Part 2: Models - (Text simplification, Collaborative and Instruction-based text rewriting)

Claire Gardent (CNRS/LORIA, France)

Simplifying Text

MLM on Simple Text Spans
Planning Text Simplification

Simplifying Text

Deleting
Rephrasing
Splitting
Re-ordering

Preserving discourse coherence

Input

The Zibelemärit is an annual market with aspects of a fair in the old town of Bern, Switzerland. It takes place the fourth Monday in November.

Historical research indicates that the "Zibelemärit" *originated in the 1850s* with "marmettes", farmer's wives from around Murten, coming to Bern at around St. Martin's Day to sell their produce; however, a persistent local legend *holds* that the "Zibelemärit" is a much older *festivity*. According to this legend, the Bernese *awarded* the people from the nearby city of Fribourg the right to sell onions in the city in reward for their aid after a fire destroyed much of Bern in 1405.

As the name indicates, it is mainly onions that are sold on the "Zibelemärit". Bernese farmers, who are proud of their decorative onion tresses and onion wreaths, also sell other onion products on the market, including Zwiebelkuchen (onion pie), onion soup and onion sausages. Decorative chains of sugar onions are also popular with children.

The "Zibelemärit" opens very early in the day, at around 03:00 to 04:00. Later in the morning, the narrow alleys are usually packed tight with people, which is what the Bernese call the "Gstungg". A general confetti battle in which mostly children *participate ensues* at four o'clock in the afternoon, *officially ending the market*.

Output

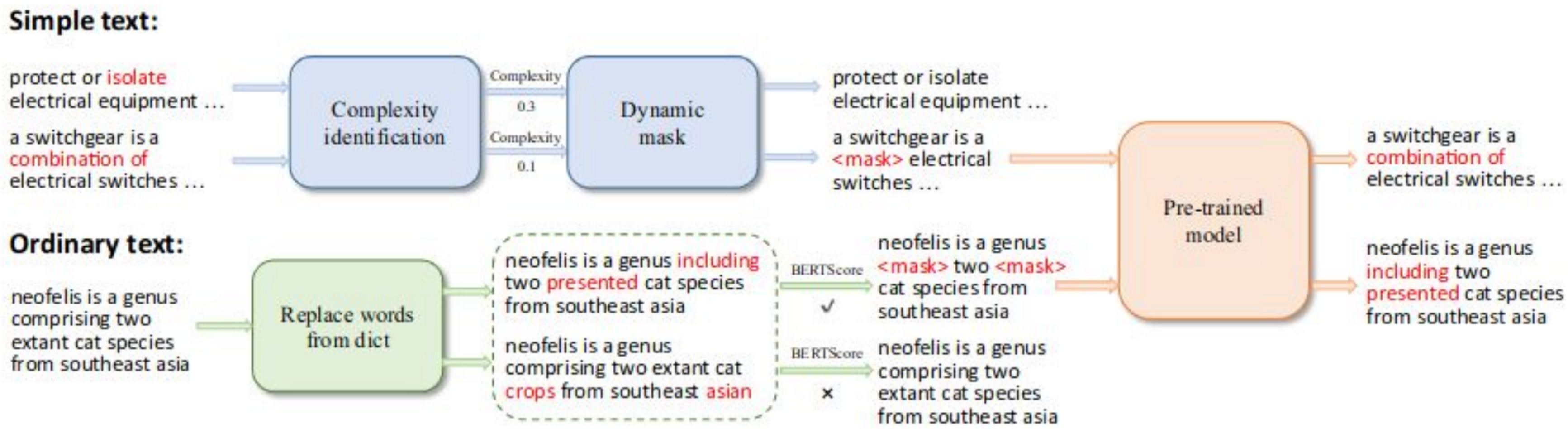
The Zibelemärit is an annual market in the old town of Bern, Switzerland. It takes place the fourth Monday in November.

The "Zibelemärit" *started around 150 years ago* with "marmettes", farmer's wives. <SPLIT> They came to Bern at around St. Martin's Day to sell their produce. <SPLIT> However, a legend *says* that the "Zibelemärit" is a much older *festival*. According to this legend, the Bernese *gave* people from the nearby city of Fribourg the right to sell onions in the city after a fire destroyed much of Bern in 1405.

In this country, it is mainly onions that are sold on the "Zibelemärit." Bernese farmers also sell other products, including Zwiebelkuchen (onion pie), onion soup and onion sausages. Decorative chains of sugar onions are also popular with children.

The "Zibelemärit" opens very early in the day, at around 03:00 to 04:00. Later in the morning, the narrow alleys are usually packed tight with people, which is what the Bernese call the "Gstungg." A general confetti battle in which mostly children *fight breaks out* at four o'clock in the afternoon *to end the market*.

SimpleBART: Continuous Pre-Training



Key idea: ***Continue pre-training a pre-trained model so that it learns to generate simple text.***

- MLM on simple text spans
  - ***Simple texts:*** identify simple text spans.
  - ***Ordinary text:*** use a dictionary to replace complex words in ordinary texts with simple words.

Renliang Sun, Wei Xu and Xiaojun Wan. "Teaching the Pre-trained Model to Generate Simple Texts for Text Simplification", ACL Findings 2023.

The D-Wikipedia Dataset

Based on the English Wikipedia and Simple English Wikipedia

- Wikipedia abstract aligned with Simple Wikipedia abstract
- Input and output length capped to 1K words

Training/Dev/Test

- 132K/3K/8K text pairs

	D-Wikipedia	
	Original	Simple
Total articles	143,546	
Total sentences	707,470	581,513
Total words	20,349,706	11,286,155
Avg words per article	141.76	78.62
-Compression ratio		0.55
Avg words per sent	28.76	19.41
-Compression ratio		0.67

SimpleBART

D-Wikipedia dataset (Sun et al., 2021)

- Wikipedia/Simple Wikipedia articles
- Maximum 1K input tokens
- Training/Dev/Test: 133K/3K/8K

Models

- BertSumextabs: text summarization (Liu and Lapata, 2019).
- BART-Large
- BART-CP: MLM fine-tuning on D-Wikipedia train
- SimpleBART: fine-tuning on D-Wikipedia train, MLM on simple text spans

.

D-Wikipedia	D-SARI↑	D_{keep}	D_{del}	D_{add}
BertSumextabs	39.88	35.71	**72.06**	11.87
BART	39.84	35.87	70.26	13.40
BART-CP	40.13	36.21	71.54	12.64
SimpleBART	**41.64**	**37.91**	71.96	**15.04**

Table 3: Results on the D-Wikipedia test set

Compared to standard MLM, MLM on simple text spans improves simplification

Renliang Sun, Wei Xu and Xiaojun Wan. “Document-Level Text Simplification: Dataset, Criteria and Baseline”, EMNLP 2021.
Renliang Sun, Wei Xu and Xiaojun Wan. “Teaching the Pre-trained Model to Generate Simple Texts for Text Simplification”, ACL Findings 2023.

