Text Summarization

Heriot Watt University , Dubai Campus
2019/2020

Bruce Tauro – H00228269
Mohamed Serry - H00313456
Salah Tamer - H00343334
Tarek Itani - H00292565

summary

/ˈsʌm(ə)ri/

noun

1.a brief statement or account of the main points of something.

2."a summary of Chapter Three"

3.Similar:

4.synopsis

5.precis

6.résumé

7.abstract

8.abridgement

9.digest

"*Automatic text summarization is the task of using computers to produce a concise and fluent summary while preserving key information content and overall meaning* "

Text Summarization Techniques: A Brief Survey 2017

“I apologize for such a long letter - I didn't have time to write a short one.”

— Blaise Pascal

History

- Initially pioneered by Hans Peter Luhn in 1950 at IBM.
- Existence and availability of internet
- Increase of amount of data

Need:

⬆ Important data ⬇ time

Applications of Text Summarization

- Marketing Search – SEO / Social Media
- Chatbots (QnA)
- Legal Contract Analysis
- Books / Document Summarization
- Media Monitoring
- Text Classification

Types of Text Summarization

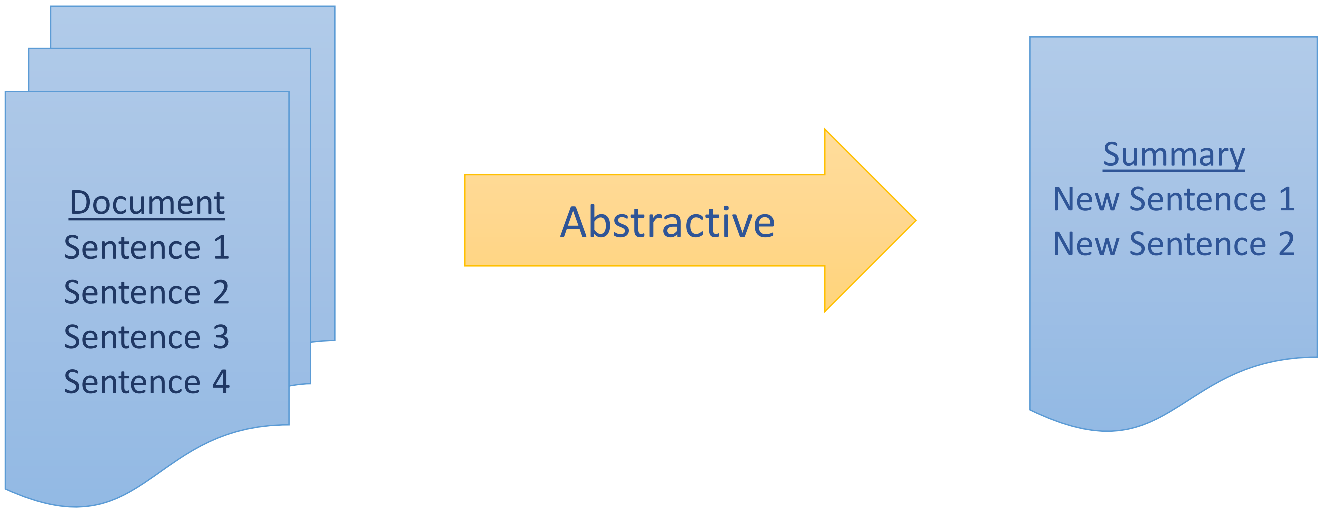
Extractive Summarization: Extractive summarization rely on extracting content in the form of pieces text and concatenating them to create a summary

Abstractive Summarization: The abstractive summarization generate entirely new text from the original one, to the extent that some parts of the generated text are not in the original corpus .

Abstractive Text Summarization

Abstractive Summarization

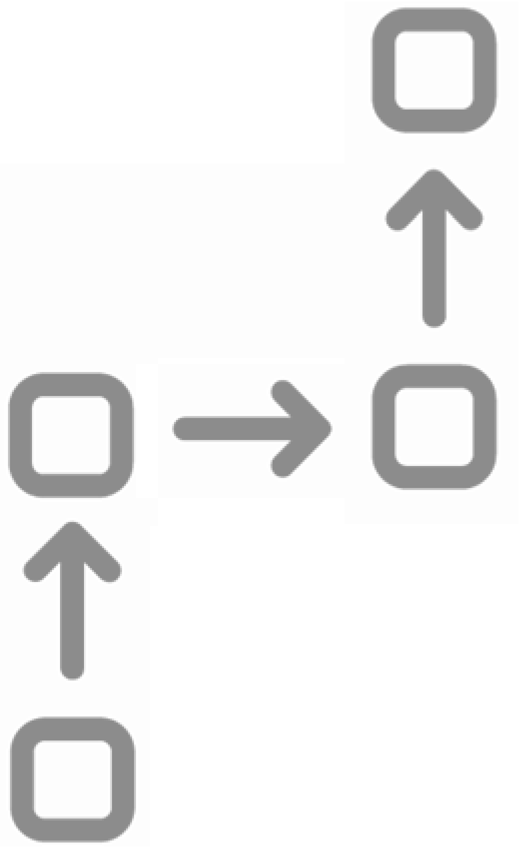
- Generate new sentences as a summarization
- Sentences do not exist in original text
- More human readable than extractive text summaries.
- Known as a Sequence to Sequence model (Many to Many)



Abstractive (How it is done?)

It follows an Encoder-Decoder Architecture

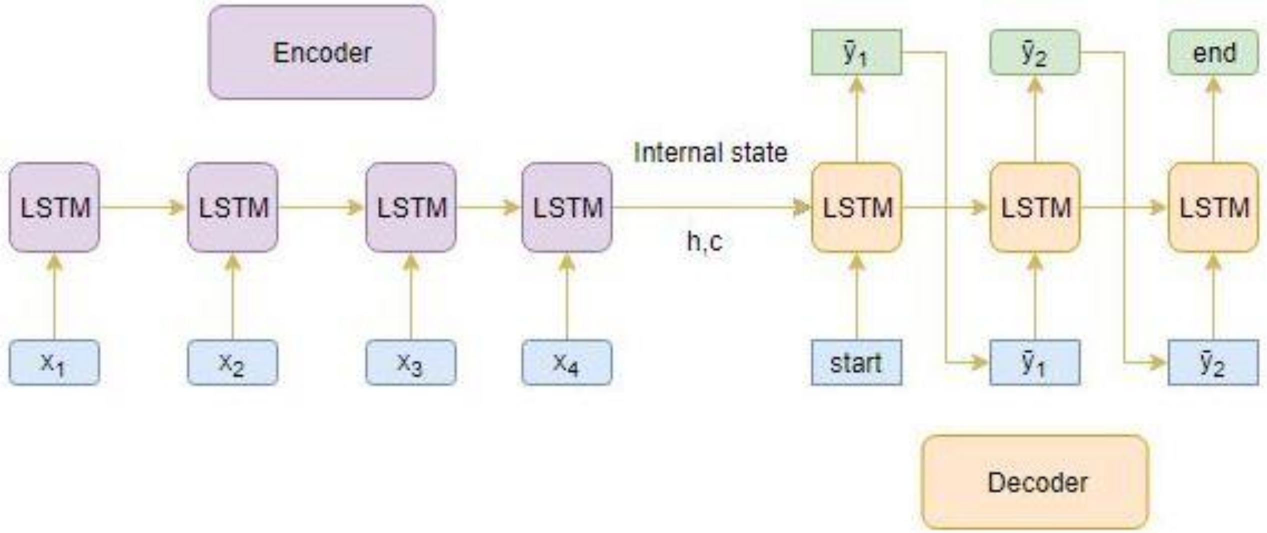
1. Creates a Semantic/Structured representation of the text (Encoding)
2. Recreates the sentences from the Semantic/Structure representation (Decoding)
3. Once the network is trained it can be tested on text to evaluate it



Abstractive (Encoding/Decoding)

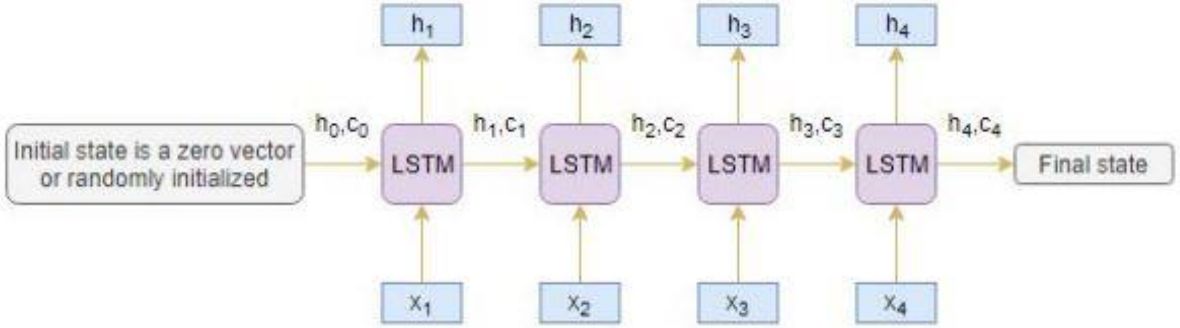
Several types of neural networks are used as Encoders/Decoders:

