Intro

Summarization

Summarization is the task of producing a shorter version of one or several documents that preserves most of the input's meaning.

1. **Abstractive** summarization: paraphrase the corpus using novel sentences
2. **Extractive** summarization: concatenate extracts taken from a corpus into a summary

Simplification

Simplification consists of modifying the content and structure of a text in order to make it easier to read and understand, while preserving its main idea and approximating its original meaning.

Image source: nlpprogress.com

Today

Metrics: ROUGE [1]

Recall-Oriented Understudy for Gisting Evaluation

ROUGE is used to compare a system summary or translation against a set of reference human summaries:

$$\texttt{ROUGE}_n = \frac{\textit{number of overlapping n-grams}}{\textit{number of n-grams in reference summary}}$$

$$R_{LCS} = \frac{LCS(X,Y)}{|X|}, P_{LCS} = \frac{LCS(X,Y)}{|Y|}, \texttt{ROUGE}_L = \frac{(1+\beta^2)R_{LCS}P_{LCS}}{R_{LCS}+\beta^2 P_{LCS}},$$

where $LCS(X,Y)$ is the length of a longest common subsequence of X and Y.

Metrics: METEOR [2]

Metric for Evaluation of Translation with Explicit ORdering

METEOR is used to compare a system summary or translation against a set of reference human summaries:

$$\mathtt{P} = \frac{\textit{number of overlapping words}}{\textit{number of words in system summary}}$$

,

$$\mathtt{R} = \frac{\textit{number of overlapping words}}{\textit{number of words in reference summary}}$$

,

$$F_{mean} = \frac{10PR}{R+9P}, penalty = 0.5(\frac{\textit{number of chunks}}{\textit{number of overlapping words}})^3$$

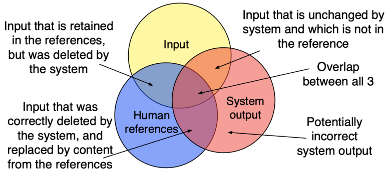
$$\mathtt{M} = F_{mean}(1-p)$$

Metrics: SARI [3]

System output against references and against the input sentence

SARI measures the goodness of words that are **added**, **deleted** and **kept** by the systems.

$$SARI = d_1 F_{add} + d_2 F_{keep} + d_3 P_{del}$$



Today

Datasets: CNN / Daily Mail [4], [5]

The dataset contains online news articles (781 tokens on average) paired with multi-sentence summaries (3.75 sentences or 56 tokens on average). The processed version contains 287,226 training pairs, 13,368 validation pairs and 11,490 test pairs.

STORY HIGHLIGHTS

- Trump will head to Texas on Tuesday
- The White House has yet to say where Trump will travel

Washington (CNN) — President Donald Trump struck a unifying tone Monday as he addressed the devastation in Texas wrought by Hurricane Harvey at the top of a joint news conference with Finland's president.

"We see neighbor helping neighbor, friend helping friend and stranger helping stranger," Trump said. "We are one American family. We hurt together, we struggle together and believe me, we endure together."

Trump extended his "thoughts and prayers" to those affected by the hurricane and catastrophic flooding that ensued in Texas, and also promised Louisiana residents that the federal government is prepared to help as the tropical storm makes its way toward that state.

"To the people of Texas and Louisiana, we are 100% with you," Trump said from the East Room of the White House.

Datasets: Webis-TLDR-17 [6]

The dataset contains 4 million content-summary pairs from Reddit.

Example Submission

Title: Ultimate travel kit
Body: Doing some traveling this year and I am looking to build the ultimate travel kit ... So far I have a Bonavita 0.5L travel kettle and AeroPress. Looking for a grinder that would maybe fit into the AeroPress. This way I can stack them in each other and have a compact travel kit.
TL;DR: What grinder would you recommend that fits in AeroPress?

