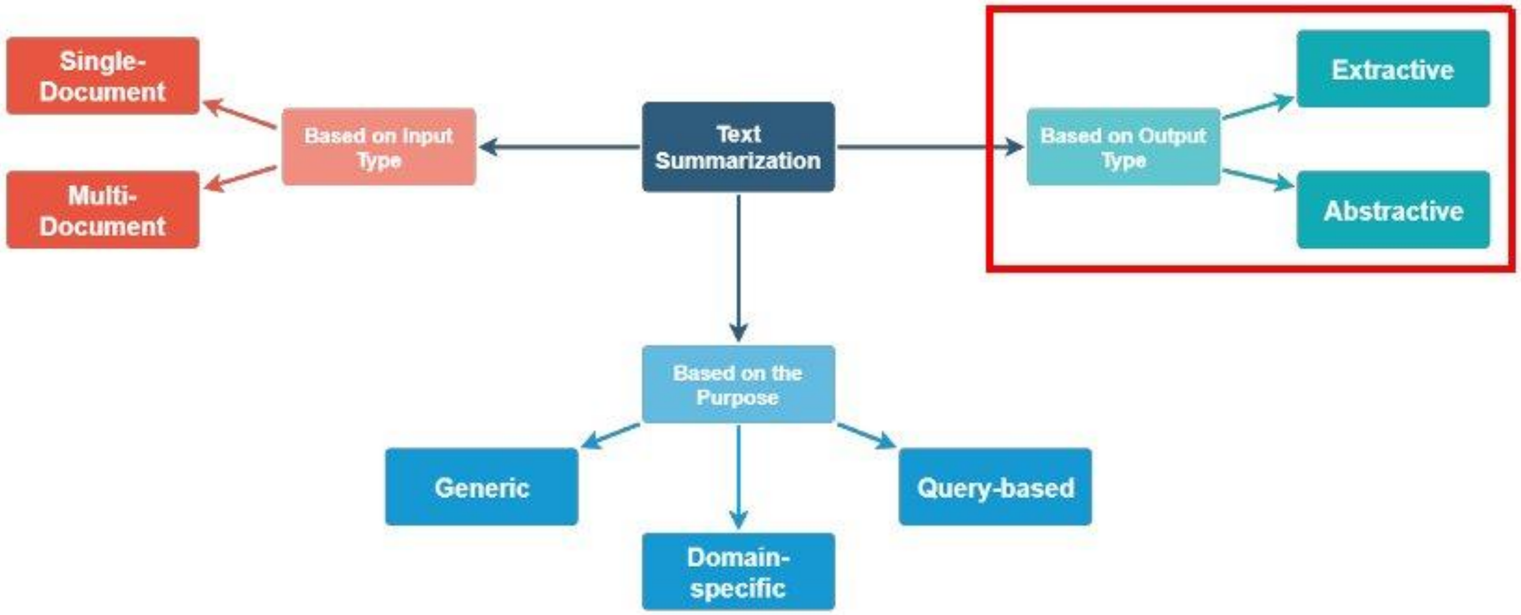
Document Summarization using Neural Networks

Manu Hegde, VIII ‘A’, CSE
1DA15CS067

Paradigms in Document/Text summarization



Major types of Document Summarization

- **Extractive document summarization** is generating summaries by selecting words, phrases or sentences which when combined can effectively represent the gist of the document.

- **Abstractive document summarization** is generating summaries that may contain words other than those present in the original document.

Abstractive summarization requires that the summarizer be able to represent a given set of sentences in an intermediate representation using which a summary can be derived. Hence until 2014, efforts for summarization were mostly Extractive.

The first major success in abstractive summarization was enabled by the creation of high dimensional word embeddings*[1]* such that words were represented as vectors and whose dot product showed their similarity.

This led to development of recurrent neural network based encoder decoder (Sequence to Sequence) models *[2]*.