PGConBART - Context-Sensitive, Plan-Guided Simplification

The planner predicts a ***Simplification Plan*** i.e., a sequence of simplification operations

$$c1, \ldots, cn \Rightarrow \hat{o}, \ldots, \hat{o}n$$

with $\hat{o}_i \in \{$***copy, rephrase, split, delete***$\}$

Liam Cripwell, Joël Legrand and Claire Gardent, “Document-Level Planning for Text Simplification”, EACL 2023. “Context-Aware Document Simplification”, ACL 2023

Context-Sensitive Plan-Guided Simplification

The planner predicts a ***Simplification Plan*** i.e., a sequence of simplification operations

$$c1, \ldots, cn \Rightarrow \hat{o}, \ldots, \hat{o}n$$

with $\hat{o}_i \in \{$*copy, rephrase, split, delete*$\}$

Simplification is guided by this plan.

$$c_i, \hat{o}_i \Rightarrow s_i$$

Liam Cripwell, Joël Legrand and Claire Gardent, “Document-Level Planning for Text Simplification”, EACL 2023. “Context-Aware Document Simplification”, ACL 2023

Context-Sensitive Plan-Guided Simplification

The planner predicts a ***Simplification Plan*** i.e., a sequence of simplification operations

$$c_1, \ldots, c_n \Rightarrow \hat{o}, \ldots, \hat{o}_n$$

with $\hat{o}_i \in \{$*copy, rephrase, split, delete*$\}$

Simplification is guided by this plan.

$$c_i, \hat{o}_i \Rightarrow s_i$$

The model uses both ***LOCAL*** and ***GLOBAL*** context.

Liam Cripwell, Joël Legrand and Claire Gardent, "Document-Level Planning for Text Simplification", EACL 2023. "Context-Aware Document Simplification", ACL 2023

Local and Global Context

Simplification Operations have different requirements

Splitting mainly depends on the sentence internal structure (*LOCAL Context)*

- The man ***who*** sleeps snores → The man sleeps. He snores.
- John went shopping ***after*** he left work → John left work. Afterwards he went shopping.

Other operations (***delete, copy, rephrase***) depend on the sentence context *(GLOBAL Context)*

Liam Cripwell, Joël Legrand and Claire Gardent, “Document-Level Planning for Text Simplification”, EACL 2023. “Context-Aware Document Simplification”, ACL 2023

Planning Simplification Operations

RoBERTa classifier with cross-attention over the global context

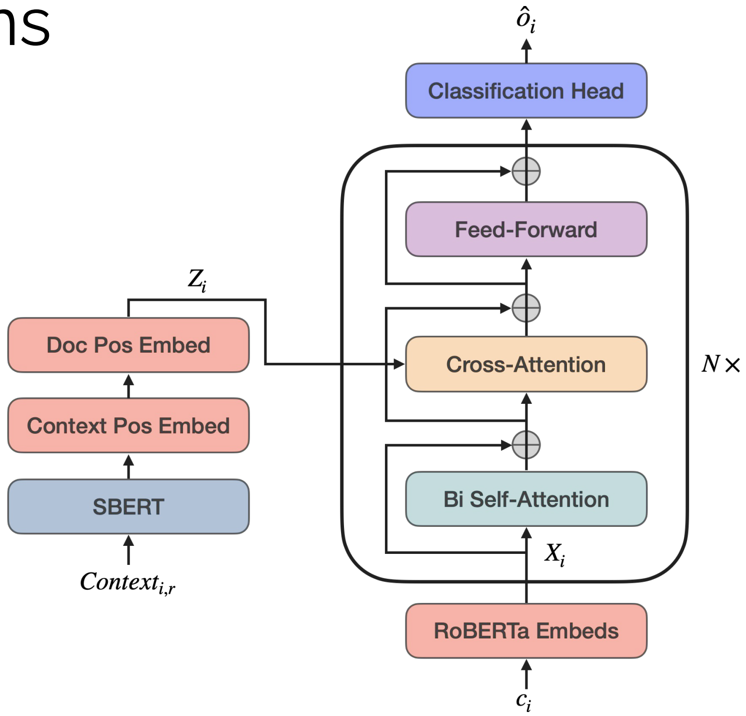
Planning viewed as Classification

Local Context

- ***Token level*** encoder of the sentence to be simplified

Global Context

- fixed window of ***Sentence level embedding*** (SBERT) for surrounding sentences



Context positional embedding: relative distance of a given sentence from the input sentence

Document positional embedding: the document quintile (1-5) that a given sentence falls into

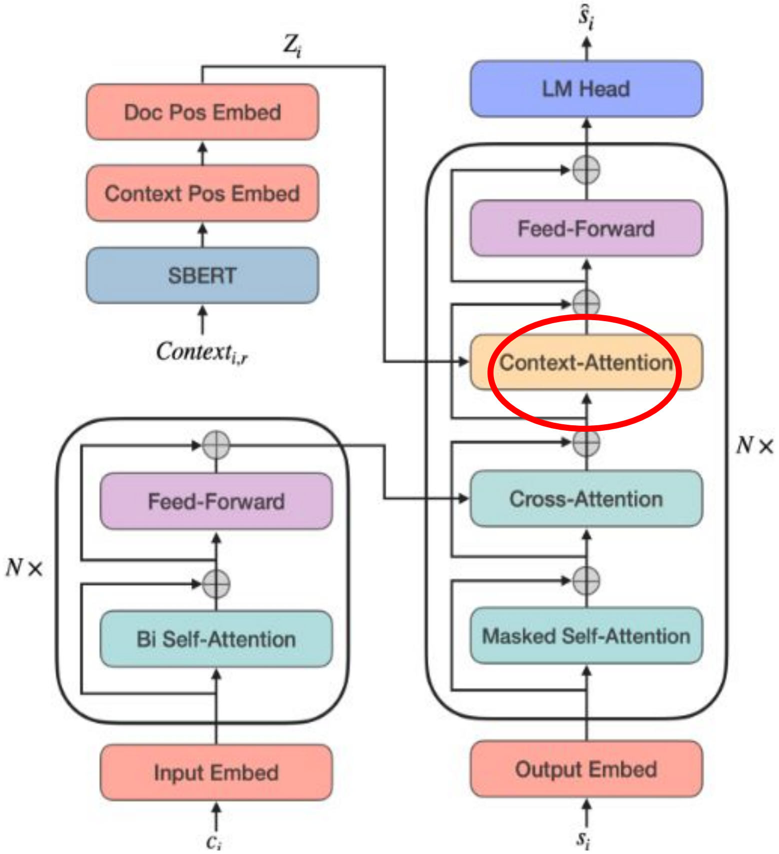
Liam Cripwell, Joël Legrand and Claire Gardent, "Document-Level Planning for Text Simplification", EACL 2023. "Context-Aware Document Simplification", ACL 2023

Context Aware BART (ConBART)

BART modified to attend over both the ***text input*** and the ***global context***

Same ***Local*** and ***Global Context*** as in the planner

- ***Token level*** encoder of the sentence to be simplified
- Fixed window of ***Sentence level embedding*** (SBERT) for surrounding sentences



Liam Cripwell, Joël Legrand and Claire Gardent, "Document-Level Planning for Text Simplification", EACL 2023. "Context-Aware Document Simplification", ACL 2023

Datasets

Newsela-auto consists of news articles, each manually rewritten at five different levels of simplification, corresponding to discrete reading levels (0-4) of increasingly simplicity.
Aligned pairs are created by pairing every article version with each other version corresponding to a higher reading level.

Wiki-auto gathers three simplification datasets which were automatically-collated from English Wikipedia and Wikipedia

In both datasets, the input document was automatically aligned with the output simplification at both the sentence and the paragraph level.