1. Recurrent Neural Networks (RNNs)
2. Convolutional Neural Networks(CNN)
3. Gated Recurrent Neural Network (GRU)
4. Long Short-Term Memory (LSTM)



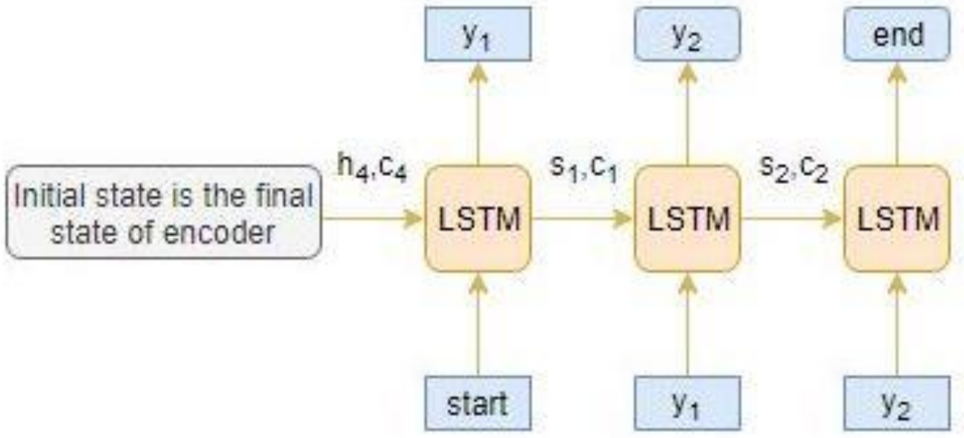
Abstractive Encoding (How it is done?)

1. A chain of the selected Neural networks (Node) are linked together
2. Each node takes an initialization value and the next word in the sentence.
3. The nodes feed their outputs as the initialization value for the next node.



Abstractive decoding (How it is done?)

1. A chain of the selected Neural networks (Node) are linked together.
2. Each node takes the output initialization value of the encoder chain as the initialization value for the decoder.
3. The decoder is given a Start token and predicts the next token as its output. This predicted token is taken as the input for the next node.
4. The sentence is summarized when a defined end token or word limit is reached.

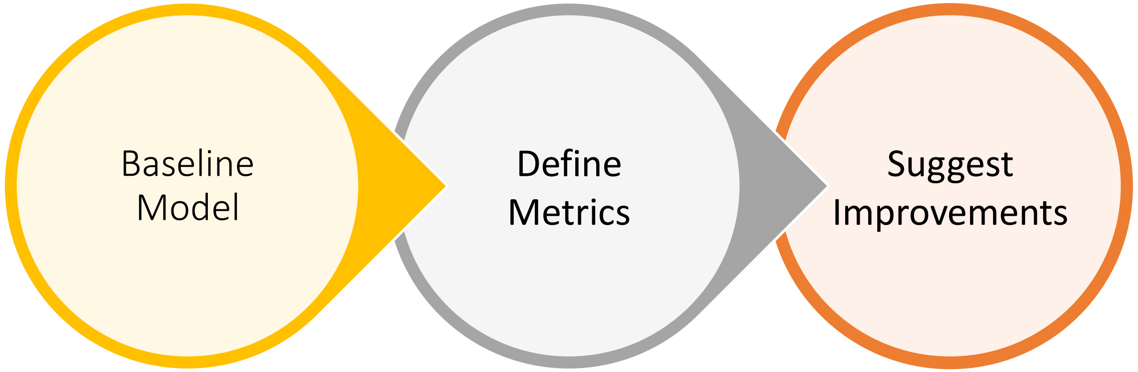


Abstractive Text Summarization

Soumye Singhal
Department of Computer Science
IIT Kanpur
soumye@cse.iitk.ac.in
Arnab Bhattacharya
Department of Computer Science
IIT Kanpur
arnabb@cse.iitk.ac.in

Abstractive Text Summarization

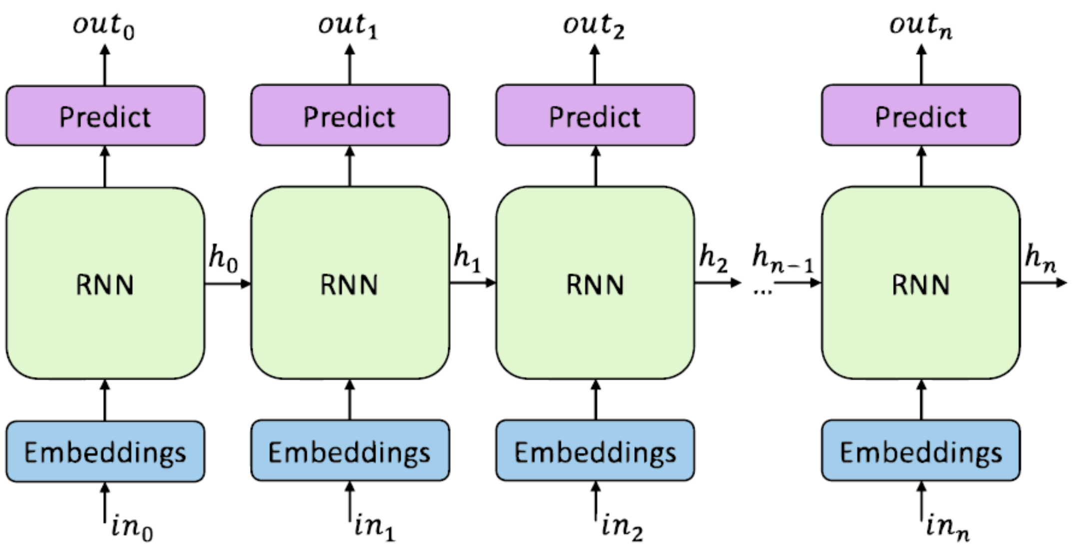
"Neural Sequence to Sequence attention models have shown promising results in Abstractive Text Summarization. But they are plagued by various problems. The summaries are often **repetitive** and **absurd**. We explore and review **different techniques** that can help overcome these issues."

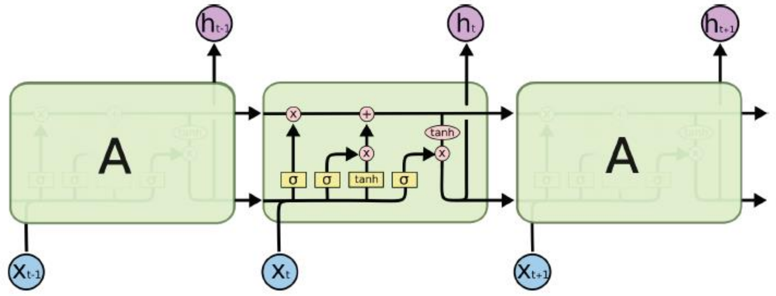


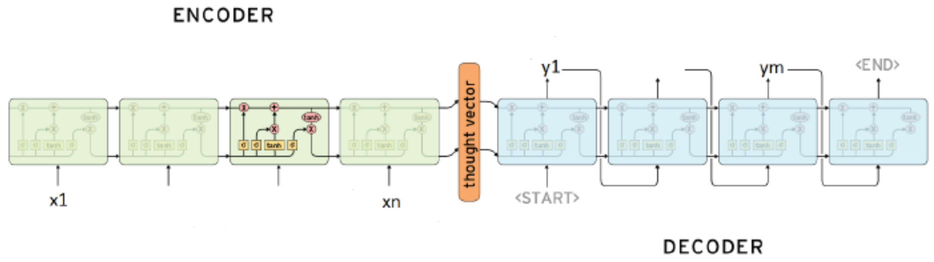
Baseline Attention Model

Baseline Model is a Neural attention Model with Encoding – Decoder implementation, where the text is encoded into hidden layer and then the decoder decodes the hidden layer to produce the summary

Bi-Directional RNN - LSTM







Metrics

Grammatically correct and Human readable

Metrics…

Target Summary Present

ROUGE : ROUGE is simply a string-matching Metric

ROUGE-N measure N-Gram similarity ,
ROUGE-L which measure Sentence level similarity and
ROUGE-S which is Skip-gram Similarity