[1] - Word2Vec https://arxiv.org/abs/1301.3781,
[2] - Sequence to Sequence Learning with Neural Networks https://arxiv.org/abs/1409.3215

Approaches to Extractive Document Summarization

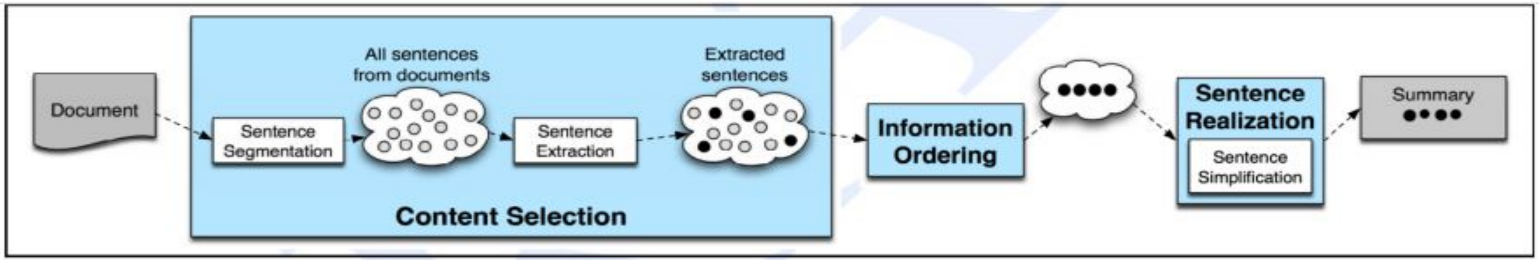
Extractive document summarization has been mostly done by complex systems carefully engineered with manual feature selection and pre processing based on language specific patterns.
Mostly they belonged to following categories:
Sentence scoring functions:

- Based on presence of topic keywords
- Features such as where the sentence appears in the document

Graph-based algorithms

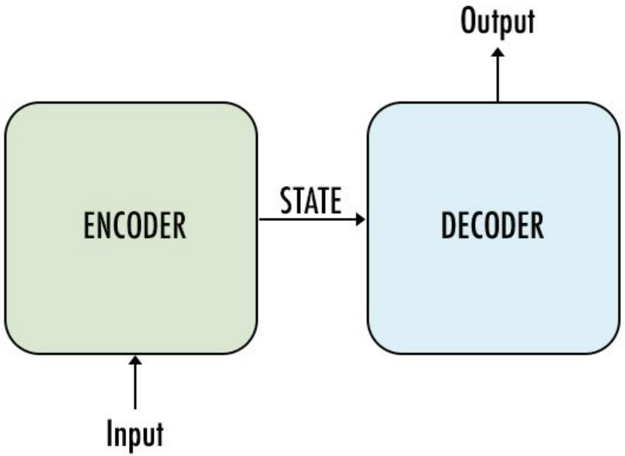
- view the document as a set of sentences (nodes), with edges between each sentence pair
- Edge weight is proportional to sentence similarity
- Use graph algorithms to identify sentences which are central in the graph

A general architecture of a pre neural era extractive summarizer



Approaches to Abstractive Document Summarization

Abstractive document summarization has been achieved using various types of models.
However, all major models are based on Encoder-Decoder architecture. The Encoder-Decoder architecture is a common base technology for various other tasks like Translation, Text classification, etc.



Encoder: Takes text as input, in the form of word embeddings (like Word2Vec), optionally adds position encoding for words and outputs a latent state/representation for given input.

Latent State/ Representation is an intermediate, fixed length value (usually), which is capable of meaningfully representing the contents of the input.

Decoder: Takes the latent representation as input and generates the final output i.e the summary.

Recurrent Bi-LSTM/GRU based Seq2Seq, Seq2Seq with Attention Mechanism and Transformers with Self Attention are three major and widely used architectures for abstractive summarization.

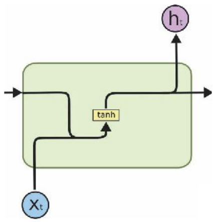
Recurrent Seq2Seq Model

Recurrent Neural Networks (RNN) are a subcategory of neural networks where output of a neuron is fed back to it along with its input. This feedback is considered as previous state of the recurrent neuron. Every state of a recurrent neuron is considered as a time step.
As the length of input increases, information from much older time steps eventually vanish. Two major variants of RNN are hence used as alternatives.