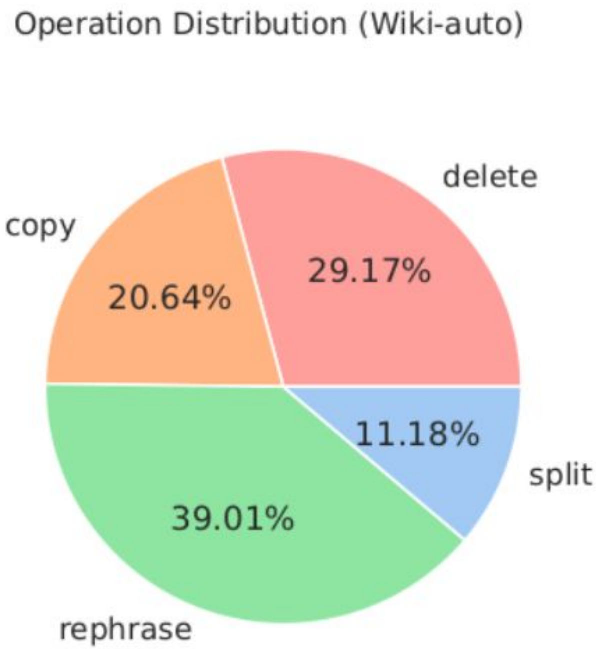
	Wiki-auto	Newsela-auto
# Doc Pairs	85,123	18,319
# Sent Pairs	461,852	707,776
Avg. $\lvert C\rvert$	155.51	868.98
Avg. $\lvert S\rvert$	97.72	674.94
Avg. $\lvert c_i\rvert$	28.64	22.49
Avg. $\lvert s_i\rvert$	21.57	15.84
Avg. n	5.43	38.64
Avg. k	4.53	42.60

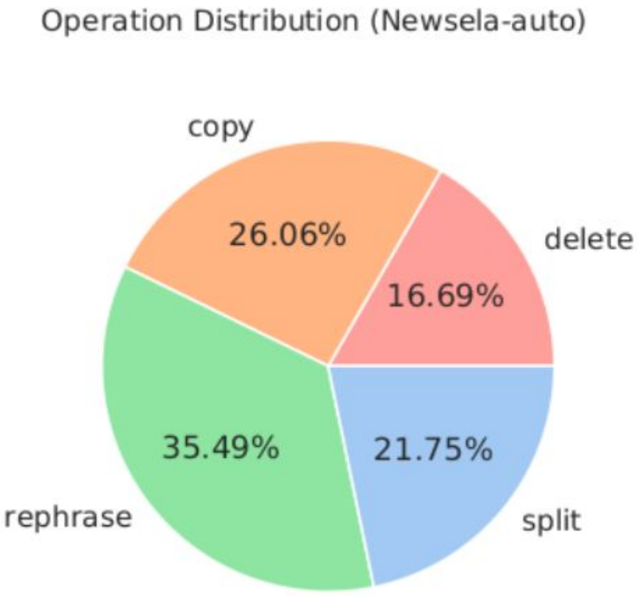
- n: the number of sentences in C
- k: the number of sentences in S

Newsela
- Input documents are longer
- Smaller dataset

Liam Cripwell, Joël Legrand and Claire Gardent, “Document-Level Planning for Text Simplification”, EACL 2023. “Context-Aware Document Simplification”, ACL 2023

Distribution of Simplification Operations





Input Text, Contexts and Models

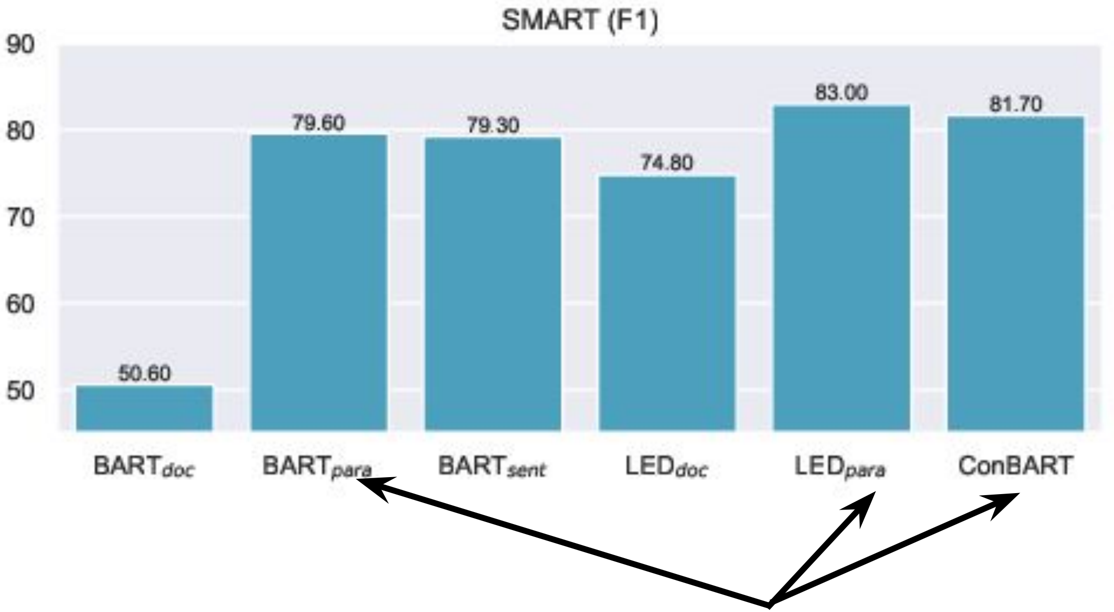
No Planning

- Document-level input: BARTdoc, LongformerDoc
- Paragraph-level input: BARTpara,LongFormerPara
- Sentence-level input: BARTsent
- Sentence + Global Context: ConBART

Plan Guided Models: O -> M

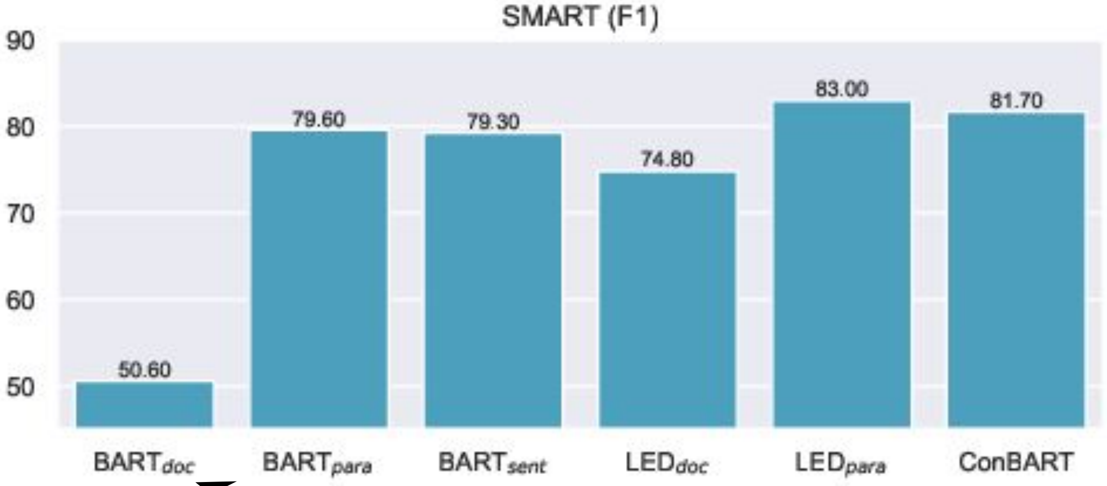
- O, a predicted simplification plan
- M, a simplification model (BART, LongFormer, ConBART)

Which contexts helps most ?



