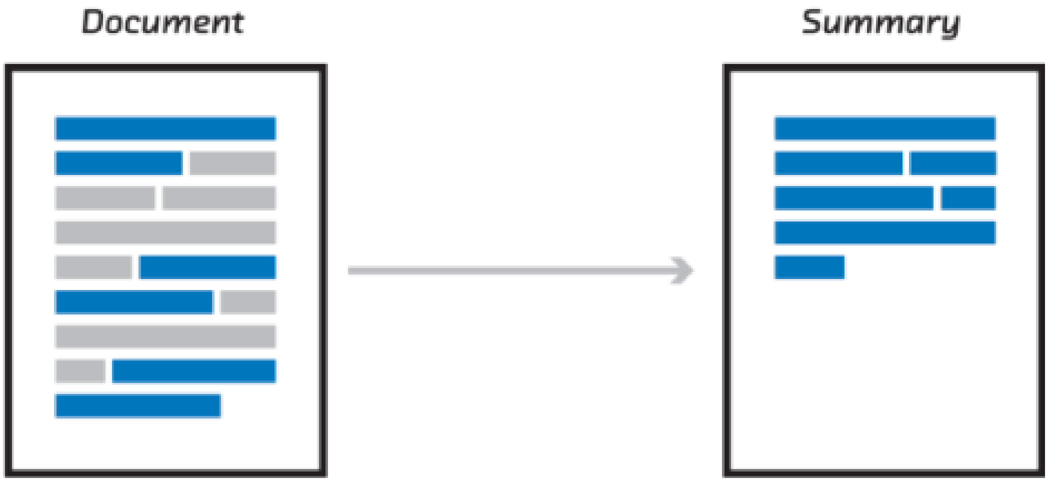
BERT FOR EXTRACTIVE SUMMARIZATION

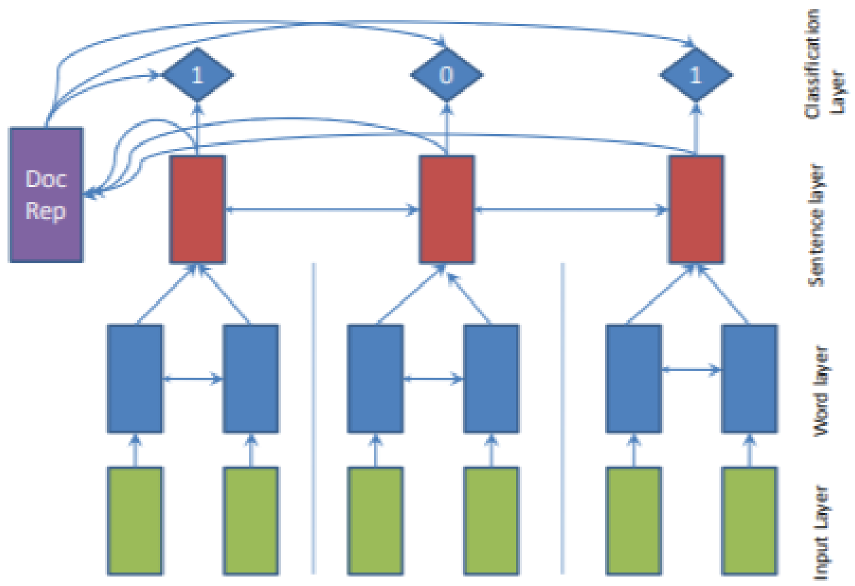
Urchade Zaratiana

EXTRACTIVE SUMMARIZATION



- In extractive summarization, the **most relevant sentences** in a document is selected as its summary.
- There is not directly a dataset for extractive summarization:
  - Some papers use an unsupervised approach to convert the **abstractive summaries** to **extractive labels**.
  - ➢ Idea: the selected sentences from the document should be the ones that maximize the Rouge score with respect to gold summaries *(Nallapati et al., 2017)*
- Commonly used dataset: CNN/DailyMail dataset (Text & Abstractive highlight pairs)
- The plan:
  1. A short **literature review** of extractive summarization.
  2. **Bert** for extractive summarization + Code

SummaRuNNer: A Recurrent Neural Network based Sequence Model for Extractive Summarization of Documents (Nallapati et al., 2017)