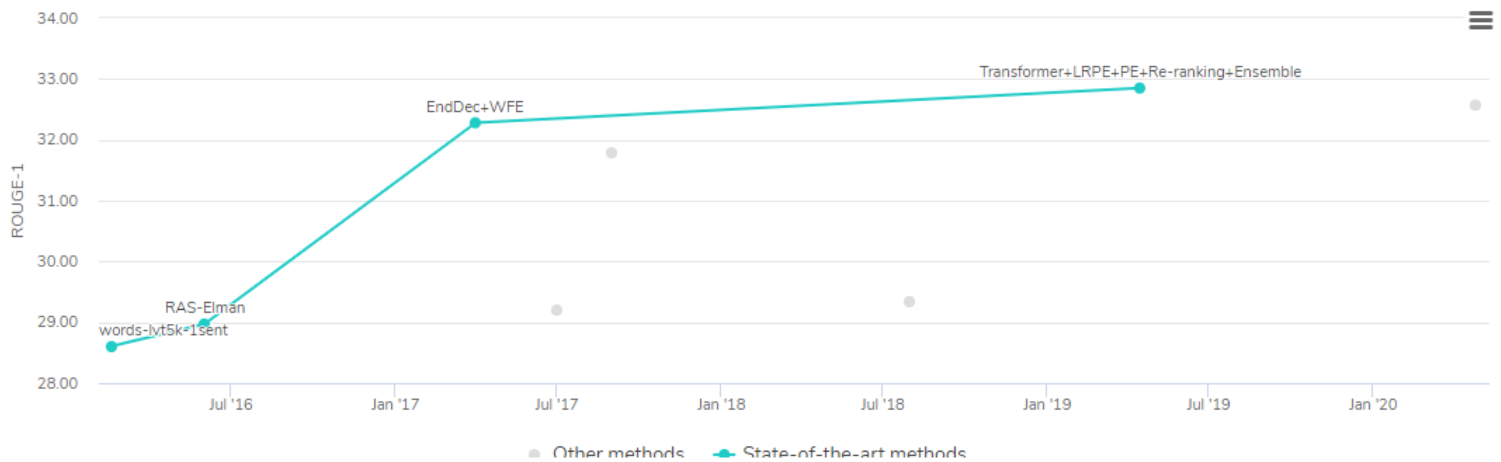
Target Summary Absent

Topic Modeling

Datasets

- DUC-2004
- Gigaword

Text Summarization on DUC 2004 Task 1



Suggested Improvements

LARGE VOCABULARY

HIERARCHICAL ATTENTION

POINTER GENERATOR NETWORK

COVERAGE MECHANISM

INTRA-ATTENTION ON DECODER OUTPUT

LEARNING FROM MISTAKES USING REINFORCED LEARNING

Large Vocabulary

Use more linguistic rich features for the input like POS (Part of speech), named-Entity and TF-IDF Speeds training , yet surprisingly decrease abstractive Capabilities

Author didn’t share exactly the training speed gained vs scoring in the metric lost.

Worth mentioning that we faced the same on Coursework1

Hierarchical Attention

The team recommends the use of **Hierarchical attention**, no detail on its metric improvement score

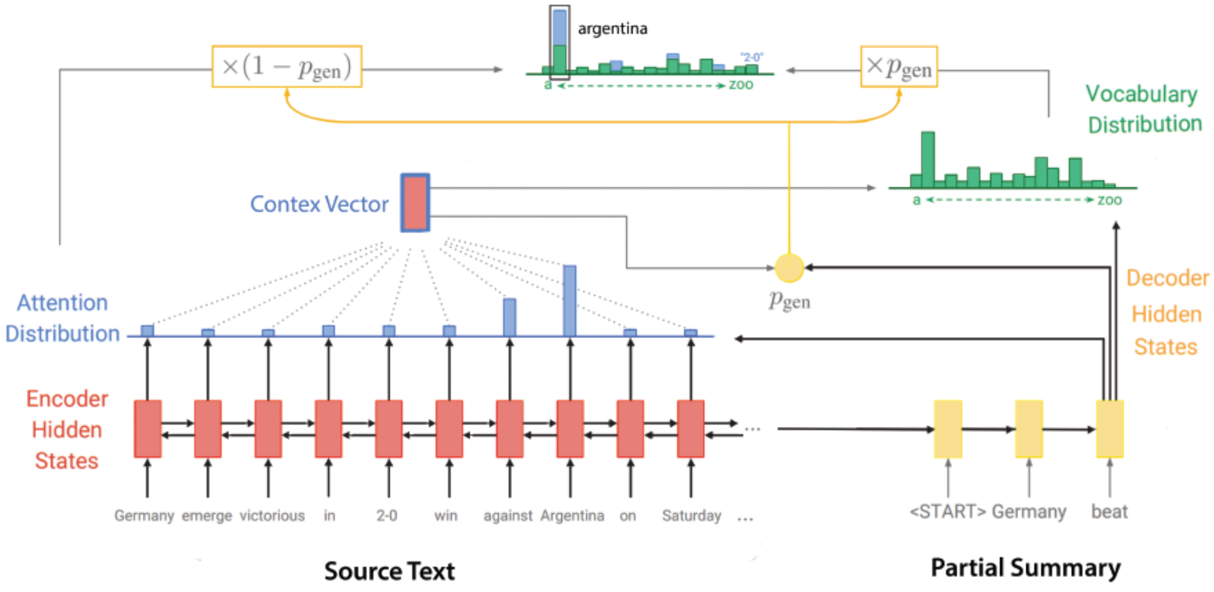
In **Hierarchical Attention Networks for Document Classification** paper, it proposed hierarchical attention networks (HAN) , obtained better visualization using the highly informative components of a document. The model progressively constructs a document vector by aggregation of words into sentence and then sentences into document..

Hierarchical Attention Networks for Document Classification, ichao Yang1 , Diyi Yang1 ,

Pointer Generator Network and Coverage Mechanism

Solves the **out of vocab problem** (UNK) by copying from (Pointing) to the source while avoiding repetition

Coverage model is simply done by Summing all the attention and penalizing the things that already been covered



In Advances in Neural Information Processing Systems 28 (NIPS 2015) paper, addressed the challenge of number of target classes in eachstep of the reliance of the output on the length of the input,

Advances in Neural Information Processing Systems 28 (NIPS 2015)

Intra-Attention on Decoder Output

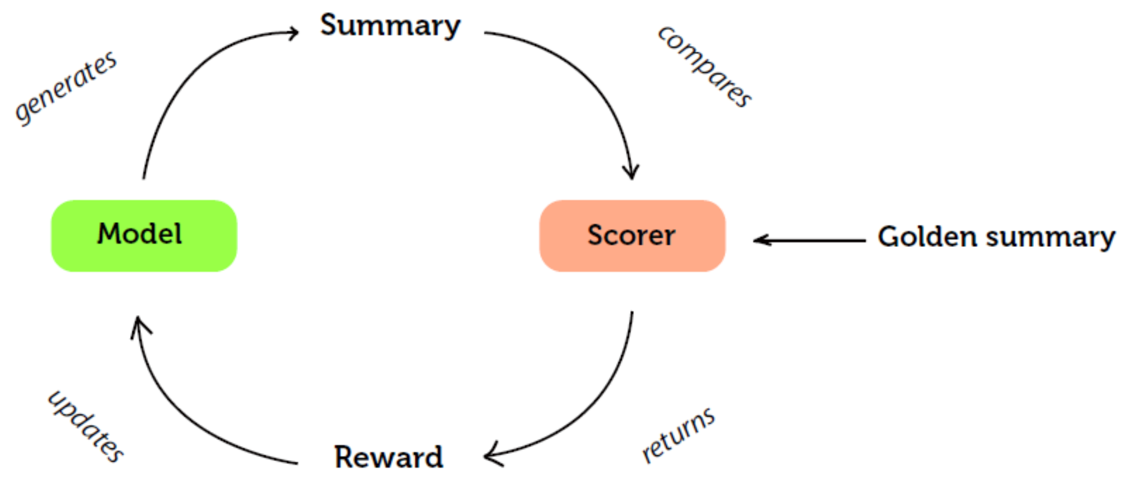
Same like Coverage Mechanism but consider also Decoder output , this avoids repeating words that has been generated already

Learning From Mistakes using reinforced learning

We sample the next word from the output of the previous step
The Major challenge in Text Summarization between testing and real implementation is real world problem.