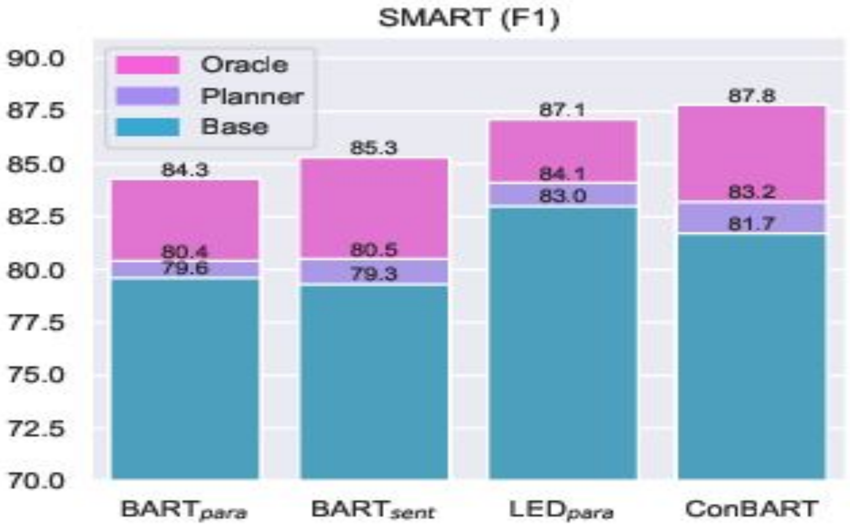
The best three models use a medium size context
(either a paragraph or a sentence window)

Which contexts helps most ?



Full Document context does not work well

Planning helps



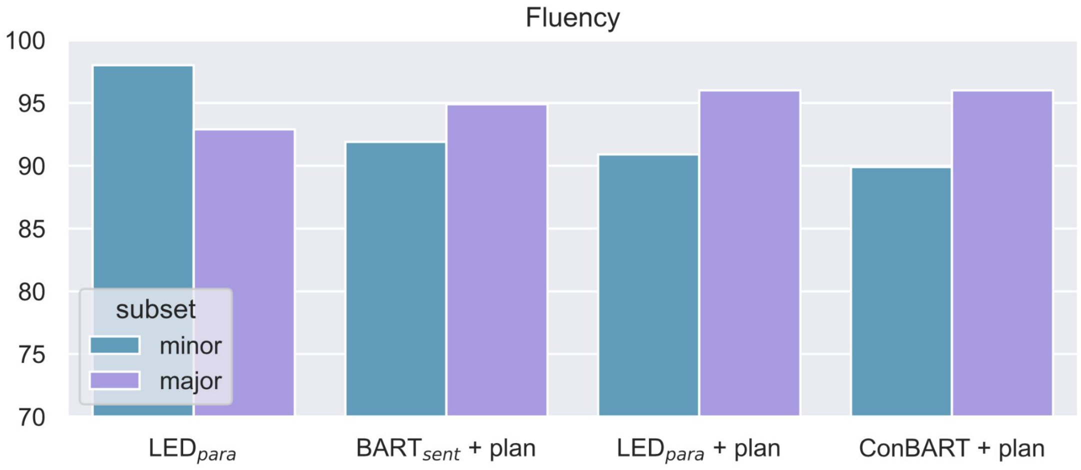
Planning

- systematically improves performance
- needs improving

Human Evaluation

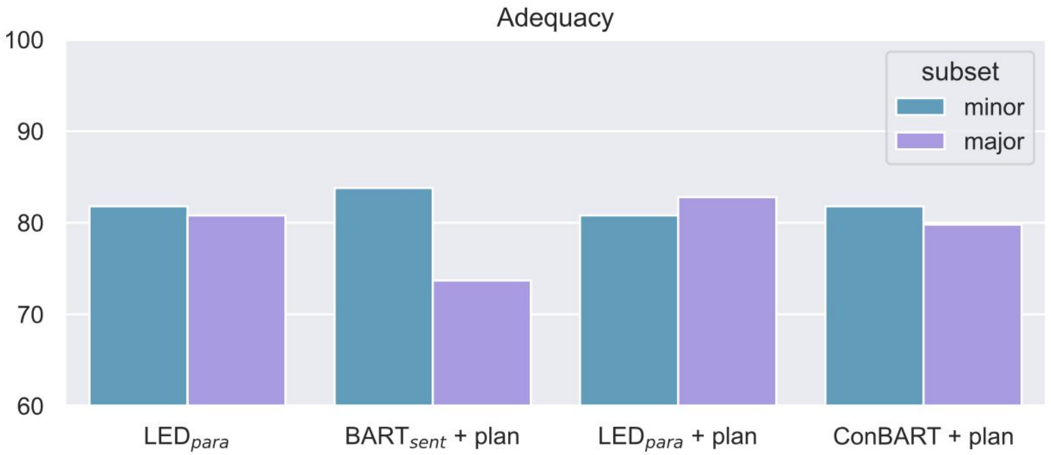
- On paragraphs
  - 33 complex paragraphs from each non-adjacent reading-level transition pairing
  - 198 paragraphs in total
  - 50% Minor: reading-level transition of two (0-2, 1-3 etc)
  - 50% Major: reading-level transition higher than two (0-3, 1-4 etc)
- Yes/No judgments on fluency, adequacy, simplicity
- Score = proportion of positive judgments
- References and outputs from 4 high performing systems
  - PGDyn, LongformerPara, O→LongformerPara, O→$ConBART$)
- 990 outputs in total

Human Evaluation



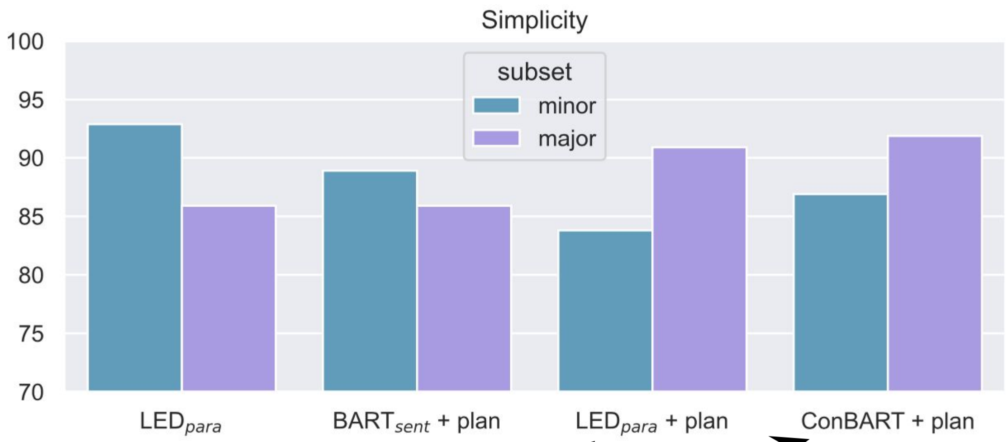
All systems achieve high fluency – not surprising given modern LM
Planning improves fluency on MAJOR cases (cases requiring higher degrees of simplification)