Example Comment (to a different submission)

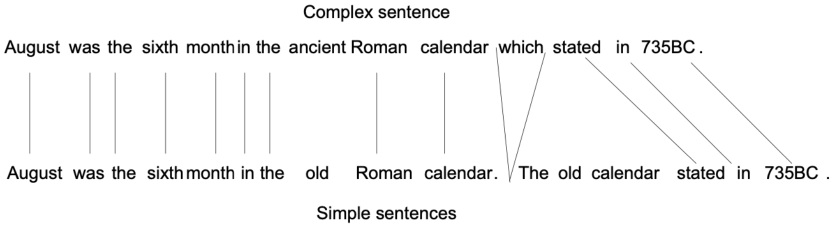
Body: Oh man this brings back memories. When I was little, around five, we were putting in a new shower system in the bathroom and had to open up the wall. The plumber opened up the wall first, then put in the shower system, and then left it there while he took a lunch break. After his break he patched up the wall and left, having completed the job. Then we couldn't find our cat. But we heard the cat. Before long we realized it was stuck in the wall, and could not get out. We called up the plumber again and he came back the next day and opened the wall. Out came our black cat, Socrates, covered in dust and filth.
TL;DR: plumber opens wall, cat climbs in, plumber closes wall, fucking meows everywhere until plumber returns the next day

Datasets: headline generation

1. Gigaword summarization dataset [7]
2. RIA news dataset [8]

Datasets: WikiSmall [9]

Main source for simplified sentences is Simple English Wikipedia.
WikiSmall is a parallel corpus with more than 108K sentence pairs from 65,133 Wikipedia articles, allowing 1-to-1 and 1-to-N alignments



Today

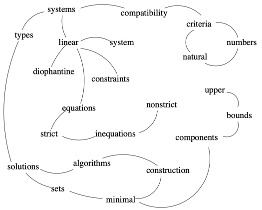
Extractive summarization

- **Key word** or **phrase** extraction: extraction meaningful words and phrases
- **Sentence classification**: for each sentence, decide whether it should be added to the summary (1), or not (0)
- Sentence classification **baselines**:
  1. Lead-1, 2, 3
  2. Oracle: add one sentence at a time incrementally to the summary, such that the Rouge score of the current set of selected sentences is maximized with respect to the entire gold summary [10]

TextRank [11]

1. Vertices: text units
2. Edges: relations that connect such text unit. Edges can be directed or undirected, weighted or unweighted
3. Calculate any graph centrality measure
4. Sort vertices based on their centrality value

Compatibility of systems of linear constraints over the set of natural numbers. Criteria of compatibility of a system of linear Diophantine equations, strict inequations, and nonstrict inequations are considered. Upper bounds for components of a minimal set of solutions and algorithms of construction of minimal generating sets of solutions for all types of systems are given. These criteria and the corresponding algorithms for constructing a minimal supporting set of solutions can be used in solving all the considered types systems and systems of mixed types.



Keywords assigned by TextRank:
linear constraints; linear diophantine equations; natural numbers; nonstrict inequations; strict inequations; upper bounds

Keywords assigned by human annotators:
linear constraints; linear diophantine equations; minimal generating sets; non–strict inequations; set of natural numbers; strict inequations; upper bounds

$$\mathrm{PR}(V_i) = (1-d) + d \times \sum_{V_j \in \mathrm{In}(V_i)} \frac{w_{ij}}{\sum_{V_k \in \mathrm{Out}(V_j)} w_{jk}} \mathrm{PR}(V_j)$$

SummaRuNNer [10]

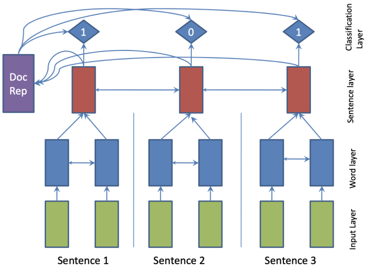
- Doc representation:

$$d = tanh(W_d \frac{1}{N_d} \sum_{j=1}^{N} {}_d[h_f, h_b] + b)$$

- Summary representation:

$$s_j = \sum_{i}^{j-1} h_i P(y_i = 1 | h_i, s_i, d)$$

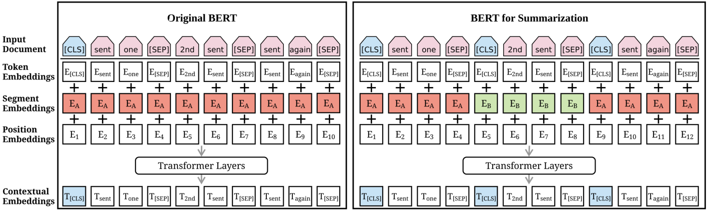
- Sentence classification:



$$P(y_j = 1 | h_j, s_j, d) = \sigma(W_c h_j + h_j^T W_s d - h_j^T W_r tanh(s_j) + W_{ap} p_j^a + W_{rp} p_j^r)$$