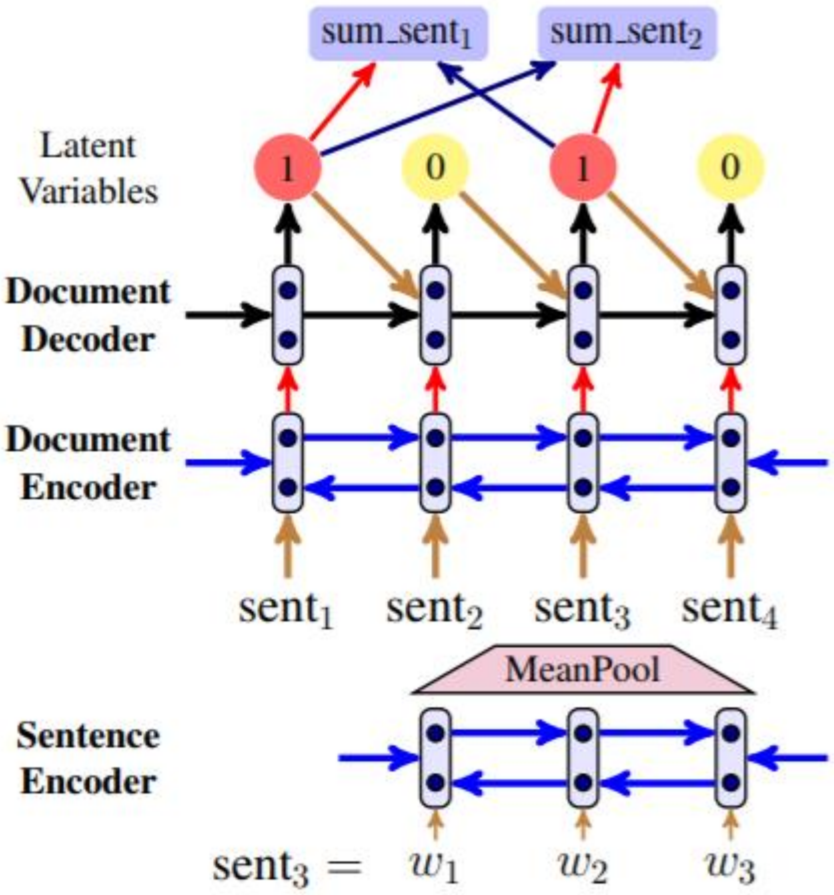
$$
\begin{aligned}
P(y_j = 1|\mathbf{h}_j, \mathbf{s}_j, \mathbf{d}) = \sigma(W_c\mathbf{h}_j \quad & \texttt{\#(content)} \\
+\mathbf{h}_j^T W_s \mathbf{d} \quad & \texttt{\#(salience)} \\
-\mathbf{h}_j^T W_r \tanh(\mathbf{s_j}) \quad & \texttt{\#(novelty)} \\
+W_{ap}\mathbf{p}_j^a \quad & \texttt{\#(abs. pos. imp.)} \\
+W_{rp}\mathbf{p}_j^r \quad & \texttt{\#(rel. pos. imp.)} \\
+b), \quad & \texttt{\#(bias term)} \qquad (6)
\end{aligned}
$$

Probability of the jth sentence to be a summary

- Treat extractive summarization as sentence classification problem.
- **Embedding layer** initialized with 100-d Word2Vec (Mikolov et al., 2013).
- 2 bidirectional-GRU (Cho et al., 2014) are used:
  - One at **word level** and an other at **sentence level**.
  - The **second bi-GRU** take as input the average-pooled, concatenated hidden states (forward and backward) of the **word level bi-GRU**.
- The average pooling of the concatenated hidden states of the **bi-directional sentence-level RNN** is fed to a **linear layer + tanh activation** to make the **document representation**.
- Last layer is a **Linear + sigmoid activation** and the loss function is Binary cross-entropy.

$$\mathbf{s}_j = \sum_{i=1}^{j-1} \mathbf{h}_i P(y_i = 1|\mathbf{h}_i, \mathbf{s}_i, \mathbf{d}).$$

Neural Latent Extractive Document Summarization
(Zhang et al., 2018)