Note :All the text we use (Like in our Class) are correct (aka Shakespear)
The challenge in real word is incorrect text , and how the model can recover

The model output is compared to the reference summary using ROUGE metric , then iterate using Reinforced learning till we get a high score ROUGE score



Current Challenges

Metric

ROUGE and text Similarity is not a good measure for abstractive Text Summarization especially how our brain interpret a good summary

Datasets

Most of the Datasets available online are news, where you can get a relatively good summary by considering the top sentences

Our Findings

No insights on results

The authors in many ways didn’t provide detailed insights of the recommendations or the experiment.

Metrics

we do have a major concern on the Metrics used, we believe that the use of ROUGE and topic modeling is not suitable for measuring the objective and improvements they mentioned in the purpose of the paper especially if there is error in the given text. The Author also acknowledged this.

Challenges with Abstractive models

- The trained model is limited to its known vocabulary (The text it is trained on) and may not be able to summarise important Out Of Vocabulary (OOV) words.
- Salient words might not be detected during training leading to inaccurate summaries.
- Limited quality datasets to train the network leads to above challenges
- Summaries are limited by needing start and end tokens to be defined or setting sentence length limits.

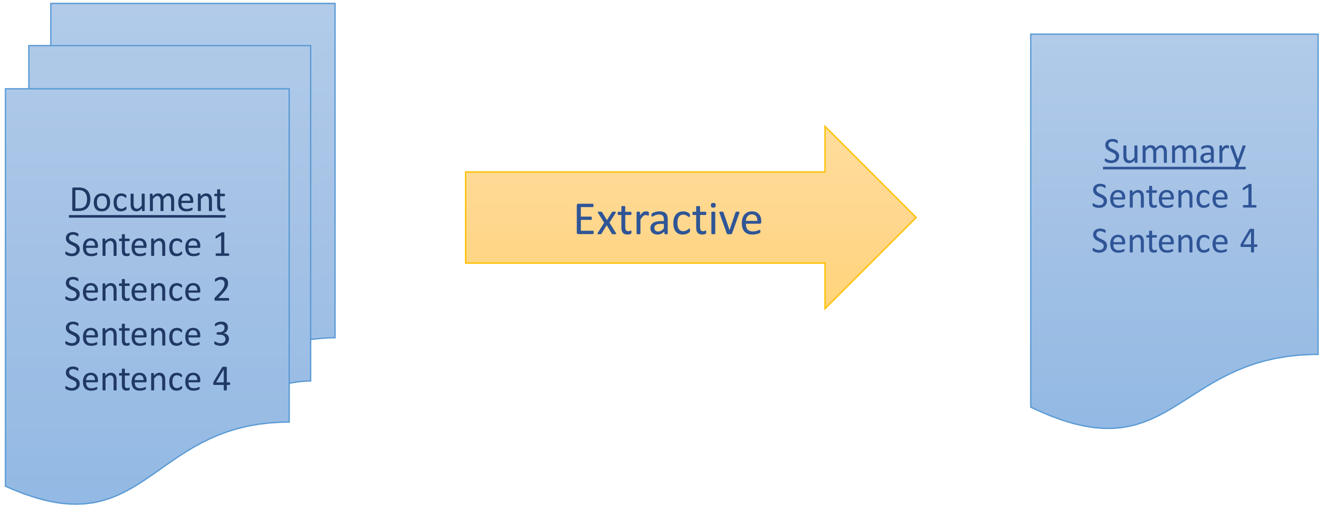
Focus on technology

To prepare for this classwork we have went through several research papers, we found that most of the papers ,including this one, focus on the technology and not the user , none of them suggested any improvements based on the user preferences, no consideration to user location, mother tongue or preferences . we believe text summarization should incorporate and be tuned for other parameters and including the context of the original text and adjust the model parameters accordingly


Extractive Text Summarization

Extractive Summarization

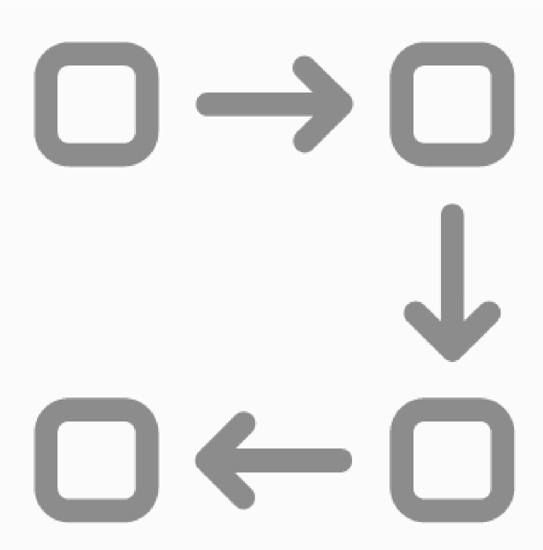
- Extracting important complete sentences from text
- No change in the sentences extracted



Extractive (How it is done?)

It is done in 3 phases:

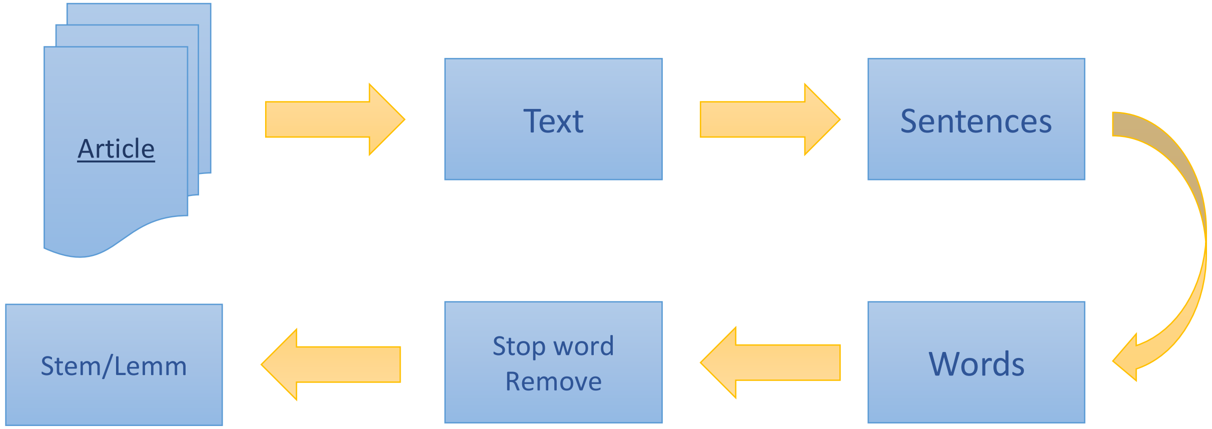
1. Build representation of the text (Preprocessing)
2. Score sentences in built representation
3. Select k most important sentences to be our summary



Extractive (Preprocessing)

This phase involves four stages:

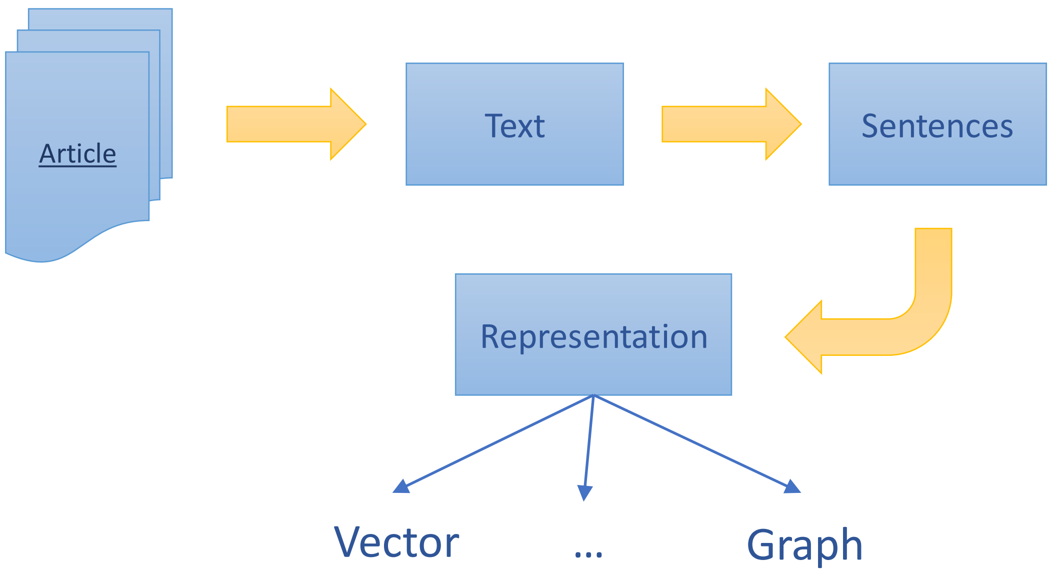
1. Sentence Segmentation
2. Tokenization
3. Stop word Removal
4. Stemming/Lemmetization



Extractive (Representation)

Most common approaches:

1. Frequency based representation
2. Semantic similarity representation
3. Vector similarity representation
4. Graph based representation



Extractive (Scoring)

After phase 1, sentences are scored based on factors such as frequency, semantics, similarity, position of sentence or word.