Human Evaluation



Window- (ConBART) and paragraph-based models are better at maintaining adequacy

Human Evaluation

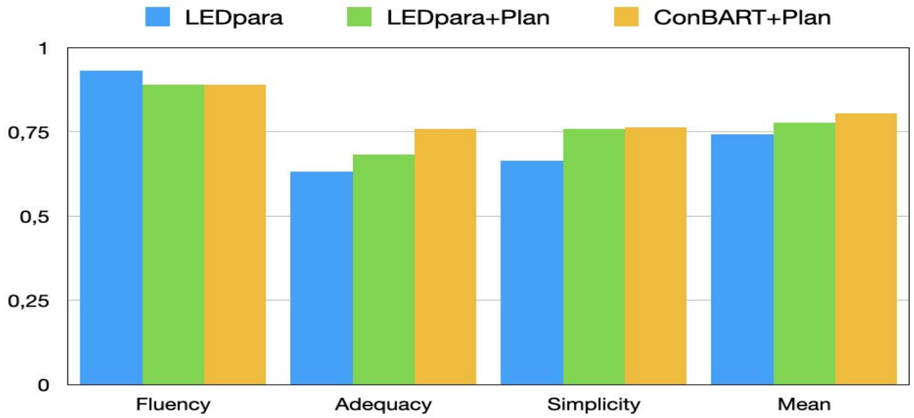


Window/paragraph-based models + Planning yields high simplicity in major cases (overcoming conservativity?)

Generalising to OOD Data

Training on Newsela

Testing on Wiki-auto



Planning helps on unseen domains.

Paragraph-based models are less adaptable to unseen domains

Summary

Planning Simplification operations and having a ***window-based context*** helps

- improve document simplification
- generalising to new domains
- handling more drastic simplification (MAJOR cases)

Summarisation and Simplification

Using Summarisation Data to Simplify Text

Mining Summarisation Data for Simplifications

Custom sentence alignment algorithm

Filter aligned pairs using
- Sentence length
- Word Complexity
- Word Frequency
- SARI

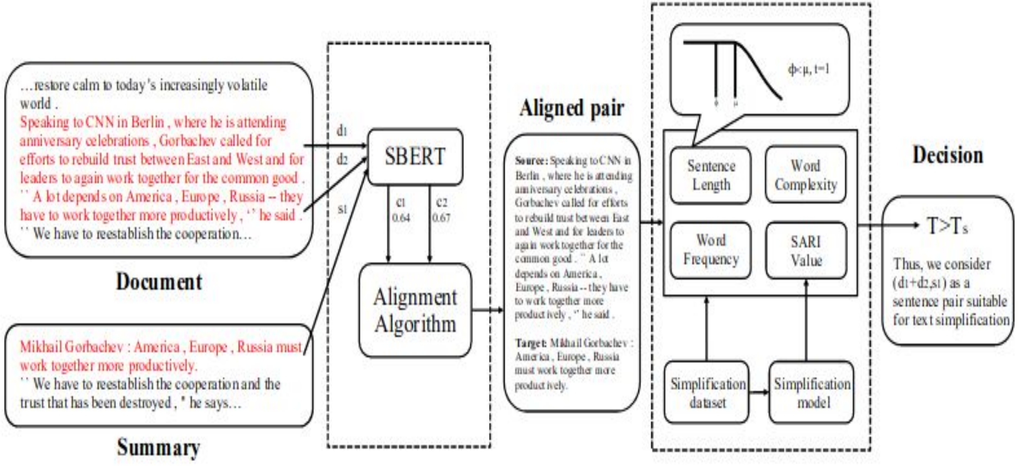


Figure 1: The process of mining suitable sentence pairs from summarization datasets.

Renliang Sun, Zhixian Yang, Xiaojun Wan, "Exploiting Summarization Data to Help Text Simplification", EACL 2023

Mining Summarisation Dataset for Simplifications

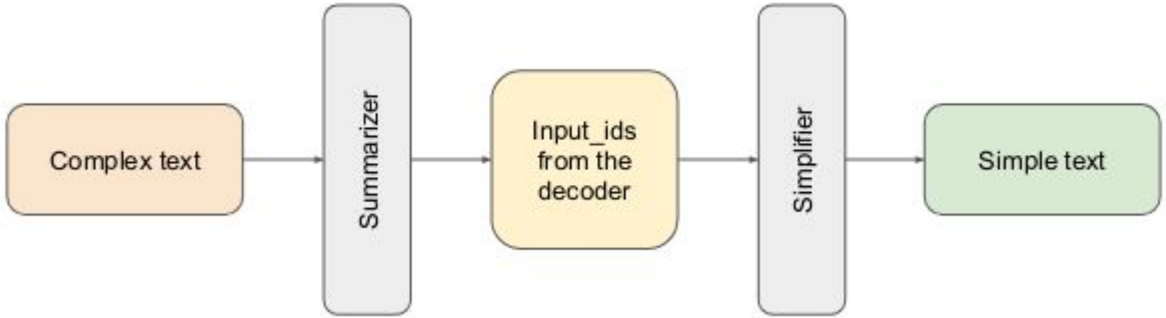
Models	WikiLarge				S4S				WikiLarge+OA				WikiLarge+S4S			
	SARI↑	F_{keep}	P_{delete}	F_{add}	SARI↑	F_{keep}	P_{delete}	F_{add}	SARI↑	F_{keep}	P_{delete}	F_{add}	SARI↑	F_{keep}	P_{delete}	F_{add}
Transformer	36.95*	70.80	36.91	3.15	34.43**	58.54	43.68	1.08	36.75*	70.79	36.38	3.06	**37.85**	71.11	39.15	3.27
BART	37.99**	72.53	37.85	3.59	36.21**	64.70	42.60	1.34	37.71**	73.02	36.81	3.31	**39.20**	70.99	42.31	4.30
ACCESS	39.67*	71.20	42.69	5.12	36.20**	65.62	41.53	1.44	39.46*	69.39	43.96	5.03	**40.71**	71.26	44.06	6.81

Models	WikiSmall				S4S				WikiSmall+OA				WikiSmall+S4S			
	SARI↑	F_{keep}	P_{delete}	F_{add}	SARI↑	F_{keep}	P_{delete}	F_{add}	SARI↑	F_{keep}	P_{delete}	F_{add}	SARI↑	F_{keep}	P_{delete}	F_{add}
Transformer	36.35*	66.69	40.53	1.82	36.75	60.23	49.49	0.53	36.38*	64.46	40.54	4.15	**38.57**	66.56	43.69	5.46
BART	35.13*	64.94	35.86	4.59	34.13*	61.06	39.95	1.39	34.65*	67.09	31.92	4.93	**36.58**	67.39	37.14	5.22
ACCESS	35.35*	65.01	38.50	2.53	34.63**	51.07	51.76	1.05	35.67*	60.95	44.29	1.77	**38.28**	58.45	53.64	2.73

Adding mined data improves simplification results

Renliang Sun, Zhixian Yang, Xiaojun Wan. "Exploiting Summarization Data to Help Text Simplification", EACL 2023

SimSum - Summarisation + Simplification



End-to-End summarisation + simplification

BART/T5 fine tuned on Wiki-Large

Keyword Prompting to encourage the model to focus on important keywords from the input text

Input text (original)
a goatee is a style of facial hair incorporating hair on one 's chin but not on one 's cheeks . the exact nature of the style has varied according to time and culture .
Input text with `kw_score` **as prompt**
one_0.06 varied_0.07 goatee_0.76 a goatee is a style of facial hair incorporating hair on one 's chin but not on one 's cheeks . the exact nature of the style has varied according to time and culture .
Input text with `kw_sep` **as prompt**
one varied goatee </s> a goatee is a style of facial hair incorporating hair on one 's chin but not on one 's cheeks . the exact nature of the style has varied according to time and culture .