Criteria: (content + salience - redundancy + abs postion + rel position)

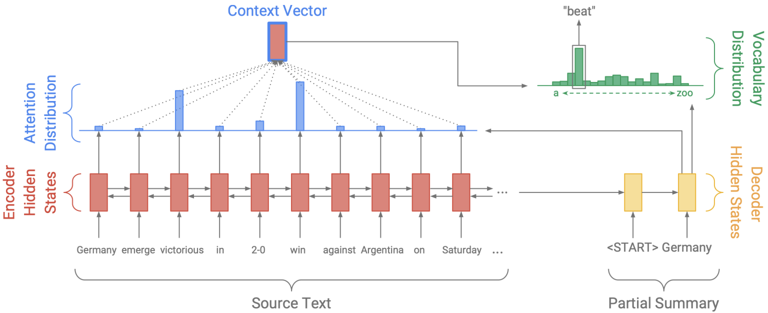
BERTSumExt [12]



1. Document: $[\text{sent}_1, \text{sent}_1, \ldots, \text{sent}_m]$
2. Alternate segment embeddings: $[E_A, E_B, E_A, E_B, E_A]$
3. $sent_i$ representation: vector of the i-th [CLS] symbol, h^L, where L stands for the layer
4. Output layer: $\hat{y} = \sigma(Wh_i^L + b)$

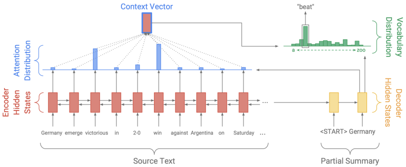
Get to the Point [13]

Sequence-to-sequence attentional model



Get to the Point [13]

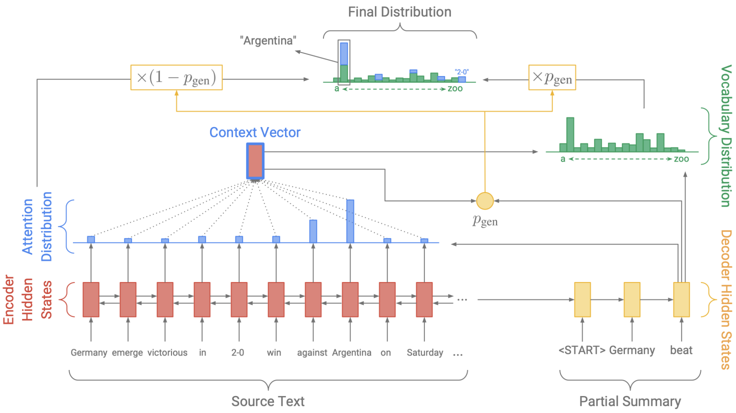
Sequence-to-sequence attentional model



- Bahdanau attention: $e_i^t = v^T \tanh(W_h h_i + W_s s_t + b_{attn})$, $a_t = \text{softmax}(e_t)$
- Context vector: $h_t = \sum_i a_i^t h_i$
- Vocabulary distribution: $P_{vocab} = \text{softmax}(V'(V[s_t, h_t] + b) + b)$
- NLL loss: $-\frac{1}{T} \sum_{t=0}^{T} \log P(w_t^*)$

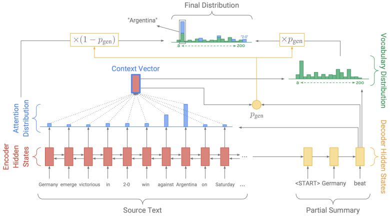
Get to the Point [13]