Extractive (Scoring) Contd.

Frequency based Scoring

If the number of times a word occurred in a document is high, then it has importance in the content of that document. Thus higher score.

Common methods used:

- Word Probability
- Bag of Words (BoW)
- Term Frequency - Inverse Document Frequency (TF-IDF)

Frequency based Scoring (Word Probability)

Is the number of occurrences of a word divided by the total number of words.

$$\text{P(w)} = \frac{freq\ of\ word}{Total\ num\ of\ words}$$

Frequency based Scoring (Bag of Words)

Bag of words is a vector with size equal to all words in our document. A BoW representation of a sentence is the the same vector with frequencies of words occurred in this sentence.

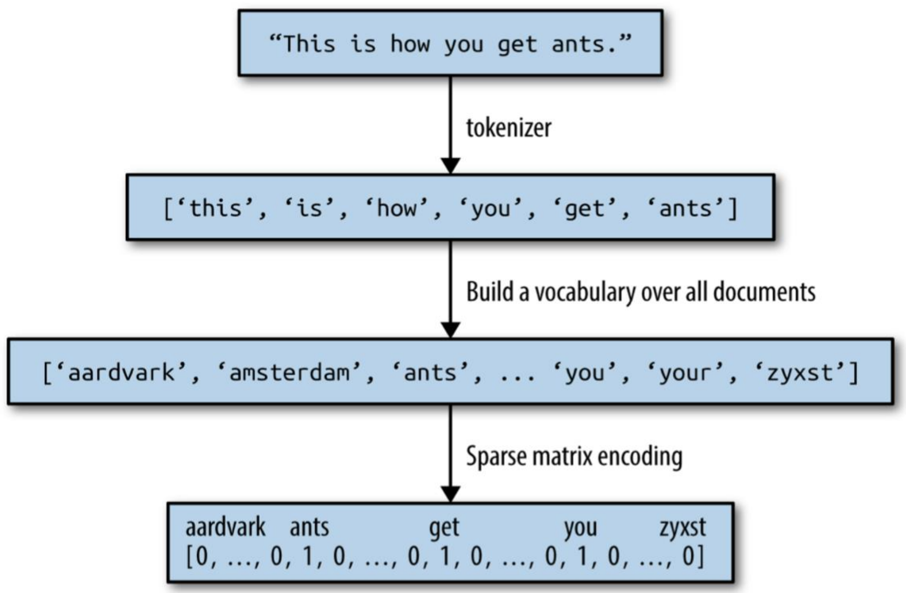
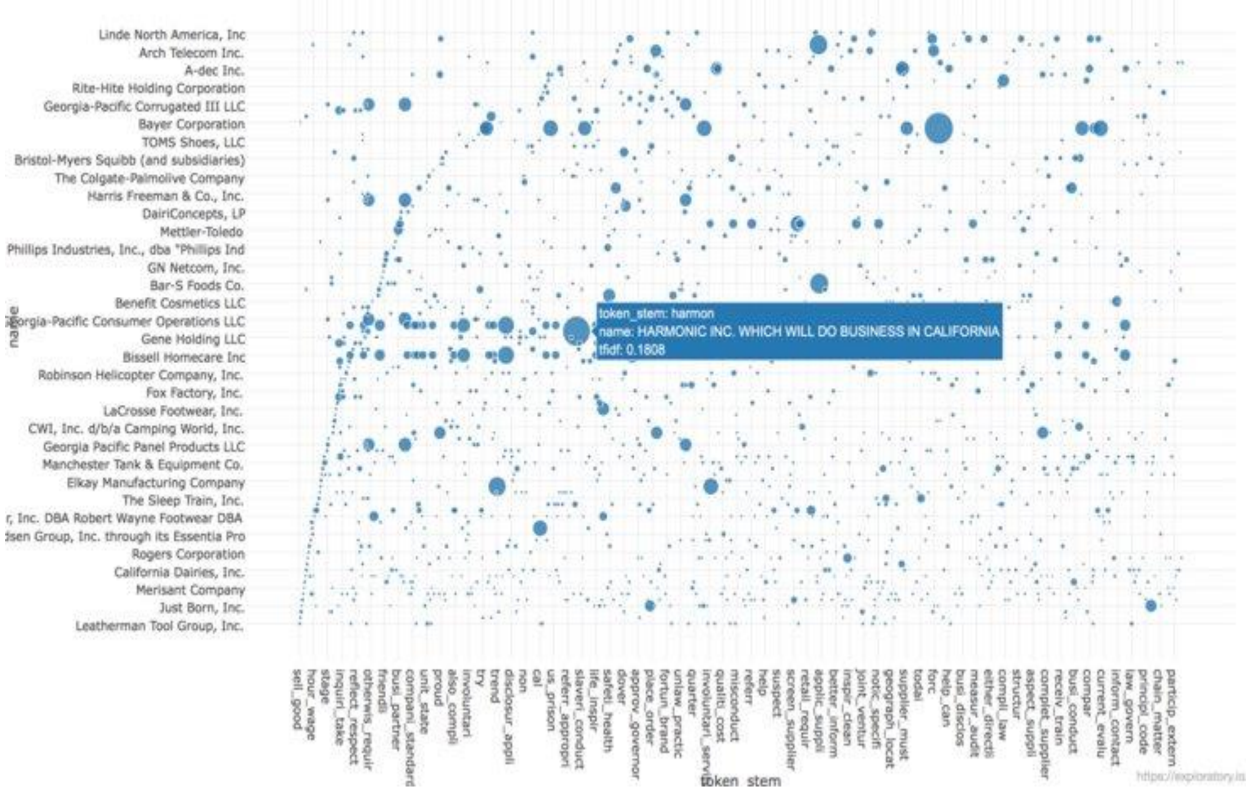


Figure 7-1. Bag-of-words processing

Frequency Based Scoring (TF-IDF)

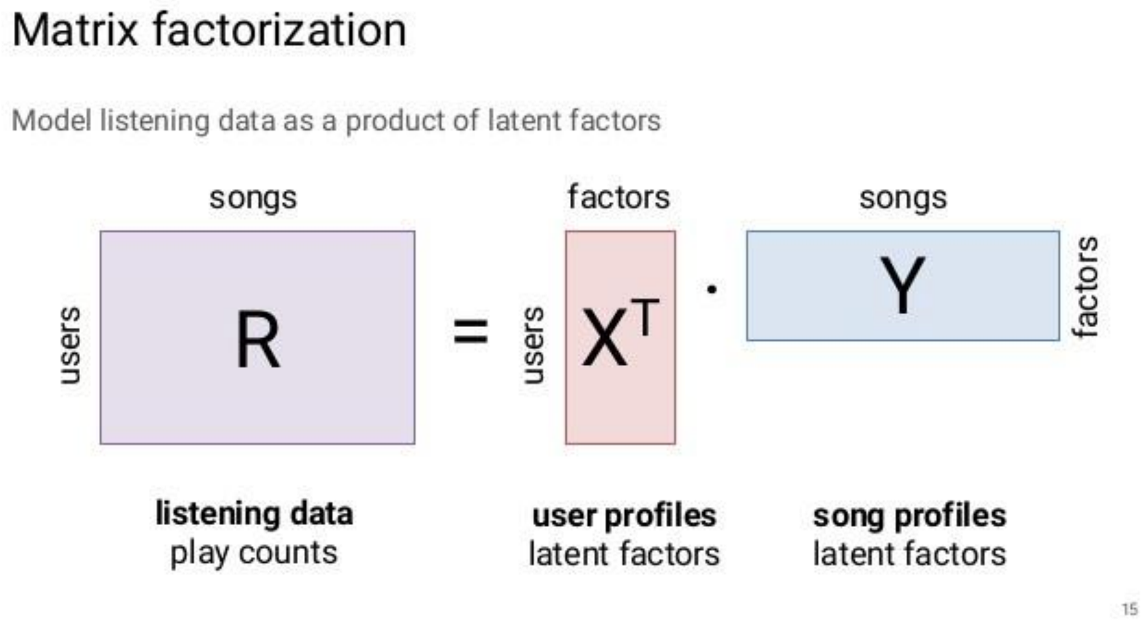
- TF: number of documents contain t
- IDF: total number of documents divided by documents containing t



Extractive (Scoring) Contd.

