Summarization Task

Outline

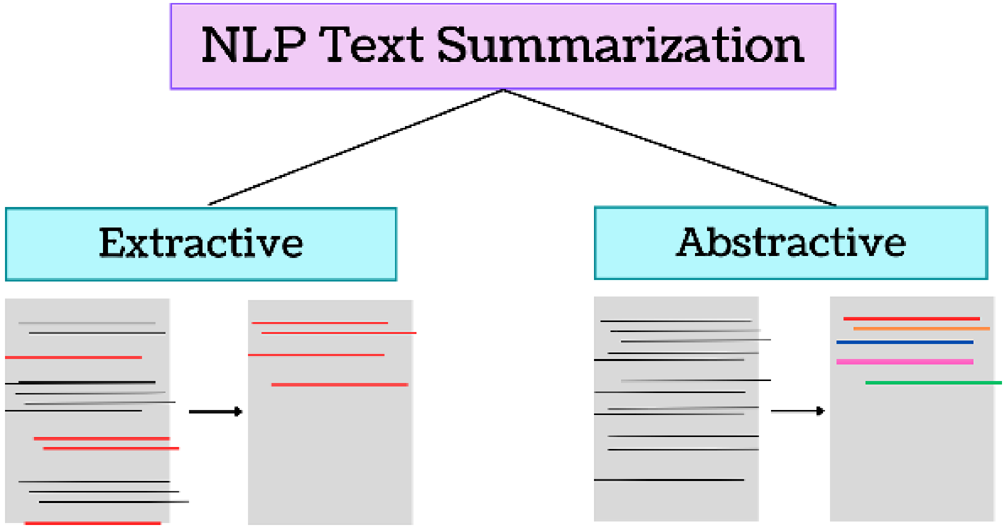
- Motivation
- Neural Networks
- Transformers
    - Encoder
    - Decoder
- Pretrained Models
- Datasets Available
- Evaluation Metrics

Motivation

- Summarization is a crucial skill in our information-rich world where the volume of available information can be overwhelming. It helps us efficiently extract the most important and relevant information from large amounts of text.