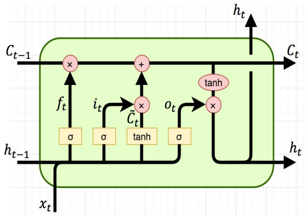
- **LSTM (Long Short-Term Memory Cells):** Introduced in 1997, these cells have the ability to store and retrieve information from many previous time steps. Hypothetically LSTMs are more powerful most other variants of RNN.

- **GRU (Gated Recurrent Units):** Introduced in 2014, These cells are much faster compared to LSTM and are proven to outperform RNNs and LSTMs in practical scenarios.

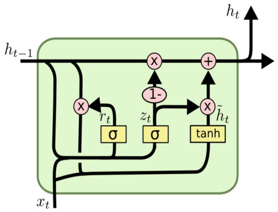
RNN



LSTM



GRU



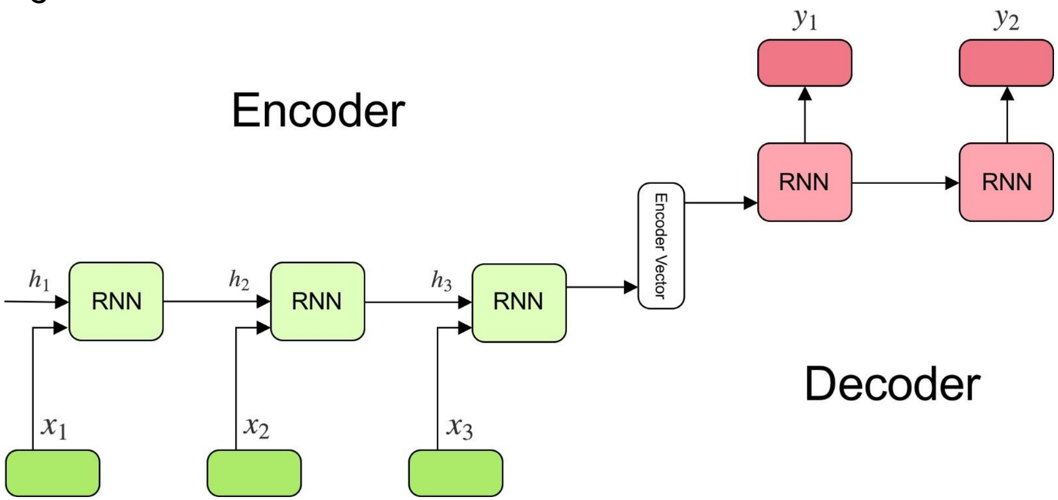
LONG SHORT-TERM MEMORY - https://www.bioinf.jku.at/publications/older/2604.pdf
Gated Recurrent Units - https://arxiv.org/abs/1412.3555

Recurrent Seq2Seq model

- **Recurrent Seq2Seq (Sequence2Sequence)** are made of either of RNNs/GRUs/LSTMs.
- LSTMs and GRUs are used more commonly and every recurrent unit can have multiple hidden states
- The number of recurrent units in the Encoder is linearly dependant on the length on the input, i.e the number of words in the input.
- This model is also used to generate contextual embeddings, i.e embeddings for words based on its position in the sentence and its surrounding words.

- **Advantages:**
- Computational efficient compared to other models and can model vocabulary of upto few million words.

- **Disadvantages:**
- Cannot be used for summarizing very long documents.
- Outputs UNK (unknown) symbol whenever a word outside its trained vocabulary is seen.