Semantic based Scoring

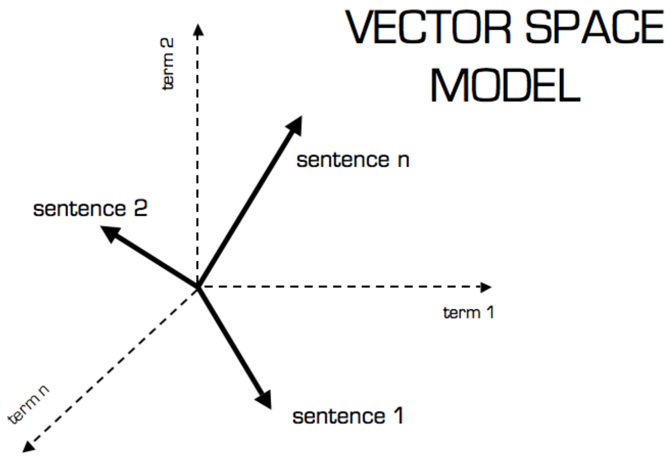
LSA



Extractive (Scoring) Contd.

Similarity based Scoring

Vector rep. of sentence



Euclidian dist or cosine dist.

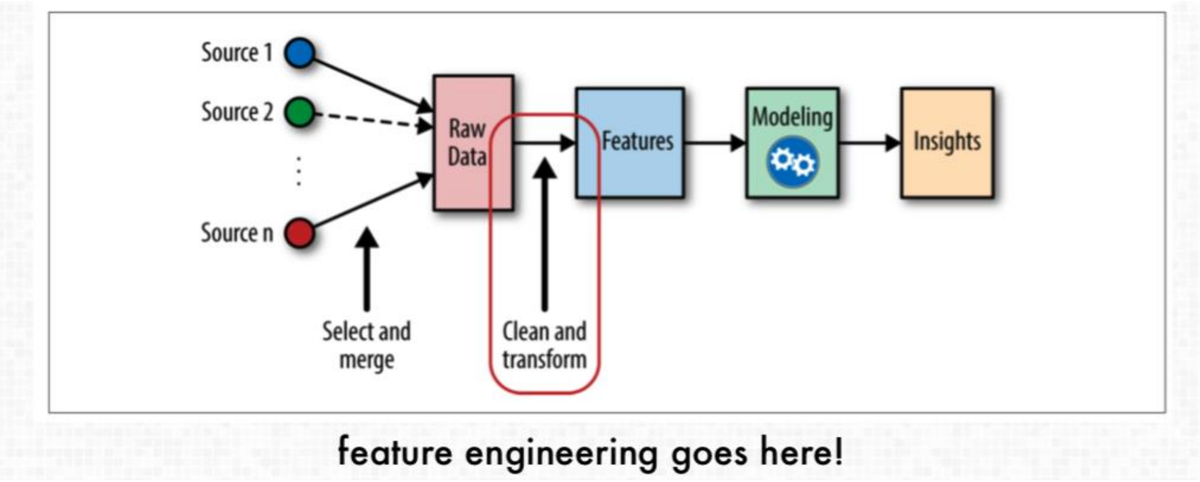
Centroid based similarity

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Extractive (Scoring) Contd.

Feature Based Scoring

Hand authored features



Domain specific features

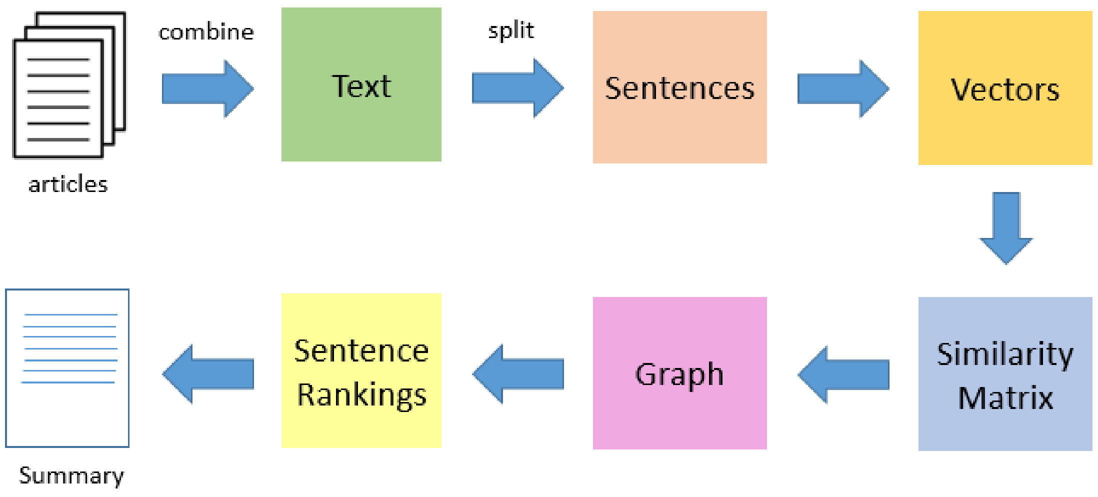
Extractive (Generation)

Sentence Ranking

Centroid based ranking

Accumulative ranking

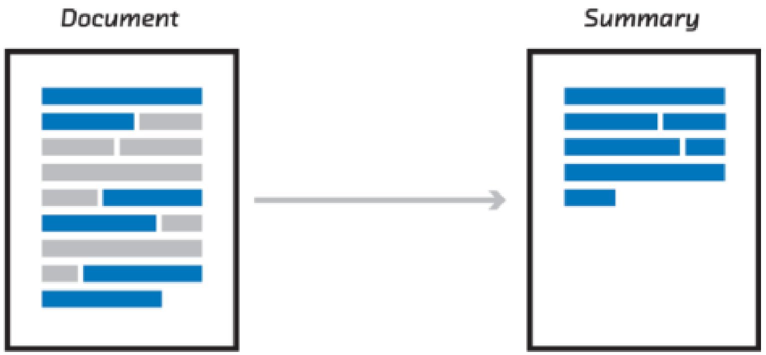
TextRank



Extractive (Generation)

Summary Selection

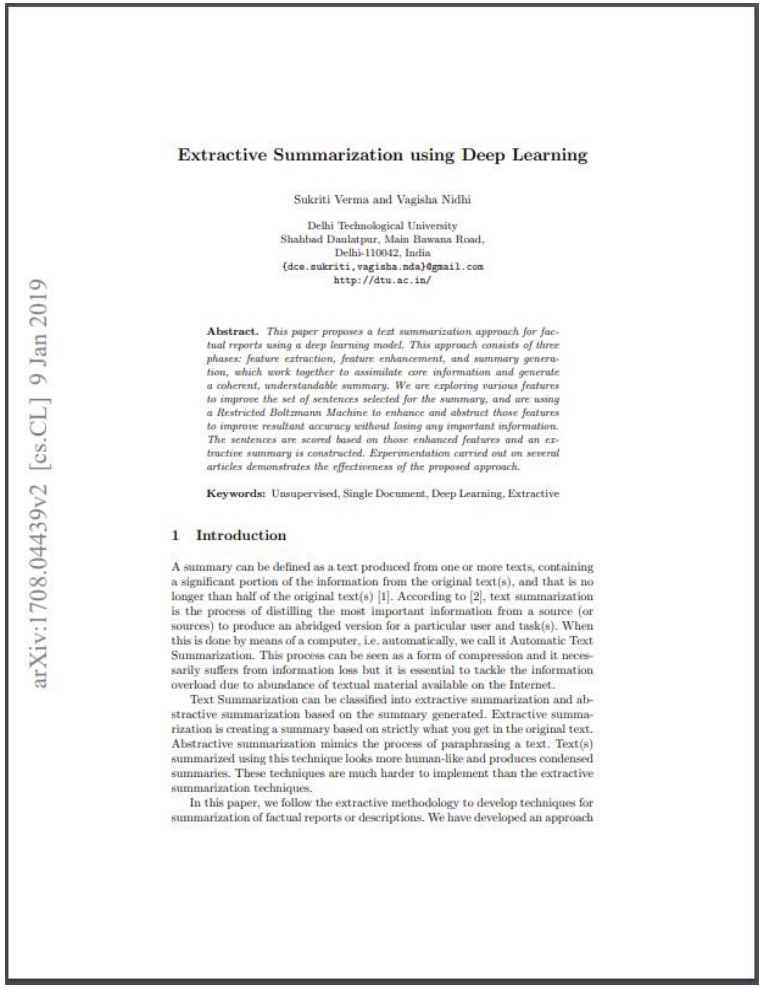
Summary Budget



Summary Constuction

Case Study - A: Overview

- Abstract
- Approach



arXiv:1708.04439v2 [cs.CL] 9 Jan 2019

Extractive Summarization using Deep Learning

Sukriti Verma and Vagisha Nidhi

Delhi Technological University
Shahbad Daulatpur, Main Bawana Road,
Delhi-110042, India
{dce.sukriti,vagisha.nda}@gmail.com
http://dtu.ac.in/

Abstract. *This paper proposes a text summarization approach for factual reports using a deep learning model. This approach consists of three phases: feature extraction, feature enhancement, and summary generation, which work together to assimilate core information and generate a coherent, understandable summary. We are exploring various features to improve the set of sentences selected for the summary, and are using a Restricted Boltzmann Machine to enhance and abstract those features to improve resultant accuracy without losing any important information. The sentences are scored based on those enhanced features and an extractive summary is constructed. Experimentation carried out on several articles demonstrates the effectiveness of the proposed approach.*

Keywords: Unsupervised, Single Document, Deep Learning, Extractive

1 Introduction

A summary can be defined as a text produced from one or more texts, containing a significant portion of the information from the original text(s), and that is no longer than half of the original text(s) [1]. According to [2], 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 and task(s). When this is done by means of a computer, i.e. automatically, we call it Automatic Text Summarization. This process can be seen as a form of compression and it necessarily suffers from information loss but it is essential to tackle the information overload due to abundance of textual material available on the Internet.