Pointer-generator model



Get to the Point [13]

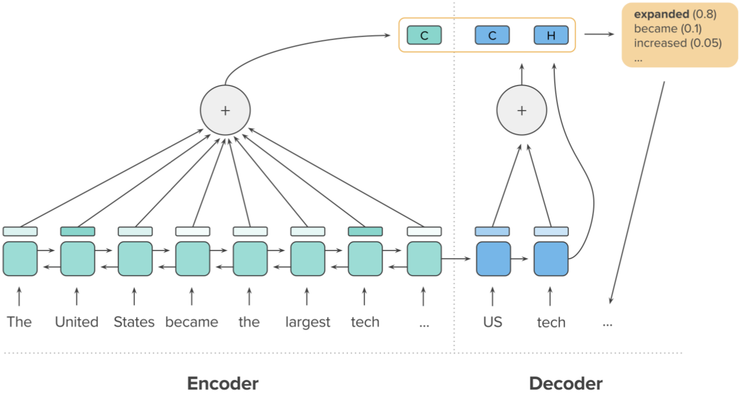
Pointer-generator model



- Generation probability: $p_{gen} = \sigma(w_{h^*}^T h_t + w_s^T s_t + w_x^T x_t + b_{ptr})$
- p_{gen} is used to switch between sampling from P_{vocab} or copying by sampling a^t
- P(w) $= p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i^t$

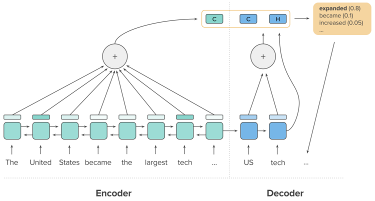
A Deep Reinforced Model

for Abstractive Summarization [14]



A Deep Reinforced Model

for Abstractive Summarization [14]



- Intra-temporal attention:
  $e_{ti} = {h_t^d}^T W_{attn}^e h_i^e$,
  $\alpha_{ti}^e = \text{softmax}(e_{ti})$,
  $c_t^e = \sum_{i=1}^n \alpha_{ti}^e h_i^e$
- Intra-decoder attention:
  $e_{tt'} = {h_t^d}^T W_{attn}^d h_i^d$,
  $\alpha_{tt'}^d = \text{softmax}(e_{tt'})$,
  $c_t^d = \sum_{i=j}^{t-1} \alpha_{tj}^d h_k^d$

A Deep Reinforced Model

for Abstractive Summarization [14]

- Token generation:

$$p(y_t|u_t = 0) = \text{softmax}(W_{out}[h_t^d, c_t^e, c_t^d] + b_{out})$$

- Pointer:

$$p(y_t = x_i|u = 1) = \alpha_{ti}^e$$

$$p(u_t = 1) = \sigma(W_u[h_t^d, c_t^e, c_t^d] + b_u)$$

- Probability distribution for the output token:

$$p(y_t) = p(u_t = 1)p(y_t|u_t = 1) + p(u_t = 0)p(y_t|u)t = 0)$$

- Sharing decoder weights: $W_{out} = \tanh(W_{emb}W_{proj})$

A Deep Reinforced Model

for Abstractive Summarization [14]

- Hybrid learning objective:

$$L_{mixed} = \gamma L_{rl} + (1\gamma) L_{ml}$$

- Teacher forcing:

$$L_{ml} = \sum_{t=1}^{n'} \log p(y_t | y_1, \dots, y_{t-1}, x)$$

- Policy learning:

$$L_{rl} = (r(\hat{y}) - r(y^s)) \sum_{t=1}^{n'} \log p(y_t^s | y_1^s, \dots, y_{t-1}^s, x),$$

where r is a reward function, $\hat{y}$ is the baseline output, obtained by maximizing the output probability distribution at each time step.

