Summarization

Brief Introduction

Martin Dočekal

Summarization

- The goal is to write summary y for input text x that is shorter than x and contains all essential information in x.
- The x could be a **single document** or **multiple documents**. In case of multiple documents we talk about **multi-document summarization**.
- Special types of summarization:
  - Keyphrase Extraction
  - Query-based summarization also known as aspect-based (more further)

Query-based summarization

- Generates summary that is relevant to given query
- It is very similar to long-form question answering
  - But the summary shouldn't be just answer to the question. It should also cover additional background information that are related to query.
  - They may use the same dataset (HotpotQA [Xie et al., 2020])

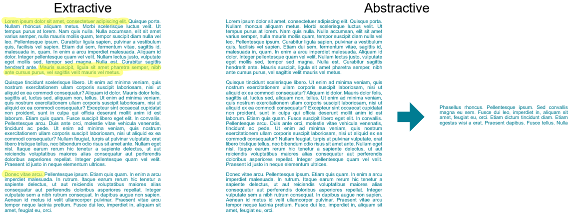
Summarization

Classification according to summary content type

- Indicative - only presents the main idea of a text
  - This type of summary may be used to encourage the reader to read the original document(s).
  - Abstracts
- Informative - contains concise information from original text and can be used instead of it
  - Is typically longer than the indicative variant.
  - Encyclopedias

Summarization - strategies

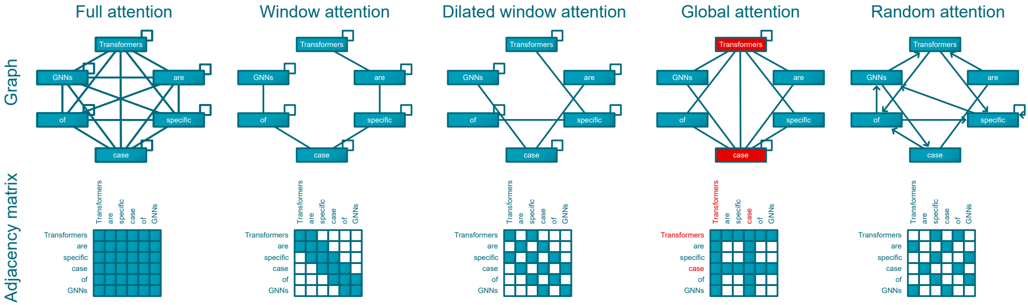
- **Extractive summarization**
  - Selects parts (typically sentences) of original text
  - As the output consists of original text parts it is not so sensible to generation of nonsense
- **Abstractive summarization**
  - Generates the summary with usage of NLG
  - Can generate more human-like output



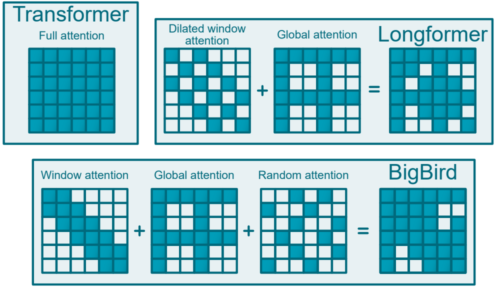
Summarization from long documents

- Full self-attention is too demanding for long sequences
- We can use models that are using sparse attention

Sparse attentions



Sparse attentions - Models



- Pretrained models for summarization:
  - **PRIMERA** [Xiao et al., 2022] - Longformer [Beltagy et al., 2020] based model pretrained for multi-document summarization
  - **Bigbird-PEGASUS** [Zaheer et al., 2020] - BigBird based model pretrained for single-document summarization
    - Original PEGASUS [Zhang et al., 2020] is based on BART [Lewis et al., 2020]

Extractive summarization on abstractive dataset

- A lot of summarization datasets contain only abstractive ground truth outputs
- An extractive model expects labeled parts, which should be extracted, during training
- The labels are typically obtained by a heuristic, such as:
  - Greedily select sentences that maximize ROUGE score with respect to gold summaries

Evaluation

Evaluation - ROUGE

- ROUGE - Recall-Oriented Understudy for Gisting Evaluation [Lin, 2004]
- Is typically used for summarization
- ROUGE-N is an n-gram recall between a candidate and possible set of references. For single reference S it is:

$$\text{ROUGE-N} = \frac{\sum_{gram_n \in S} \text{Count}_{match}(\text{gram}_n)}{\sum_{gram_n \in S} \text{Count}(\text{gram}_n)} \tag{1}$$