Motivation

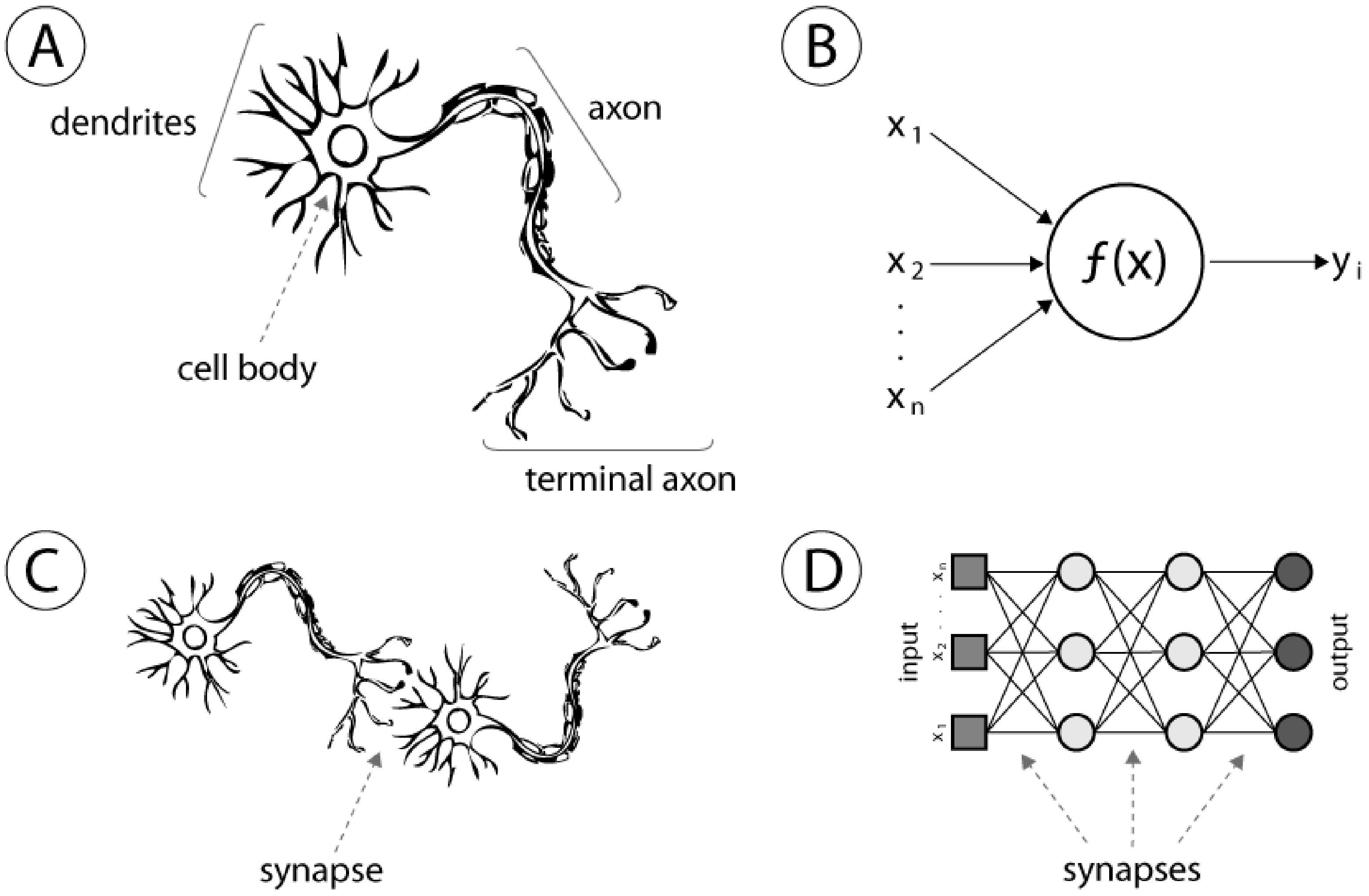
- Extractive summarization involves selecting important sentences or phrases from the original text and combining them to create a summary.
- Abstractive summarization involves generating new sentences that capture the meaning of the original text.



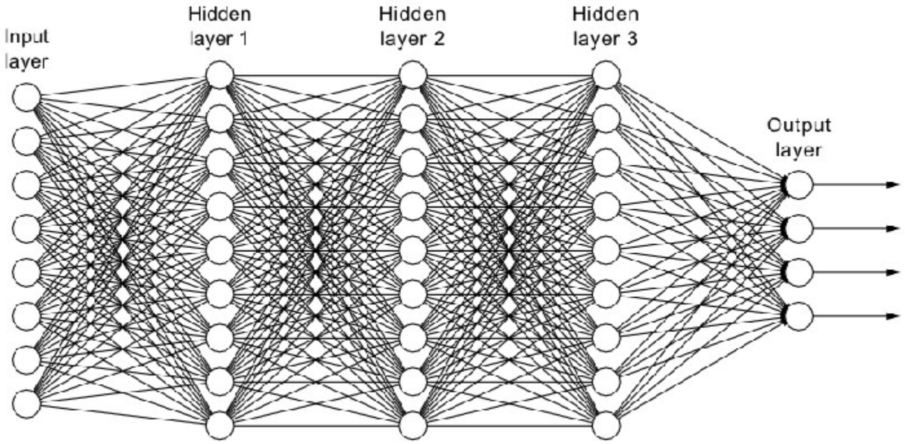
Neural Networks

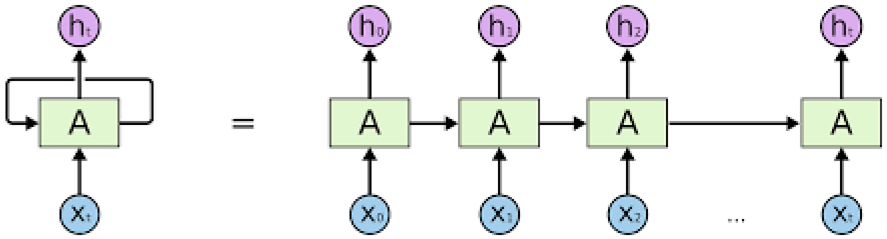
- Neural networks are a type of machine learning algorithm that mimics the structure and function of the human brain.
- They consist of interconnected nodes or neurons that process and transmit information.
- Neural networks are trained on a dataset to minimize the difference between predicted and true outputs, once trained, they can be used to make predictions on new data.
- Neural networks have shown impressive performance in a variety of applications and are a powerful tool for solving complex problems.