Pegasus

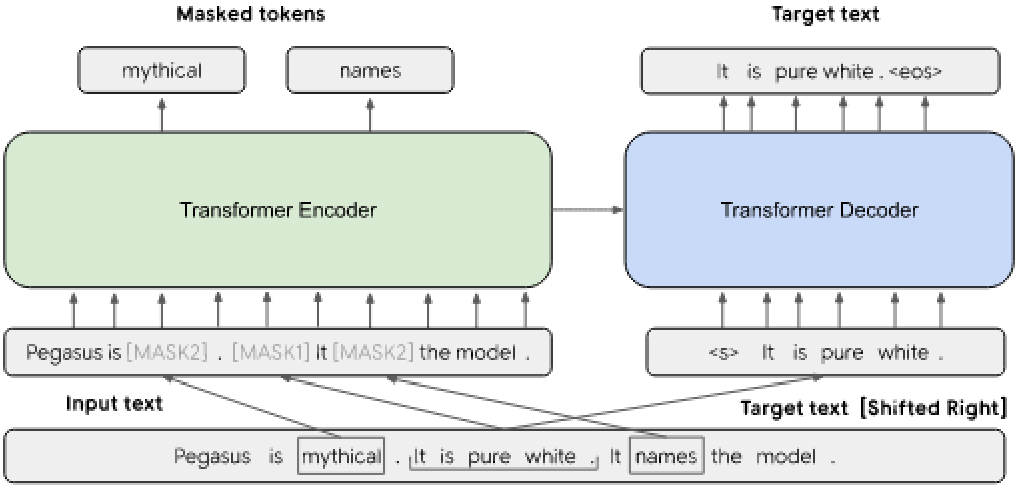
1. Standard Transformer encoder-decoder

2. Self-supervised task:
    1. In pre-training important sentences are extracted and removed from an input document
    2. Extracted sentences joined together as one output sequence from the remaining sentences.
    3. The task is to recover the extracted sentences

3. Encoder outputs masked tokens and decoder generates gap sentences.

4. Different strategies for selecting gap sentences.

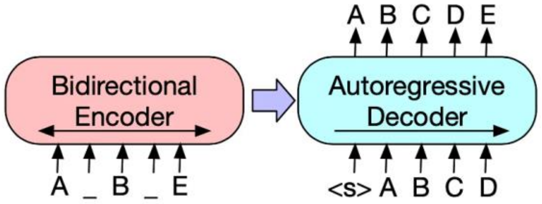
Pegasus



Masked tokens
mythical
names
Target text
It is pure white . <eos>
Transformer Encoder
Transformer Decoder
Pegasus is [MASK2] . [MASK1] It [MASK2] the model .
<s> It is pure white .
Input text
Target text [Shifted Right]
Pegasus is mythical . It is pure white . It names the model .

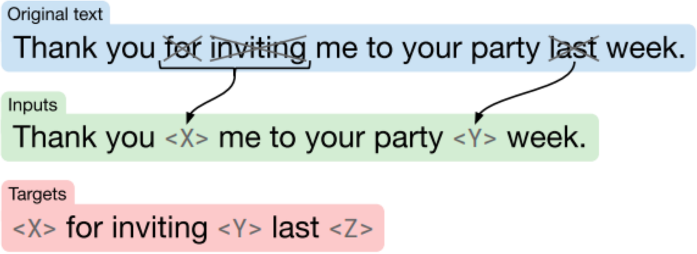
BART

1. Denoising encoder-decoder architecture
2. Extended nosing techniques
    1. **Text Infilling:** A fixed number of contiguous tokens are deleted and replaced with a single [MASK] token. The model must learn the content of the missing tokens and the number of tokens.
    2. **Sentence Permutation:** Sentences (separated by full stops) are permuted randomly. This helps the model to learn the logical entailment of sentences.
3. About 30% of tokens being masked and all sentences permuted.
4. Autoregressive decoder: looking only on the previous tokens