Evaluation - ROUGE-L

- Is longest common subsequence (LCS) based F-measure
  - LCS is not necessarily contiguous. For example a b c is subsequence of a 1 b 2 c.
- For reference X of length m and candidate Y of length n the ROUGE-L is calculated as follows:

$$R_{\text{lcs}} = \frac{\text{LCS}(X, Y)}{m} \tag{2}$$

$$P_{\text{lcs}} = \frac{\text{LCS}(X, Y)}{n} \tag{3}$$

$$\text{ROUGE-L} = F_{\text{lcs}} = \frac{(1 + \beta^2) R_{\text{lcs}} P_{\text{lcs}}}{R_{\text{lcs}} + \beta^2 P_{\text{lcs}}} \tag{4}$$

- Where $\beta = P_{\text{lcs}} / R_{\text{lcs}}$ (1 is also used) and the $\text{LCS}(X, Y)$ is length of a longest common subsequence of X and Y.

Evaluation - ROUGE-L - example

- Reference (X): There is a cat on the mat
- Candidate (Y): The cat is on the mat

Evaluation - ROUGE-L - example

- Reference (X): There is a **cat on the mat**
- Candidate (Y): The **cat** is **on the mat**
- $\text{LCS}(X, Y) = 4$
- $m = 7$
- $n = 6$

Evaluation - ROUGE-L - example

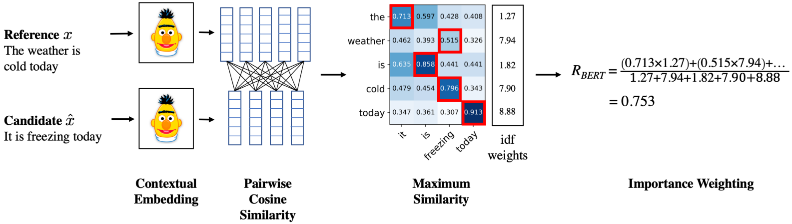
- Reference (X): There is a **cat on the mat**
- Candidate (Y): The **cat** is **on the mat**
- $\mathrm{LCS}(X, Y) = 4$
- $m = 7$
- $n = 6$
- $R_{\mathrm{lcs}} = \frac{\mathrm{LCS}(X,Y)}{m} = \frac{4}{7}$
- $P_{\mathrm{lcs}} = \frac{\mathrm{LCS}(X,Y)}{n} = \frac{4}{6}$
- $\beta = 1$

Evaluation - ROUGE-L - example

- Reference (X): There is a **cat on the mat**
- Candidate (Y): The **cat** is **on the mat**
- $\text{LCS}(X, Y) = 4$
- $m = 7$
- $n = 6$
- $R_{\text{lcs}} = \frac{\text{LCS}(X,Y)}{m} = \frac{4}{7}$
- $P_{\text{lcs}} = \frac{\text{LCS}(X,Y)}{n} = \frac{4}{6}$
- $\beta = 1$
- $\text{ROUGE-L} = F_{\text{lcs}} = \frac{(1+\beta^2)R_{\text{lcs}}P_{\text{lcs}}}{R_{\text{lcs}}+\beta^2 P_{\text{lcs}}} = \frac{2\frac{4}{7}\frac{4}{6}}{\frac{4}{7}+\frac{4}{6}} = 0.615$

Evaluation

- Evaluation of summarization (NLG in general) is difficult task and the automatic n-gram overlap based metrics are not perfect.
- They don't consider semantics and typically we don't have exhaustive set of semantically same gold **references**.
- There is an effort to capture semantics using **word embeddings**
  - E.g. BertScore [Zhang* et al., 2020]



References I

Beltagy, I., Peters, M. E., and Cohan, A. (2020).
Longformer: The long-document transformer.
arXiv preprint arXiv:2004.05150.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020).
BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.
In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Lin, C.-Y. (2004).
Rouge: A package for automatic evaluation of summaries.
In *Text summarization branches out*, pages 74–81.

References II

Xiao, W., Beltagy, I., Carenini, G., and Cohan, A. (2022).
PRIMERA: Pyramid-based masked sentence pre-training for multi-document summarization.
In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5245–5263, Dublin, Ireland. Association for Computational Linguistics.

Xie, Y., Zhou, T., Mao, Y., and Chen, W. (2020).
Conditional self-attention for query-based summarization.
arXiv preprint arXiv:2002.07338.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., et al. (2020).
Big bird: Transformers for longer sequences.
Advances in neural information processing systems, 33:17283–17297.

References III

Zhang, J., Zhao, Y., Saleh, M., and Liu, P. J. (2020).
Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.
In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Zhang*, T., Kishore*, V., Wu*, F., Weinberger, K. Q., and Artzi, Y. (2020).
Bertscore: Evaluating text generation with bert.
In *International Conference on Learning Representations*.