Text Summarization can be classified into extractive summarization and abstractive summarization based on the summary generated. Extractive summarization is creating a summary based on strictly what you get in the original text. Abstractive summarization mimics the process of paraphrasing a text. Text(s) summarized using this technique looks more human-like and produces condensed summaries. These techniques are much harder to implement than the extractive summarization techniques.

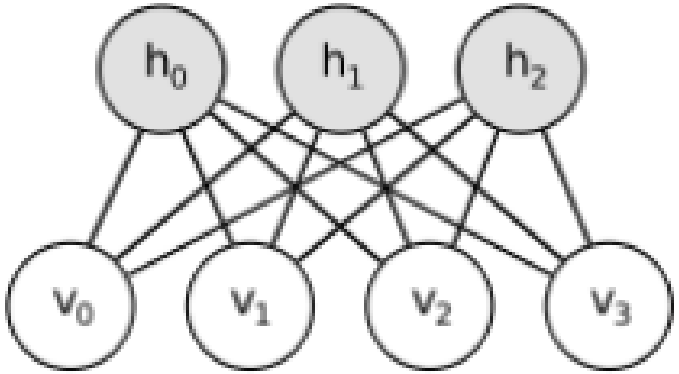
In this paper, we follow the extractive methodology to develop techniques for summarization of factual reports or descriptions. We have developed an approach

Case Study - A: Models - RBM

- Boltzmann machines are stochastic and generative neural networks capable of learning internal representations and can represent and (given sufficient time) solve difficult combinatoric problems.

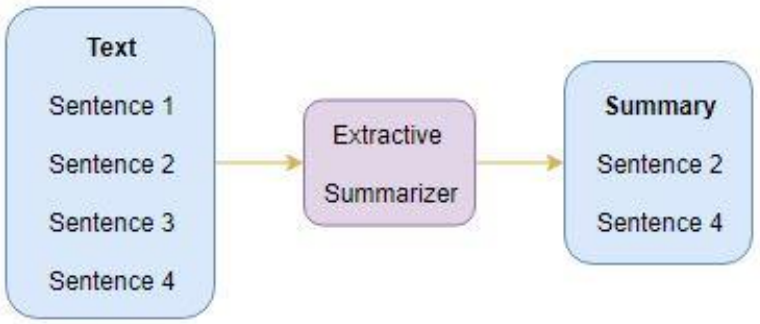
- Boltzmann machines are non-deterministic (or stochastic) generative Deep Learning models with only two types of nodes — `hidden` and `visible` nodes. There are no output nodes! This may seem strange, but this is what gives them this non-deterministic feature.

- unlike the other traditional networks (A/C/R) which don’t have any connections between the input nodes, a Boltzmann Machine has connections among the input nodes.



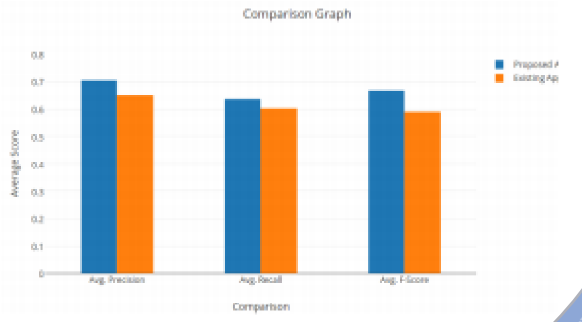
Case Study - A: Architecture

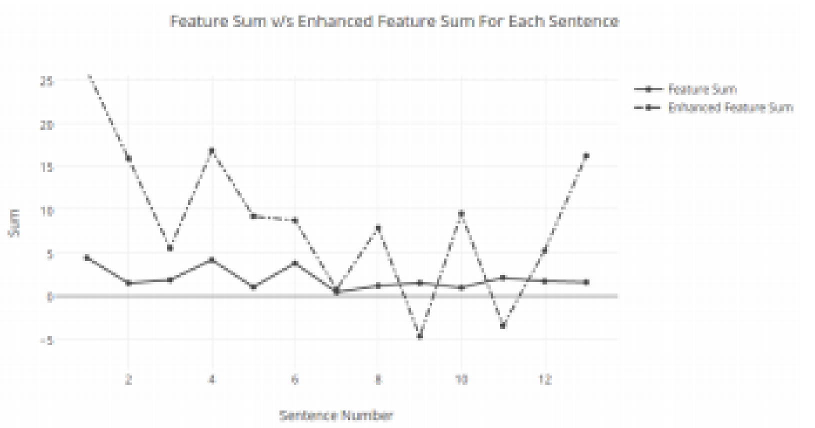
- Pre-processing
- Feature Extraction
- Feature Enhancement
- Summary Generation



Case Study - A: Experiments

- Benchmarks Used
- Results





Case Study - A: Conclusion

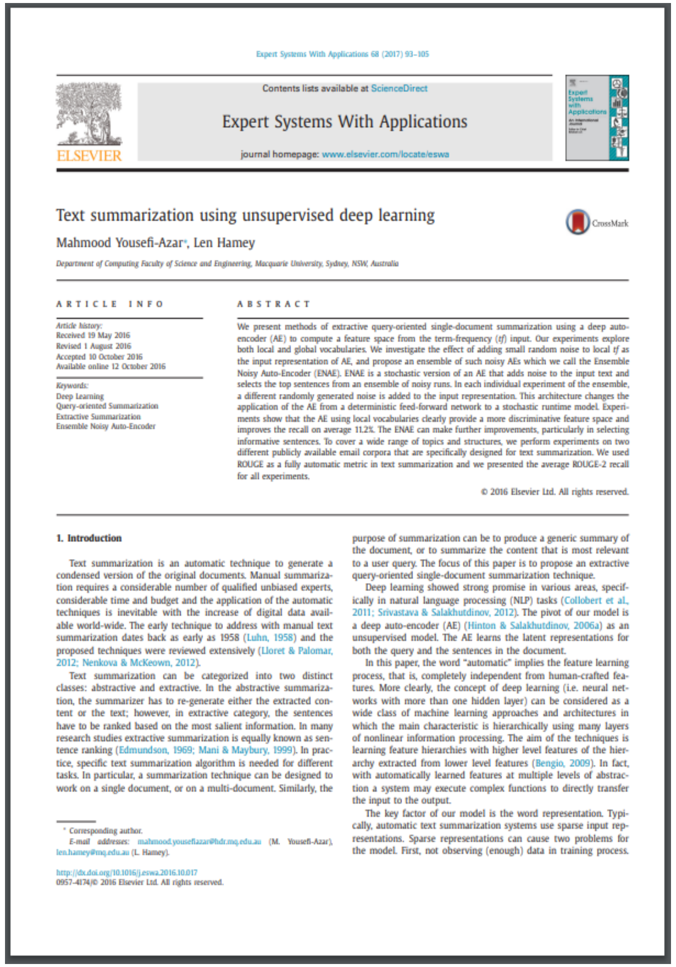
- Conclusion
- Our Findings

6 Conclusion

We have developed an algorithm to summarize single-document factual reports. The algorithm runs separately for each input document, instead of learning rules from a corpus, as each document is unique in itself. This is an advantage that our approach provides. We extract 9 features from the given document and enhance them to score each sentence. Recent approaches have been using 2 RBMs stacked on top of each other for feature enhancement. Our approach uses only

Case Study - B: Overview

- Yousefi et al use an Auto Encoder (AE) a type of unsupervised deep learning neural network to refine the features in the term frequencies of a document for summarization.
- First a chain of Restricted Boltzmann Machines are used to refine the weights for the text features.
- The weights are then used in the Auto encoder to generate a summary