Sofia Blinova, Xinyu Zhou, Martin Jaggi. "SIMSUM: Document-level Text Simplification via Simultaneous Summarization", ACL 2023

SimSum - Summarisation + Simplification

	D-Wikipedia		Wiki-Doc	
	Complex	*Simple*	*Complex*	*Simple*
Total sentences	546,744	349,561	258,303	55,885
Total words	17,740,142	703,550	5,927,616	906,988
Avg sents per article	5.20	3.33	14.81	3.20
Avg words per sent	32.45	20.24	22.95	16.23

Both datasets derived from existing datasets and post-processed to

- keep pairs where the simplified text is at most 5 words longer than the input
- improve input/output alignment

Sofia Blinova, Xinyu Zhou, Martin Jaggi. “SIMSUM: Document-level Text Simplification via Simultaneous Summarization”, ACL 2023
https://github.com/epfml/easy-summary/tree/main

SimSum - Summarisation + Simplification

	D-Wikipedia			Wiki-Doc		
model	SARI↑	D-SARI↑	FKGL↓	SARI↑	D-SARI↑	FKGL↓
T5	45.64	36.23	8.36	50.63	41.05	6.79
BART	47.05	38.13	8.14	49.55	40.95	7.93
$\text{BART}^{\dagger}_{CNN}$	44.52	36.01	8.32	49.39	40.98	7.70
BRIO	48.24	29.86	6.39	48.65	33.06	6.84
MUSS	39.45	26.43	12.72	35.99	27.94	10.91
SimSum(T5)♣	49.04	39.54	6.04	50.20	40.32	**6.75**
SimSum(BART)♣	48.33	37.11	6.48	**50.67**	41.42	7.55
$\text{SimSum(T5)}^{\ddagger}$	**49.44**	**39.77**	**6.04**	49.11	**41.53**	6.79

SimSum with keyword prompting yields the best results

MUSS (Martin et al., 2021) - Sota multilingual sentence simplification system.
BRIO (Liu et al., 2022) - BART-Large pre-trained model with top performance on various sequence-to-sequence tasks fine-tuned on simplification data

Sofia Blinova, Xinyu Zhou, Martin Jaggi. “SIMSUM: Document-level Text Simplification via Simultaneous Summarization”, ACL 2023

PEER -Plan, Edit, Explain, Repeat

A collaborative model mimicking human writing

Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Yinxiao Liu, Simon Tong, Jindong Chen and Lei Meng. RewriteLM: An Instruction-Tuned Large Language Model for Text Rewriting, AAAI 2024.

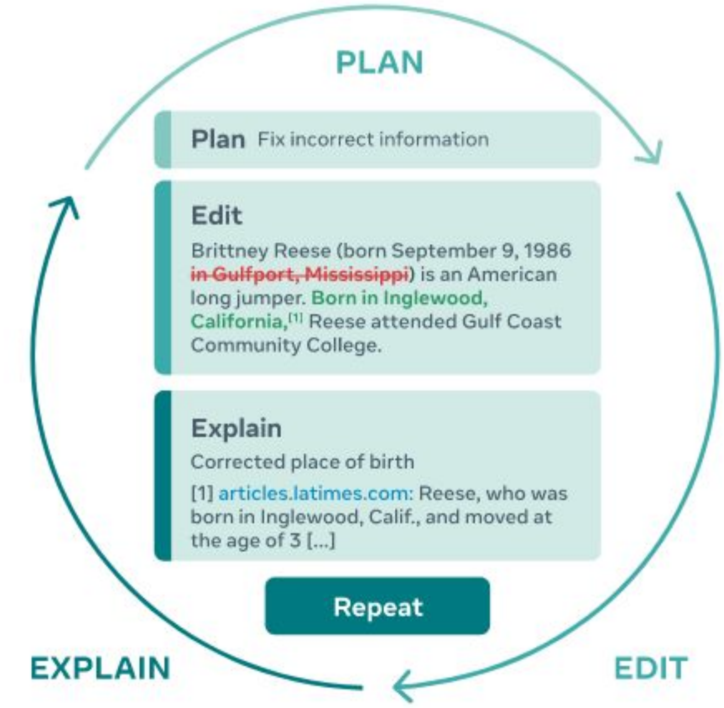
PEER - Collaborative Text Editing

PEER (Plan, Edit, Explain, Repeat)

A language model that can act as a writing assistant by following **plans** to perform a variety of different textual **edits**, ranging from syntactic and stylistic edits to changing the meaning of a text by removing, updating or adding information

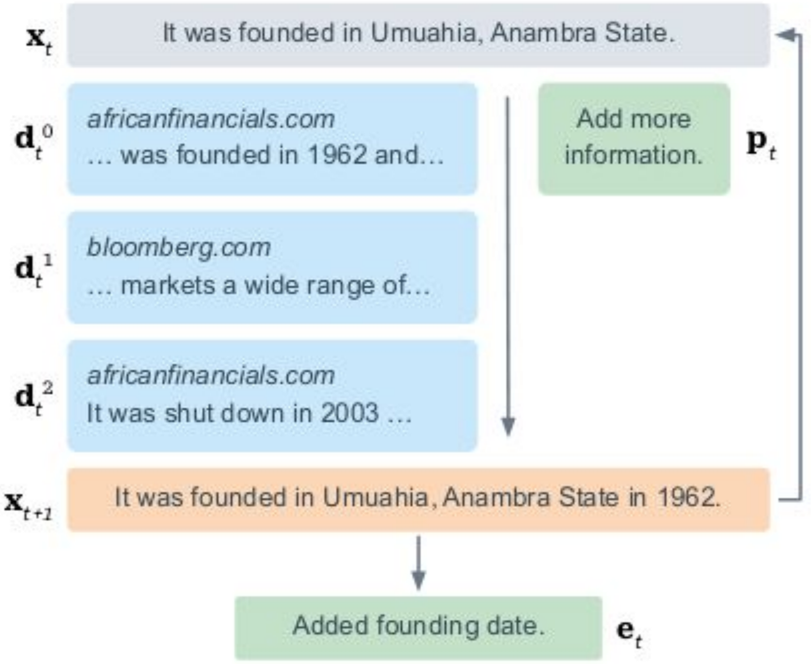
Models text writing as an **iterative** process, where we repeatedly plan and realize changes.

