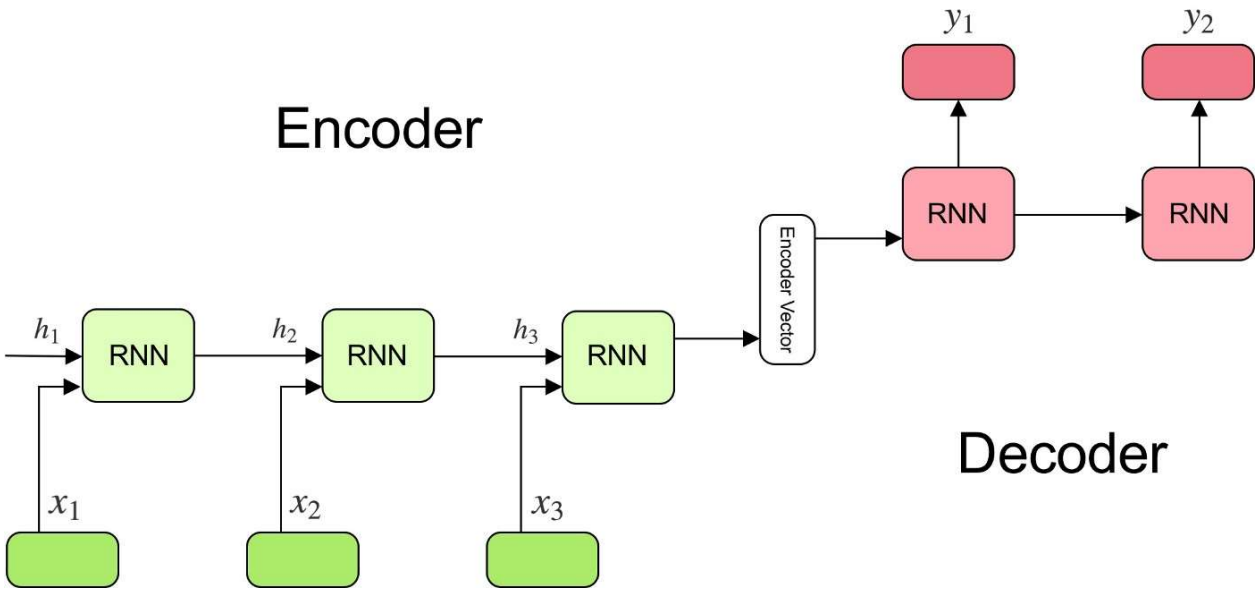
LSTM architecture. Image source: Long Short Term Memory | Architecture Of LSTM

An LSTM(Long Short Term Memory) is a special type of RNN which eliminates the vanishing gradient problem(Vanishing gradient problem | Engati.) encountered in an RNN. All the above architectures denote a case where the number of outputs are the same as inputs

- .An NLP task like summarization will definitely need a lesser number of outputs as compared to inputs,i.e in some cases we need to design a many to many RNN/LSTM in which number of outputs is different from number of inputs

- In many cases regarding an NLP task(like generating a sentence), we need information about the next states as well the information about the previous states. In those scenarios weight matrices of previous states have to be adjusted based on input of current states.

To solve this a type of LSTMs/RNNs called Encoder-Decoder is used. The encoder takes in inputs and encodes it into a matrix called encoder vector. The decoder then decodes this encoder vector to produce meaningful result.



Enc-Dec architecture. Image source: Understanding Encoder-Decoder Sequence to Sequence Model | by Simeon Kostadinov | Towards Data Science

An animated illustration: Encoder-Decoder."tf-seq2seq is a general-purpose encoder-decoder framework for Tensorflow that can be used for Machine Translation, Text Summarization, Conversational Modeling, Image Captioning, and more."(From Tensorflow api)

General deep learning models for summarization will perform text cleaning, feed the cleaned text into a seq2seq model and the result will be a summary. Error is computed via ROUGE scores and minimized via backpropagation.State of the art model like PEGASUS finds out the important sentences of a text , then feeds it into a seq2seq model which then produces a summary.