Similar to previous architecture

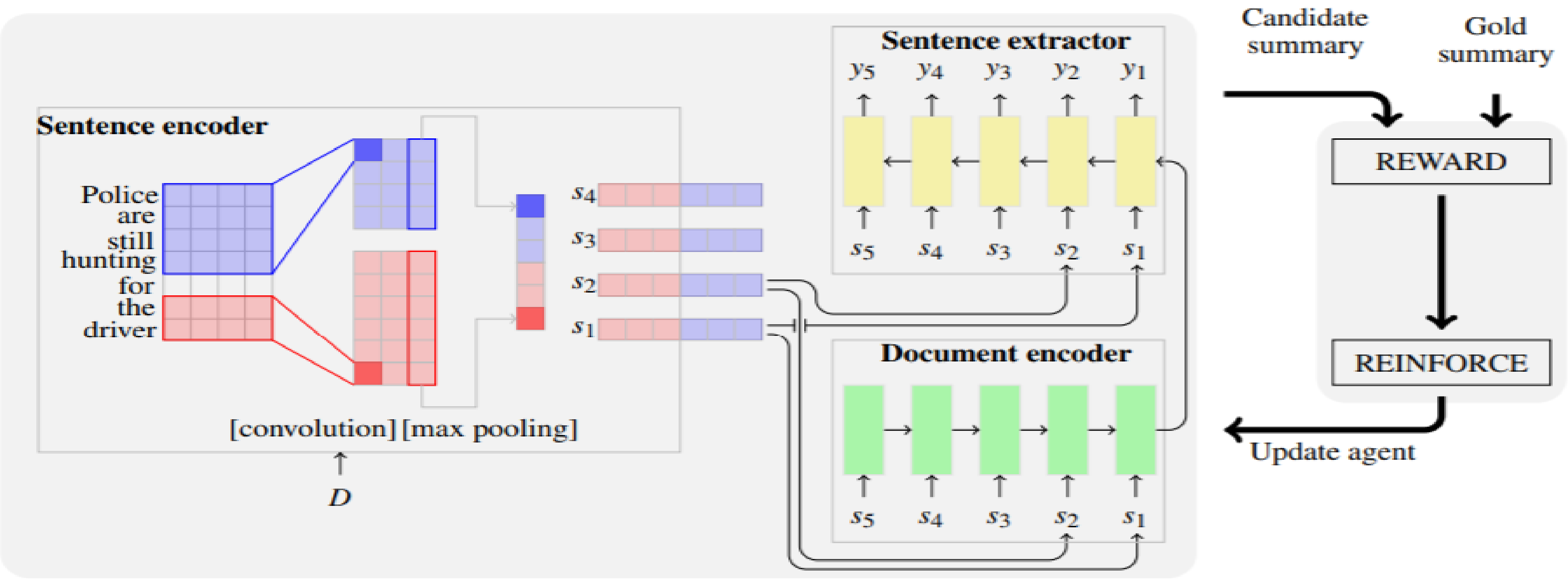
- Extractive summarization as sentence classification
- Input: Use 300-d fastext to initialize the word. embeddings
- bi-LSTM(Hochreiter and Schmidhuber, 1997) architecture for **sentence** and **document** encoders.
- An LSTM for **document decoder**
  - ➢ The previous prediction is used as additional information to the next time step.
- **Sigmoid layer** for prediction and **BCE** (or NLL) as loss function.

Ranking Sentences for Extractive Summarization with Reinforcement Learning *(**Narayan** et al., 2018)*

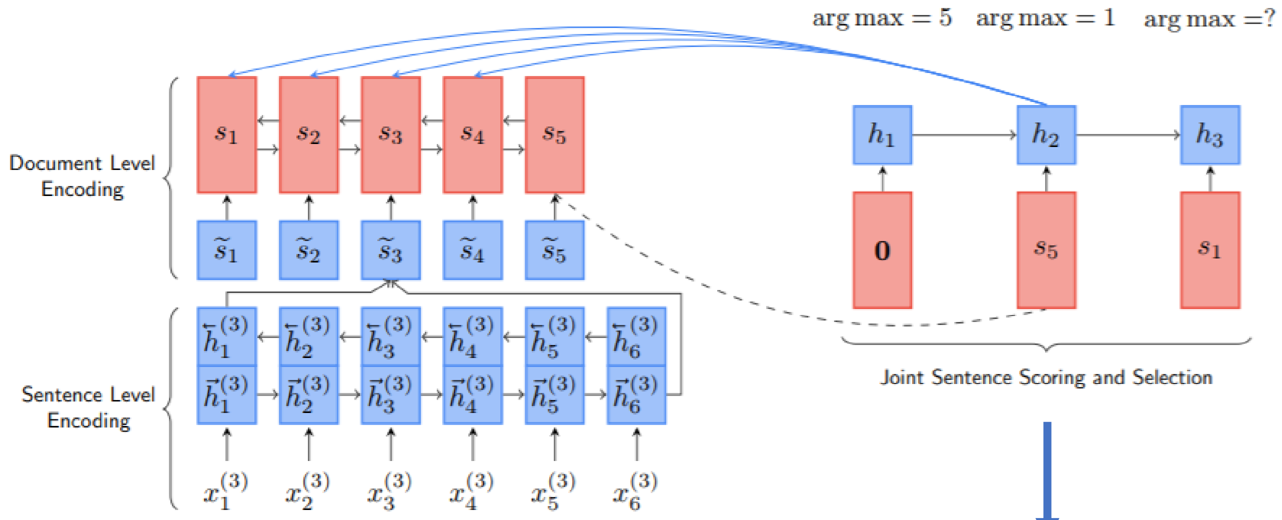
Idea: Optimize directly the ROUGE metric through a reinforcement learning objective

- 1D CNN sentence encoder

- LSTM for document encoder and sentence extractor
- A **sigmoid** is used to make prediction whether a sentence is a summary or not.
- The **rouge score (reward)** is computed using the candidate and gold summary.
- Use **Reinforce** algorithm to update the model.