Supports **interactive** editing



Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave and Sebastian Riedel. PEER: A COLLABORATIVE LANGUAGE MODEL, ICLR 2023

PEER - Generation Process



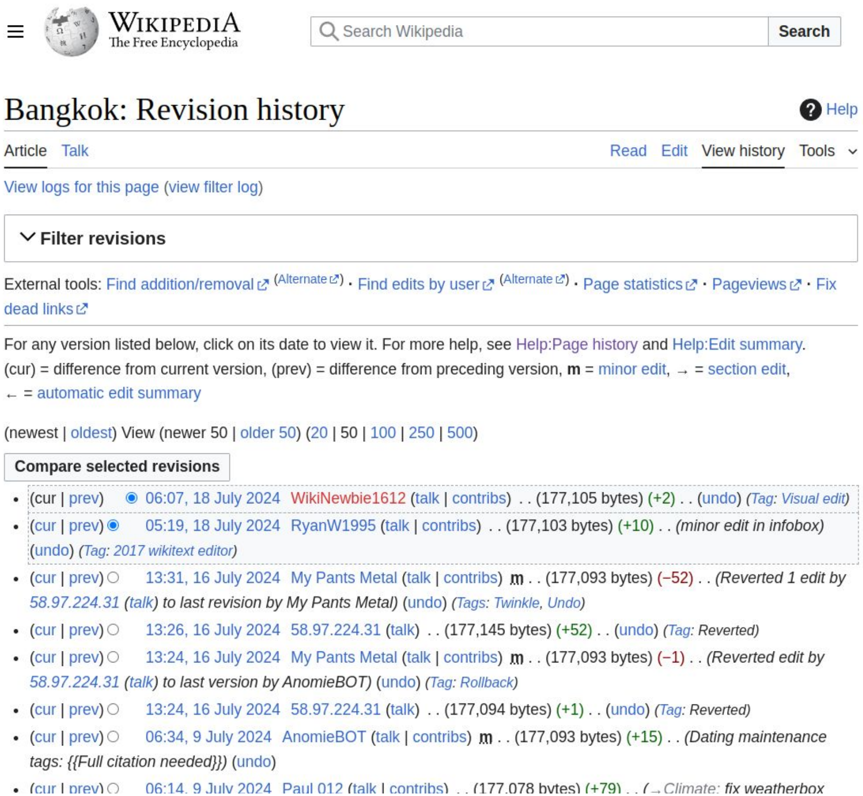
Given a text x_t and a collection of documents D_t^i, generate, realise and explain an edit plan.

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave and Sebastian Riedel. PEER: A COLLABORATIVE LANGUAGE MODEL, ICLR 2023

PEER - Training Data (x_t, x_{t+1}, d_1, d_2, d_3, p_t, e_t)

Wikipedia Revision History

- edit, comments and frequently contain citations, which is helpful for finding relevant documents.



CONS

- Writing style, plans, edits specific to Wikipedia
- Noisy comments, not always an appropriate proxy for plans or explanations.
- Often lack citations and so lack of background information

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave and Sebastian Riedel. PEER: A COLLABORATIVE LANGUAGE MODEL, ICLR 2023

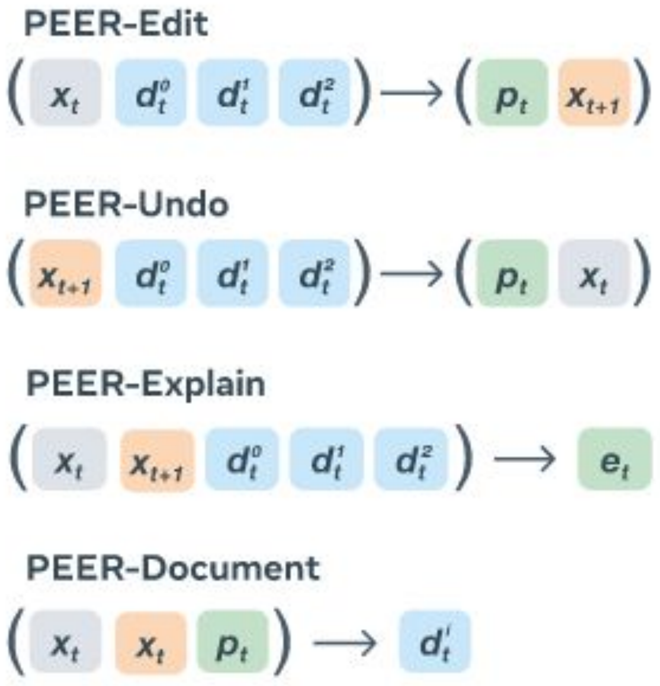
PEER - 4 models to infill various parts of the process

PEER-Edit: given an input text and a set of documents, plan and realise edits

PEER-Undo: given a text sequence and a set of documents, guess and undo the latest edit

PEER-Explain: given an edit and a set of documents, generates an explanation

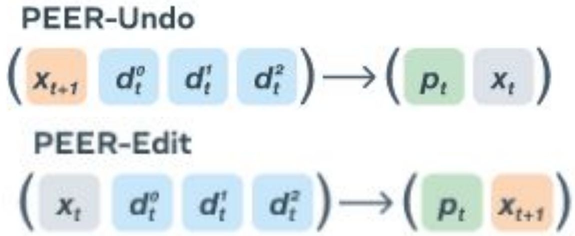
PEER-Document: given an edit., generate a document that provides useful background information



Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave and Sebastian Riedel. PEER: A COLLABORATIVE LANGUAGE MODEL, ICLR 2023.

PEER - Creating Synthetic Data

Training on text without edit history
- Use PEER-Undo for generating synthetic plans and edits
- Train PEER edit on the resulting data



Generating explanations
- Use PEER-Explain to select the most likely explanation (sample and select explanation that makes the edit most likely

Generating documents
- Use PEER-Document (sample and select documents that makes the edit most likely
- Only used during training (not inference)

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave and Sebastian Riedel. PEER: A COLLABORATIVE LANGUAGE MODEL, ICLR 2023

PEER Domain Adaptation - Can Peer rewrite text across different domains ?

Test on Natural Edits, a collection of naturally occuring edits for different text types and domain

- use PEER-Undo to create synthetic edits from plain texts

- Domain adapted PEER (PEER DA)
  - Finetune PEER-Edit on a balanced mixture of examples from the original training distribution and synthetic in-domain edits for 1,000 steps

PEER - Domain Adaptation

PEER can adapt to new domains
Synthetic plans improve results

	Wiki	News	Cooking	Garden	Law	Movies	Politics	Travel	Workpl.
Copy	0.0 / 32.7	0.1 / 32.8	0.0 / 31.6	0.0 / 32.0	0.0 / 31.1	0.0 / 31.5	0.0 / 31.8	0.0 / 31.2	0.0 / 31.5
PEER (no plans)	16.6 / 50.7	10.8 / 41.3	4.5 / 36.3	1.8 / 35.1	2.6 / 35.8	2.9 / 35.3	2.1 / 36.5	1.6 / 34.8	3.1 / 34.7
PEER	**26.2 / 55.5**	21.3 / 49.3	11.0 / 40.2	4.4 / 37.7	7.5 / 36.4	6.7 / 39.2	6.8 / 38.7	6.7 / 38.1	6.9 / 36.7
PEER (DA)	–	**23.3 / 51.6**	**13.2 / 42.9**	**8.1 / 44.9**	**9.4 / 39.0**	**9.9 / 42.4**	**11.6 / 41.3**	**9.1 / 40.2**	**8.3 / 39.2**

Table 3: EM-Diff / SARI scores on all subsets of Natural Edits. The domain-adapted (DA) variants of PEER clearly outperform regular PEER, demonstrating the usefulness of synthetic edits generated with PEER-Undo.