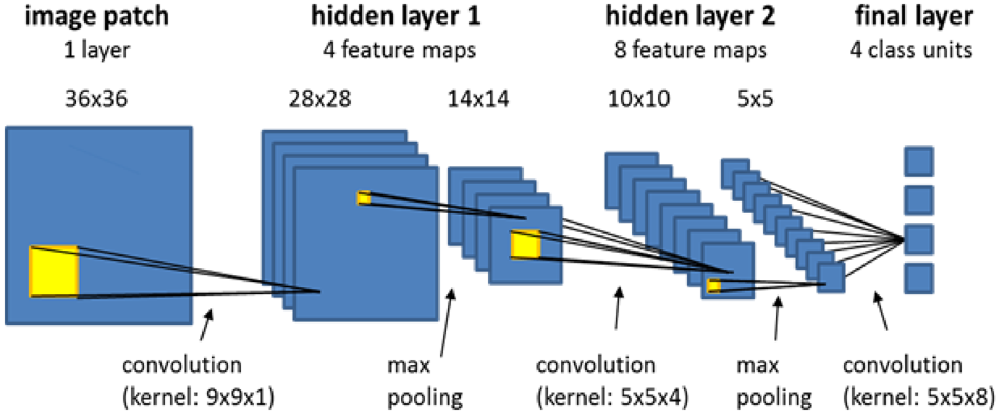
Neural Networks

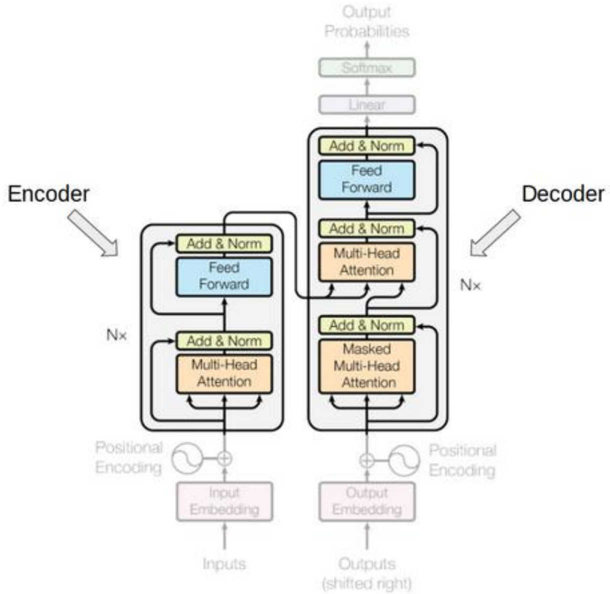


Neural Networks Architecture





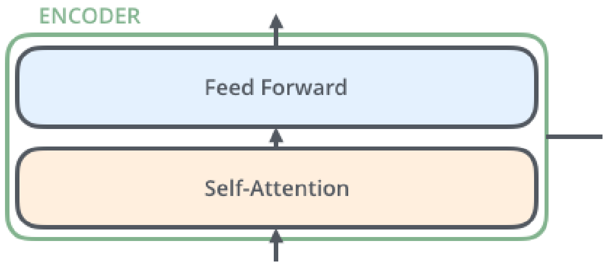




Transformers

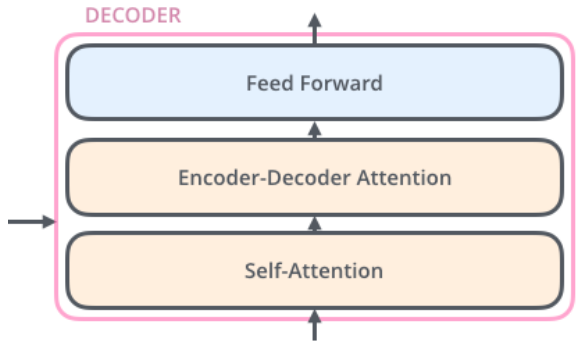
- Transformers have shown impressive performance in a wide range of natural language processing (NLP) tasks, including text summarization.
- They are particularly well-suited to tasks that require modeling long-range dependencies and can handle input sequences of variable length.
- Transformers have become a fundamental building block in many state-of-the-art NLP models and are an active area of research in the field.