Expert Systems With Applications 68 (2017) 93–105

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

ELSEVIER

Text summarization using unsupervised deep learning

Mahmood Yousefi-Azar*, Len Hamey

Department of Computing Faculty of Science and Engineering, Macquarie University, Sydney, NSW, Australia

ARTICLE INFO

Article history:
Received 19 May 2016
Revised 1 August 2016
Accepted 10 October 2016
Available online 12 October 2016

Keywords:
Deep Learning
Query-oriented Summarization
Extractive Summarization
Ensemble Noisy Auto-Encoder

ABSTRACT

We present methods of extractive query-oriented single-document summarization using a deep auto-encoder (AE) to compute a feature space from the term-frequency (*tf*) input. Our experiments explore both local and global vocabularies. We investigate the effect of adding small random noise to local *tf* as the input representation of AE, and propose an ensemble of such noisy AEs which we call the Ensemble Noisy Auto-Encoder (ENAE). ENAE is a stochastic version of an AE that adds noise to the input text and selects the top sentences from an ensemble of noisy runs. In each individual experiment of the ensemble, a different randomly generated noise is added to the input representation. This architecture changes the application of the AE from a deterministic feed-forward network to a stochastic runtime model. Experiments show that the AE using local vocabularies clearly provide a more discriminative feature space and improves the recall on average 11.2%. The ENAE can make further improvements, particularly in selecting informative sentences. To cover a wide range of topics and structures, we perform experiments on two different publicly available email corpora that are specifically designed for text summarization. We used ROUGE as a fully automatic metric in text summarization and we presented the average ROUGE-2 recall for all experiments.

1. Introduction

Text summarization is an automatic technique to generate a condensed version of the original documents. Manual summarization requires a considerable number of qualified unbiased experts, considerable time and budget and the application of the automatic techniques is inevitable with the increase of digital data available world-wide. The early technique to address with manual text summarization dates back as early as 1958 (Luhn, 1958) and the proposed techniques were reviewed extensively (Lloret & Palomar, 2012; Nenkova & McKeown, 2012).

Text summarization can be categorized into two distinct classes: abstractive and extractive. In the abstractive summarization, the summarizer has to re-generate either the extracted content or the text; however, in extractive category, the sentences have to be ranked based on the most salient information. In many research studies extractive summarization is equally known as sentence ranking (Edmundson, 1969; Mani & Maybury, 1999). In practice, specific text summarization algorithm is needed for different tasks. In particular, a summarization technique can be designed to work on a single document, or on a multi-document. Similarly, the purpose of summarization can be to produce a generic summary of the document, or to summarize the content that is most relevant to a user query. The focus of this paper is to propose an extractive query-oriented single-document summarization technique.

Deep learning showed strong promise in various areas, specifically in natural language processing (NLP) tasks (Collobert et al., 2011; Srivastava & Salakhutdinov, 2012). The pivot of our model is a deep auto-encoder (AE) (Hinton & Salakhutdinov, 2006a) as an unsupervised model. The AE learns the latent representations for both the query and the sentences in the document.

In this paper, the word "automatic" implies the feature learning process, that is, completely independent from human-crafted features. More clearly, the concept of deep learning (i.e. neural networks with more than one hidden layer) can be considered as a wide class of machine learning approaches and architectures in which the main characteristic is hierarchically using many layers of nonlinear information processing. The aim of the techniques is learning feature hierarchies with higher level features of the hierarchy extracted from lower level features (Bengio, 2009). In fact, with automatically learned features at multiple levels of abstraction a system may execute complex functions to directly transfer the input to the output.

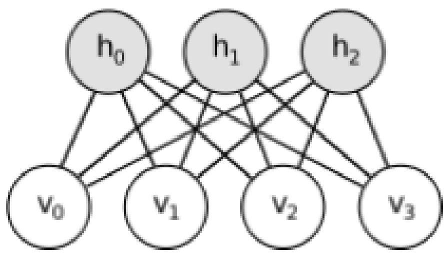
The key factor of our model is the word representation. Typically, automatic text summarization systems use sparse input representations. Sparse representations can cause two problems for the model. First, not observing (enough) data in training process.

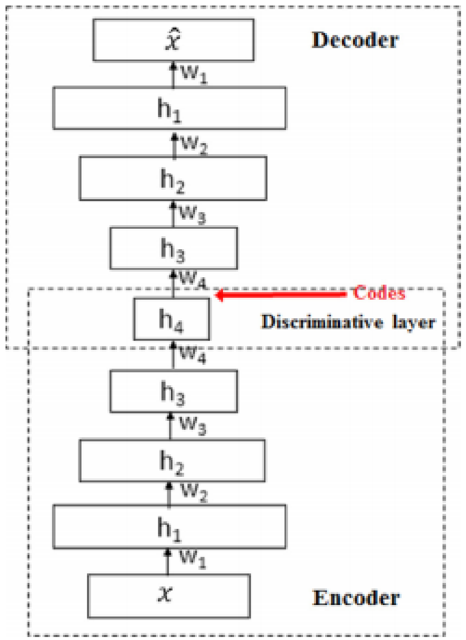
* Corresponding author.
E-mail addresses: mahmood.yousefiazar@hdr.mq.edu.au (M. Yousefi-Azar), len.hamey@mq.edu.au (L. Hamey).

http://dx.doi.org/10.1016/j.eswa.2016.10.017

Case Study - B:
Models - Autoencoder

- The AE neural network is feed forward network, the main feature of this network is the bottleneck in it hidden layer, its input and output layers have the same number of nodes and the network replicates its input as its output.
- The case study uses this reconstructive ability by adding random noise to their inputs, by doing this the most salient features would be elicited by the encoder.
- The study used several encoders with different random noise masks this created several feature maps.
- Using multiple maps the most prominent features across all maps were found and these features were used to generate the summary.





Evaluation Techniques

- ROUGE, or Recall-Oriented Understudy for Gisting Evaluation are a set of metrics that are used to evaluate the results of NLP applications such as Machine translation, Auto Summarization etc.
- There are five metrics that are used to evaluate NLP results:
    1. Rouge-1: Unigram Overlap
    2. Rouge-2: Bigram Overlap
    3. Rouge-L: Longest Common Sequence (LCS)
    4. Rouge-W: Weighted LCS
    5. Rouge-S: Skip Bigram
    6. Rouge-SU: Co-occurrence of Rouge-S and Rouge-1

References

- Quantifying documents by calculating tf-idf in R
- Introduction to Latent Matrix Factorization recommender Systems
- Comprehensive guide to text summarization using deep learning
- Feature engineering framework techniques
- https://medium.com/@social_20188/text-summarization-cfdbbd6fb800
- https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/
- O. Foong, S. Yong and F. Jaid, "Text Summarization Using Latent Semantic Analysis Model in Mobile Android Platform," *2015 9th Asia Modelling Symposium (AMS)*, Kuala Lumpur, 2015, pp. 35-39.
- Gupta, Som & Gupta, S. K, 2019. Abstractive summarization: An overview of the state of the art. Expert Systems With Applications, 121, pp.49–65.

Different Papers Rouge score

RANK	METHOD	ROUGE-1	ROUGE-2	ROUGE-L	PAPER TITLE	YEAR	PAPER	CODE
1	Transformer+LRPE+PE+Re-ranking+Ensemble	32.85	11.78	28.52	Positional Encoding to Control Output Sequence Length	2019		
2	Transformer+LRPE+PE+ALONE+Re-ranking	32.57	11.63	28.24	All Word Embeddings from One Embedding	2020		
3	EndDec+WFE	32.28	10.54	27.8	Cutting-off Redundant Repeating Generations for Neural Abstractive Summarization	2017		
4	DRGD	31.79	10.75	27.48	Deep Recurrent Generative Decoder for Abstractive Text Summarization	2017		
5	Seq2seq + selective + MTL + ERAM	29.33	10.24	25.24	Ensure the Correctness of the Summary: Incorporate Entailment Knowledge into Abstractive Sentence Summarization	2018		
6	SEASS	29.21	9.56	25.51	Selective Encoding for Abstractive Sentence Summarization	2017		
7	RAS-Elman	28.97	8.26	24.06	Abstractive Sentence Summarization with Attentive Recurrent Neural Networks	2016		
8	words-lvt5k-1sent	28.61	9.42	25.24	Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond	2016		
9	ABS+ ()	28.18	8.49	23.81				
10	ABS ()	26.55	7.06	22.05				

Attention Mechanism

- Due to the chained nature of the Encoder Decoder Architecture, the initialization variable is transformed through the chain.
- The chain discards the intermediate variable between nodes.
- The Attention Mechanism preserves the intermediate variables and combines it with the final output.