Scoring methods

Text summarization needs a specific type of scoring as the task performed is very specific and involves many variables .

ROUGE

Rouge is an abbreviation of Recall-Oriented Understudy for Gisting Evaluation. Main idea behind Rouge scores is the use of Recall. Let S_g be the Summary generated by an automated procedure and S_r be the reference summary, usually written by humans.
in context of text summary Recall is defined as
(Number of overlapping words between S_r and S_g) / (Total words in S_r)
Recall produces a value between 0 and 1 , Recall* 100 denotes the percent of words in the original summary represented in the generated summary. A recall of 1 is very good but at the same time, the summary generated might be very very long which also then covers all S_r words hence getting high recall.
To combat this, Precision is used. Precision is defined as
(Number of overlapping words between S_r and S_g) / (Total words in S_g)
High precision and recall implies a better summary, thus combining both ,we generate a new score called F1 score:
F_1 Score: $2 * (precision * recall)/(precision + recall)$
F_β Score: $(1 + \beta^2) * ((precision * recall)/((\beta^2.precision) + recall)$

Unigrams Bigrams and Polygrams :
Unigram is the set of all words taken separately. Bigram is the set of 2 words, where each 2 words are produced by taking each word in the summary along with the word after it, it is similar for trigrams(3 words) and polygrams.

- Rouge N denotes Rouge score for taking overlaps of N-gram.
- Rouge L denotes Rouge scores for longest matching sequences (sequences means there need not be consecutive matches)
- Rouge S denotes Rouge scores for bigrams with a maximum allowed word gap between them.

Further reading: ROUGE: A Package for Automatic Evaluation of Summaries - ACL Anthology

Human evaluation is also used to score the summaries.

My Project

My project is an Abstractive text summarizer built to summarize news articles. Inspired by SumBasic and PEGASUS models. Text Summarization is achieved in a total of steps as follows:

1. **Data Cleaning:** text from news articles contains various information apart from just the article, like web pages on which the article is published, publishing time , location etc.Text cleaning aims to remove these along with stopwords, and convert the article's letters into fully lowercase.
2. **Data Engineering:** each and every word from the cleaned data is then taken separately and functions are designed to provide information about their features. These features include:
    a. Absolute Score: Absolute score(x), where x∈words, denotes the probability of the picked word being x, when you select a random word from the given set of words.
    b. Relative Score: Relative score(x), where x∈words, is computed like this:
    $Relative\ score(x) = Absolute\ score(x) \times Sentence\ Score(most\ prominent(x))$
        i. Sentence Score(sentence): gives the average Absolute score of the sentence ,that is $\sum Absolute\ score(x)/N$ where x∈words(in sentence) and N is the number of words in the sentence.
        ii. Most prominent(x) : where x∈words, gives the sentence in which x has the most appearance.
    c. inFirst : inFirst(x) where x∈words, will give a value 1 or 0, based on whether the word x is present in the first sentence of the article. At times this is important as the first sentence in most articles tends to provide an overview about the rest of the article.
    d. SummaryLength: SummarLength(x) where x∈words, is either input from the user denoting required length of the summary , or length of the summary of the article (in case of supervised training).
3. **Classification:** A classification model(Random Forest is used) then predicts the probability of a word being included in the summary or not based on the set of engineered features.
    a. Training: pre existing training data taken from CNN daily mail dataset is cleaned and engineered and a label is provided whether the word is in summary or not.The model trains on this.
    b. Testing: given a word, the model predicts the probability of the word being present in summary.
4. **Generating a summary:** The words are then sorted in descending order based on their probability , then the top n words are taken and then fed to a t5 transformer designed to create sentences from given words (where n=70% of required summary length). Then the generated sentences are fed to a grammar correction model, which provides a grammatically correct summary.

Addressing the issue of poor performance:

The model performs mediocre-poor most of the time compared to State of the art summarizers. That is because our model is pre trained on 70000 articles , while most of the summarizers available are trained on almost a billion articles.