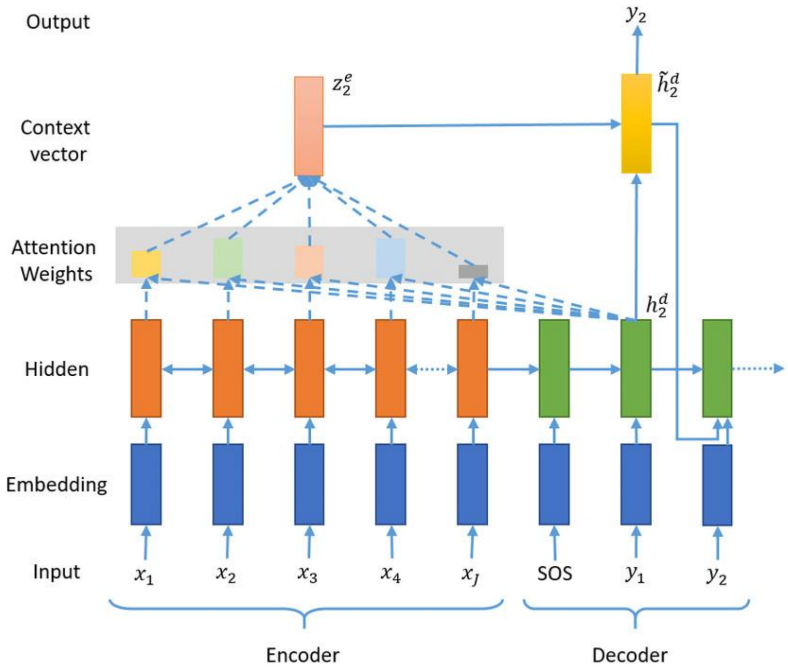


Skip Thought Vectors - https://arxiv.org/abs/1506.06726
Sequence to Sequence Learning with Neural Networks - https://arxiv.org/abs/1409.3215

Recurrent Seq2Seq with Attention

- **Recurrent Seq2Seq (Sequence2Sequence) with Attention** consists of an additional attention mechanism over the existing Seq2Seq architecture.
- Attention A is calculated as

$$A(q, K, V) = \sum_i \frac{e^{q \cdot k_i}}{\sum_j e^{q \cdot k_j}} v_i$$



- **Advantages:**
- Attention can be used to concentrate on important parts of a sentence, in a self supervised manner.
- It can be easily accelerated using GPU since its not sequential.
- Enables in considering larger context of the document.
- **Disadvantages:**
- An Extra layer of computation that grows quadratically with input size and dimension of the latent state.

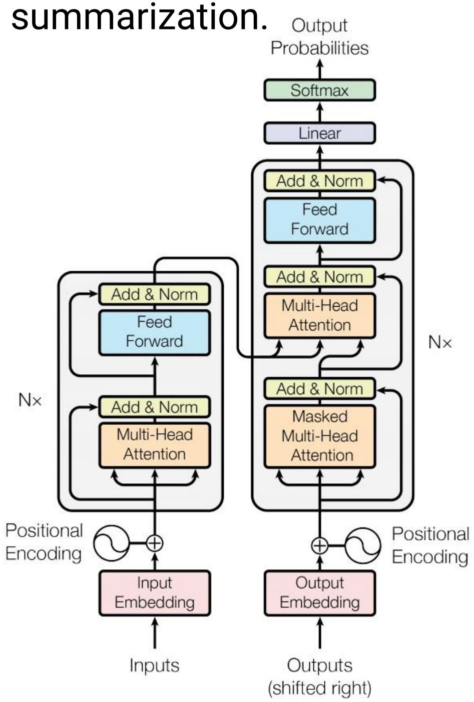
A Neural Attention Model for Abstractive Sentence Summarization - https://arxiv.org/pdf/1509.00685.pdf
Attention Is All You Need - https://arxiv.org/abs/1706.03762

Transformer Models

- **Transformer models with Attention** is a novel architecture that uses pure attention, without any recurrent neurons for various tasks like translation, language modelling and summarization.

- **Advantages:**
- Multi head attention is used to learn various smaller and larger contexts in a document.
- Very good performance even on very large input.
- Enhanced learning due to positional encoding and context vectors.
- Can work with character level language modelling, i.e it can understand words even with repeated letters, i.e it knows 'yesss' and 'yes' are the same.
- It can understand that same word in different position has different meaning.

- **Disadvantages:**
- Very computationally intensive to train. One of the most popular model i.e BERT by Google has 117 Million parameters.



Attention Is All You Need - https://arxiv.org/abs/1706.03762
Unsupervised Multi task Learners - https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf
BERT: Deep Bidirectional Transformers https://arxiv.org/pdf/1810.04805.pdf

Applications

- Summarizers can be used for quick understanding of lengthy academic papers.
- It can also be used for obtaining simplified explanations of complex scientific literature.
- For generation of news feed of news articles.
- For summarizing tenders and legal documents including end user licence agreements which are known to be very long.
- For quick comparison of documents.