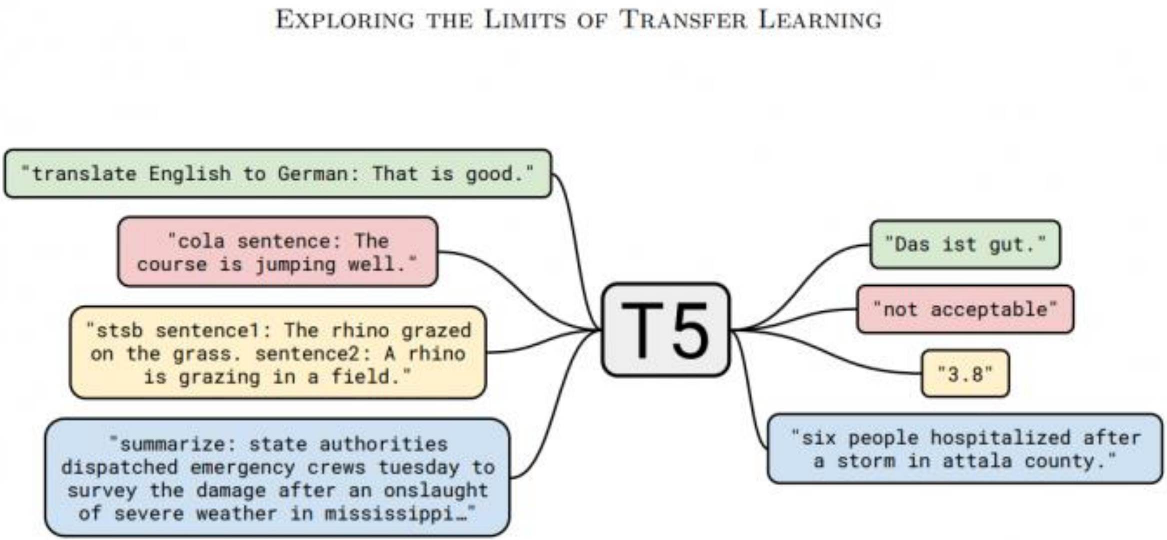
T5

1. T5 (Text-To-Text Transfer Transformer) is a transformer model that is trained in an end-to-end manner with **text as input** and **modified text as output**, in contrast to BERT-style models that can only output either a class label or a span of the input.
2. Trained with MLM but T5 replaces multiple consecutive tokens with the single Mask Keyword, unlike BERT which uses Mask token for each word.
3. Expects a prefix before the input text to understand the task given by the user. For example, “**summarize**:” for the summarization.



T5’s mask language modeling (Raffel et al., 2019)

T5

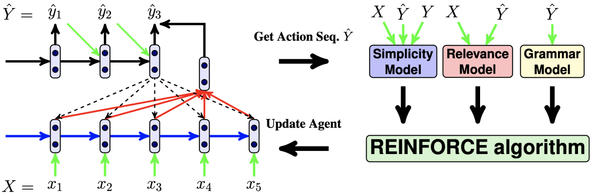


Exploring the Limits of Transfer Learning
"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Today

DRESS: Deep REinforcement Sentence Simplification [15]

X: In 1883, Faur married Marie Fremiet, with whom he had two sons.
Y: In 1883, Faur married Marie Fremiet. They had two sons.



Agent reads the source sentence X and takes an **action** $\hat{y} \in V$ according to a **policy** $P_{RL}(\hat{y}_t|\hat{y}_{1:t-1}, X)$ until EOS is produced. $\hat{Y}$ is the simplified output. The **reward** $r(\hat{Y})$ is received and the agent is updated.

DRESS: Deep REinforcement Sentence Simplification [15]

The **reward** captures simplicity, relevance, and fluency:

$$r(X, Y, \hat{Y}) = \lambda^S r^S + \lambda^R r^R + \lambda^F r^F, \lambda^S, \lambda^R, \lambda^F \in [0, 1]$$

- **Simplicity**: $r^S = \beta \mathrm{SARI}(X, \hat{Y}, Y) + (1 - \beta)\mathrm{SARI}(X, Y, \hat{Y})$
- **Relevance**: $r^R = \cos(q_x, q_{\hat{Y}})$
  Sequence auto-encoder produces vector representations of X, $\hat{Y}$.
- **Fluency**: $r^F = \exp(\frac{1}{|\hat{Y}|}\sum_{i=1}^{|\hat{Y}|} \log P_{LM}(\hat{y}_i|\hat{y}_{0:i-1}))$