Transformer- Encoder



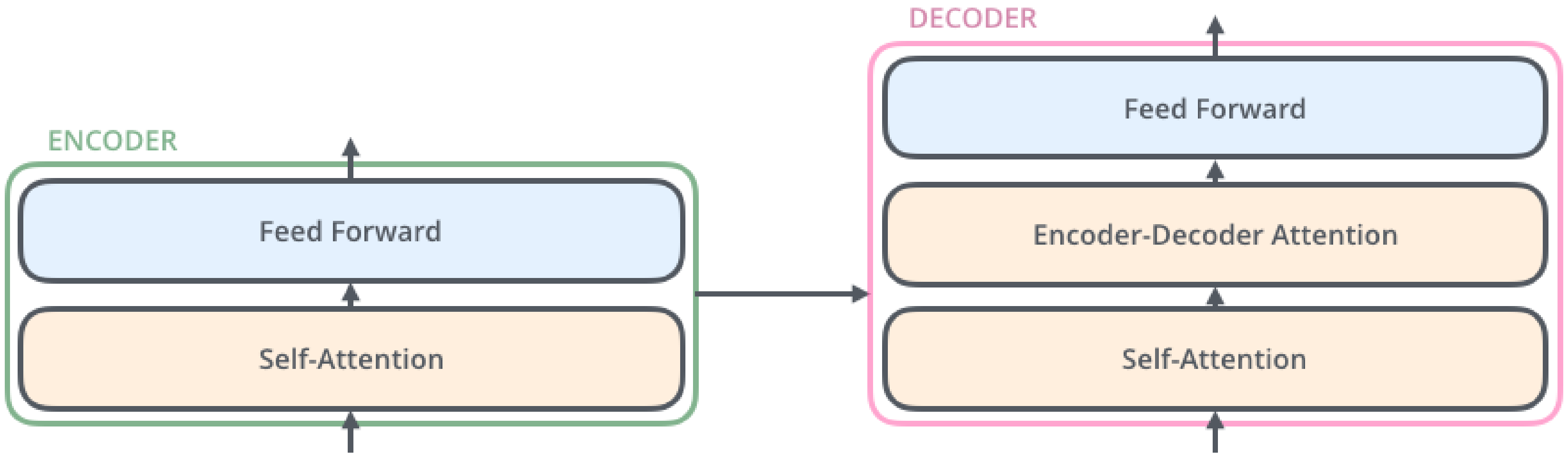
- It consists of two sub-layers: the self-attention layer and the feedforward layer.
  - The self-attention layer allows the model to selectively attend to different parts of the input sequence and capture long-range dependencies.
  - The feedforward layer applies a non-linear transformation to the output of the self-attention layer.
- The transformer encoder layer has been shown to be highly effective in a variety of natural language Understanding tasks.

Transformer-Decoder



- It consists of three sub-layers: the self-attention layer, the encoder-decoder attention layer, and the feedforward layer.
  - The self-attention layer is the same as the encoder.
  - The encoder-decoder attention layer allows the decoder to attend to the encoder outputs, enabling the model to generate an output sequence that is conditioned on the input sequence.
  - The feedforward layer applies a non-linear transformation to the output of the previous two layers.
- The transformer decoder layer is a key component in sequence-to-sequence models, which have been shown to be highly effective in tasks such as language translation and text summarization.

Transformers – Encoder-decoder



Model Pretraining

- Pretraining is a machine learning technique where a model is trained on a large, labeled dataset before being fine-tuned on a specific task
- In NLP, pretraining has been used to develop language models such as BERT and GPT.
- Pretraining allows models to learn general language representations that can be applied to different tasks, leading to better performance with less labeled data.
- Pretrained models have become a crucial component in many state-of-the-art NLP systems.