Neural Document Summarization by Jointly Learning to Score and Select Sentences (Zhou *et al., 2018*)



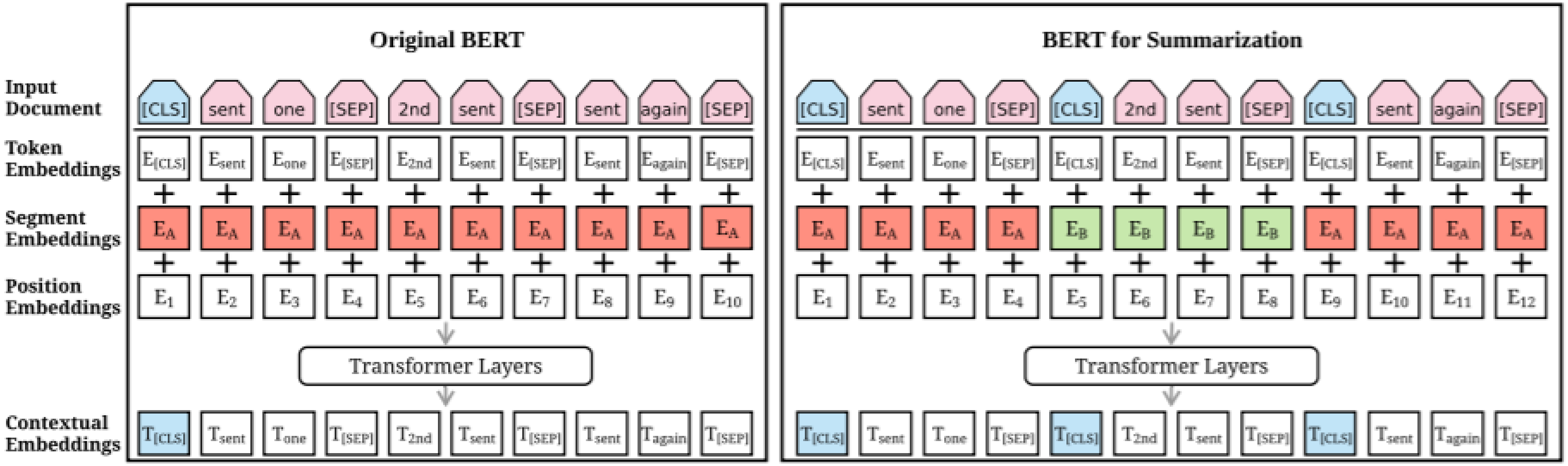
- Bidirectional-GRU for **sentence encoding**:
  - The sentence representation is the concatenation of the final hidden states (forward and backward)
- Bi-GRU for **document encoding**
- Another GRU Layer for **joint sentence scoring and selection**.

$$h_t = \mathrm{GRU}(s_{t-1}, h_{t-1})$$
$$\delta(S_i) = \mathbf{W}_s \tanh\left(\mathbf{W}_q h_t + \mathbf{W}_d s_i\right)$$

$$h_0 = \tanh\left(\mathbf{W}_m \overleftarrow{s}_1 + b_m\right)$$
$$S_0 = \varnothing$$
$$s_0 = \mathbf{0}$$

Fine-tune BERT for Extractive Summarization
(Liu, 2019; Liu and Lapata, 2019)

- BERTSUM use **pretrained BERT** model (Devlin et al., 2018) to represent documents for extractive summarization.
- → Text Summarization as token classification
- Each sentence of a document are separated by **[SEP]** and **[CLS]** tokens.
- The representation of t-th **[CLS]** token $T_{[CLS]}$ is used as representation for the t-th sentence.
- **Segment embeddings** to distinguish multiple sentences within a document.
- Each sentence representation $T_{[CLS]}$ is fed to a sigmoid layer to decide whether it should be considered as summary or not.
- The model loss is **binary cross-entropy** between the predicted and the true label.
- Variant: Add some **Transformer layers** (or even **LSTM**) between BERTSUM and the sigmoid layer.



CODE FOR BERTSUM