Use REINFORCE algorithm to find an agent that maximizes the expected reward:

$$\nabla\mathcal{L} \approx \sum_{t=1}^{\hat{Y}} \nabla \log P_{RL}(\hat{y}_t|\hat{y}_{1:t-1}, X)[r(\hat{y}_{1:|\hat{Y}|} - b_t)]$$

DRESS: Deep REinforcement Sentence Simplification [15]

Pre-train encoder-decoder to align lexical substitution:
Source sentence: $X = (x_1, x_2, \ldots, x_{|X|}) \rightarrow (v_1, v_2, \ldots, v_{|X|})$ are hidden states of a LSTM
Target sentence: $Y = (y_1, y_2, \ldots, y_{|Y|})$
Alligment scores: $\alpha_{t1}, \alpha_{t2}, \ldots, \alpha_{t|X|}$
Lexical substitution probability:

$$P_{LS}(y_t|X, \alpha_t) = \text{softmax}(W_l \sum_{i=1}^{|X|} \alpha_{ti} v_i), W_l \in \mathbb{R}^{|V| \times d}$$

Lexical simplification + RL learning:

$$P(y_t|y_{1:t1}, X) = (1 - \eta) P_{RL}(y_t|y_{1:t1}, X) + \eta P_{LS}(y_t|X, \alpha_t), \eta \in [0, 1]$$

DRESS: Deep REinforcement Sentence Simplification [15]

- Pretrain the agent with NLL objective
- Curriculum learning strategy: use NLL objective to train the first L tokens and apply the RL algorithm to the $(L+1)$-th tokens onwards. Every two epochs decrese $L = -3$, at $L = 0$ terminate
- Was trained on `https://newsela.com/`

Today

Take away messages

- Summarization and simplification are two monolingual sequencer transformation tasks.
- Methods similar to MT are used to approach them.
- Special metrics are introduced to evaluate the quality of the models.
- RL-based methods are very promising.

Reference I

C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

W. Xu, C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch, "Optimizing statistical machine translation for text simplification," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 401–415, 2016.

K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, *Teaching machines to read and comprehend*, 2015. arXiv: 1506.03340 [cs.CL].

Reference II

R. Nallapati, B. Zhou, C. dos Santos, Ç. GuÌ‡lçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 280–290. DOI: 10.18653/v1/K16-1028. [Online]. Available: https://www.aclweb.org/anthology/K16-1028.

M. Völske, M. Potthast, S. Syed, and B. Stein, "TL;DR: Mining Reddit to learn automatic summarization," in *Proceedings of the Workshop on New Frontiers in Summarization*, Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 59–63. DOI: 10.18653/v1/W17-4508. [Online]. Available: https://www.aclweb.org/anthology/W17-4508.

Reference III

- A. M. Rush, S. Chopra, and J. Weston, *A neural attention model for abstractive sentence summarization*, 2015. arXiv: 1509.00685 [cs.CL].
- D. Gavrilov, P. Kalaidin, and V. Malykh, *Self-attentive model for headline generation*, 2019. arXiv: 1901.07786 [cs.CL].
- Z. Zhu, D. Bernhard, and I. Gurevych, "A monolingual tree-based translation model for sentence simplification," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 1353–1361. [Online]. Available: https://www.aclweb.org/anthology/C10-1152.
- R. Nallapati, F. Zhai, and B. Zhou, *Summarunner: A recurrent neural network based sequence model for extractive summarization of documents*, 2016. arXiv: 1611.04230 [cs.CL].

Reference IV

R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 404–411. [Online]. Available: https://www.aclweb.org/anthology/W04-3252.

Y. Liu and M. Lapata, *Text summarization with pretrained encoders*, 2019. arXiv: 1908.08345 [cs.CL].

A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1073–1083. DOI: 10.18653/v1/P17-1099. [Online]. Available: https://www.aclweb.org/anthology/P17-1099.

Reference V

- R. Paulus, C. Xiong, and R. Socher, *A deep reinforced model for abstractive summarization*, 2017. arXiv: 1705.04304 [cs.CL].
- X. Zhang and M. Lapata, "Sentence simplification with deep reinforcement learning," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 584–594.