Available Pretrained Models

- There a lot of pretrained models available but we need it to be pretrained on Arabic
    - Mbart
    - Mt5
    - Arat5
    - Deltalm

Models Fine-tunning

- Model fine-tuning is a machine learning technique that involves further training a pre-trained model on a smaller labeled dataset for a specific task.
- Fine-tuning saves time and resources compared to training a model from scratch.
- Fine-tuning has been successfully applied in many NLP tasks.
- Fine-tuning has become a standard approach in many NLP applications.

Datasets

- WikiLingua
  - The dataset includes ~770k article and summary pairs in 18 languages from WikiHow.
  - Dataset contains around 29,229 article-summary pairs with a parallel article-summary pair in English.
- XL-SUM
  - The latest version contains a total of **1.35 million** article-summary pairs in 44 languages.
  - Dataset contains around 46,897 article-summary pairs.

Tools

- PyTorch
- Transformers
- Tokenizers
- Sentincepiece
- Fairseq

Evaluation Metrics

ROUGE : Recall-Oriented Understudy for Gisting Evaluation

- ROUGE can be divided into
  - ROUGE-N : Overlap of n-grams between the system and reference summaries
    - ROUGE-1
    - ROUGE-2
  - ROUGE-L : Ratio of the Longest Common Subsequence between the system and reference summary.

ROUGE-1

ROUGE-1 Example

- Reference Text : “the cat was under the bed”, Length = 6
- System Summary : “the cat was found under the bed”, Length = 7

We start by calculating the precision and recall of 1 grams (single words) :

- **precision** : Number of overlapping words / Length of system summary = 6/7
- **recall** : Number of overlapping words / Length of reference summary = 6/6
- **ROUGE-1 Score** (F1-Score) : $\frac{2 * \boldsymbol{precision} * \boldsymbol{recall}}{\boldsymbol{precision} + \boldsymbol{recall}} = \frac{2 * \left(\frac{6}{7}\right) * \left(\frac{6}{6}\right)}{\left(\frac{6}{7}\right) + \left(\frac{6}{6}\right)} = 0.55$

ROUGE-2

ROUGE-2 Example

We start by calculating the precision and recall of 2 grams :

- Reference Text : “the cat was under the bed”
  - Reference 2-grams : (the cat, cat was, was under, under the, the bed) , Length = 5
- System Summary : “the cat was found under the bed”
  - System 2-grams : (the cat, cat was, was found, found under, under the, the bed) , Length = 6

- Overlapping 2-grams = 4
- **precision** : Number of overlapping 2-grams / Length of system 2-grams = 4/6
- **recall** : Number of overlapping 2-grams / Length of reference 2-grams = 4/5
- **ROUGE-1 Score** (F1-Score) : $\frac{2*\boldsymbol{precision}*\boldsymbol{recall}}{\boldsymbol{precision}+\boldsymbol{recall}} = \frac{2*\left(\frac{4}{6}\right)*\left(\frac{4}{5}\right)}{\left(\frac{4}{6}\right)+\left(\frac{4}{5}\right)} = 0.729$

ROUGE-L

ROUGE-L Example

We start by calculating the longest common subsequence

- Reference Text : "the cat was under the bed" , Length = 6
- System Summary : "the cat was found under the bed" , Length = 7

- Length of longest common subsequence = 6
- **precision** : Length of longest common subsequence / Length of system 1-grams = 6/7
- **recall** : Number of overlapping 2-grams / Length of reference 1-grams = 6/6
- **ROUGE-1 Score** (F1-Score) : $\frac{2 * \boldsymbol{precision} * \boldsymbol{recall}}{\boldsymbol{precision} + \boldsymbol{recall}} = \frac{2 * \left(\frac{6}{7}\right) * \left(\frac{6}{6}\right)}{\left(\frac{6}{7}\right) + \left(\frac{6}{6}\right)} = 0.923$

Thank you !

Omar.khaled@aic.gov.eg

Applied Innovation Center (AIC)

Ministry of Communications and Information Technology