- Plans help
- PEER (DA) clearly outperform regular PEER for all subsets of Natural Edits
- This demonstrates the effectiveness of generating synthetic edits for applying PEER in different domains.

PEER - A generic model for multiple rewriting tasks

- JFLEG: Grammatical error correction
- ASSET: single-sentence simplification
- ITERATER: five edit intentions across three different domains
- WNC: remove or mitigate biased words to make sentences more neutral
- FRUIT: texts from Wikipedia that need to be updated based on a set of reference documents from Wikipedia are provided;
- WAFER-INS: insert a sentence in a Wikipedia paragraph given documents from the Sphere corpus that contain relevant background information.

		Without Documents				With Documents		
Model	**Params**	JFLEG	ASSET	ITER	WNC	FRUIT	WAFER	**Avg**
Copy	–	26.7 / 40.5	20.7	30.5	31.9 / 0.0	29.8 / 0.0	33.6	28.9
T*k*-Instruct	3B	31.7 / 38.7	28.3	36.2	30.3 / 0.0	12.7 / 3.9	1.6	23.5
T0	3B	42.9 / 38.6	28.6	28.1	17.8 / 0.0	13.1 / 5.7	6.1	22.8
T0++	11B	35.9 / 43.8	25.8	36.1	27.0 / 0.0	16.1 / 3.7	3.9	24.1
PEER	3B	54.8 / 55.1	29.9	36.5	56.4 / 31.9	39.4 / 28.3	35.2	42.0
PEER (SP)	3B	59.0 / 57.2	**33.2**	37.1	56.6 / 32.7	40.3 / **33.9**	35.5	43.6
PEER (SP)	11B	**59.9** / **58.6**	32.4	**37.8**	**58.8** / **34.7**	**40.7** / 33.5	**35.9**	**44.3**
OPT	175B	49.2 / 49.4	25.8	31.4	25.1 / 0.0	35.6 / 27.4	21.1	31.4
GPT3	175B	50.6 / 51.8	25.0	30.7	26.0 / 0.5	33.6 / 25.9	22.9	31.5
InstructGPT	175B	62.3 / 60.0	35.4	38.2	33.9 / 0.7	37.5 / 23.4	29.2	39.4
Sup. SotA	–	*– / 62.4*	*44.2*	*37.2*	*– / 45.8*	*– / 47.4*	–	–

PEER outperforms Tk-Instruct, T0, T0++ (variants of T5), GPT3 and OPT despite the fact that OPT and GPT3 are much larger models.

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave and Sebastian Riedel. PEER: A COLLABORATIVE LANGUAGE MODEL, ICLR 2023.

PEER - can generate text

400 intro sections from Wikipedia, each with three reference documents.

Wiki-LM: trained to predict a text given D_t and the page's title.

Autonomous mode
- the model writes and realizes its own plans;

Manual mode
- the model is given a series of human-written plans.

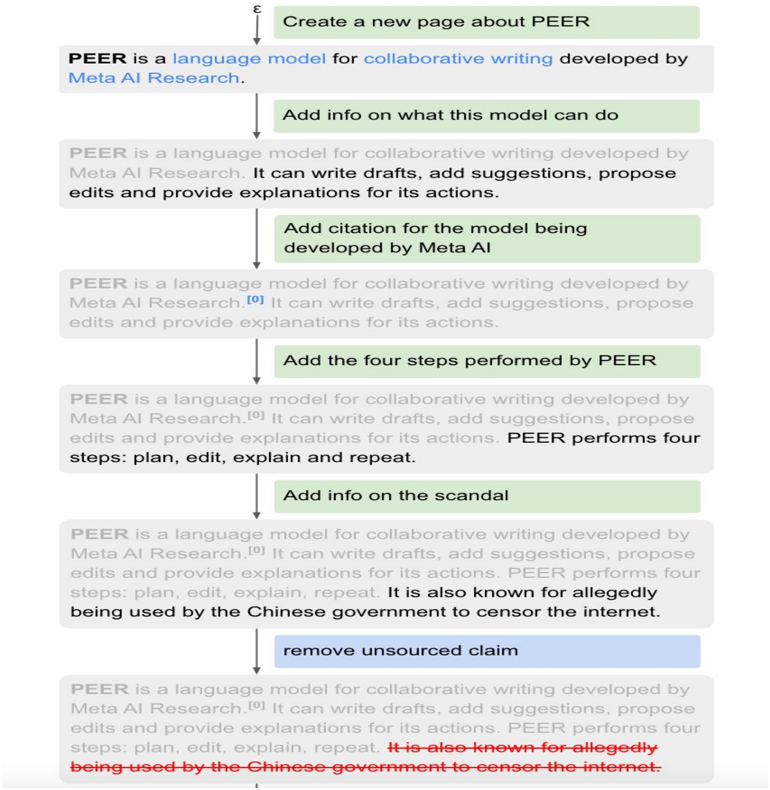
Collaborative mode
- human-written plans are interleaved with plans proposed by PEER

Model	LP	R1 / R2 / RL	QuestEval
Wiki-LM	5.0	38.4 / 16.9 / 27.3	38.7
PEER (autonomous)	5.0	37.7 / 15.8 / 26.2	40.6
PEER (manual)	2.0	39.4 / 17.0 / 28.1	**41.1**
PEER (collaborative)	2.0	**39.5 / 17.2 / 28.4**	41.0

All variants of PEER perform considerably better in terms of QuestEval scores than WikiLM, suggesting that iteratively updating text helps the model stay more **faithful** to the provided reference documents.

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave and Sebastian Riedel. PEER: A COLLABORATIVE LANGUAGE MODEL, ICLR 2023.

Collaborative Text Simplification



Human, Model

PEER can

- Add information and citations
- Incorrectly accepts an incorrect plan and follows it by hallucinating a scandal about internet censorship.
- correct the misinformation it has produced in a next step

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave and Sebastian Riedel. PEER: A COLLABORATIVE LANGUAGE MODEL, ICLR 2023

OpenRewriteEval and RewriteLM

Creating Test and Training data for a wide range of rewriting tasks
Training a rewriting model using Reinforcement Learning

Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Yinxiao Liu, Simon Tong, Jindong Chen and Lei Meng. RewriteLM: An Instruction-Tuned Large Language Model for Text Rewriting, AAAI 2024.

OpenRewriteEval: A Benchmark for Text Rewriting

A novel benchmark that covers a wide variety of rewriting instructions and is designed to facilitate the evaluation of open-ended rewriting of **long-form texts**

Source Text	The way in which we feed our children determines how healthy they are. It also determines how nicely they develop and develop. Children need to be fed a selection of foods each day in order to make sure their physical, emotional, and mental health. All foods are from 1 of the fundamental food teams. Milk and cheese arrive from the dairy team, for instance, and green beans arrive from the vegetable team. Bread arrives from the grain team and beef comes from the meat team. Chocolate arrives from the body fat and sugars team. Our children need so many servings for each day from each of these meals teams to preserve great health.
Instruction	Rewrite the text so that it is easy to understand.
Target Text	What we give our kids to eat affects how healthy they are and how they grow. Kids need to eat different types of food each day to stay physically, emotionally, and mentally healthy. All foods belong to one of the five food groups: dairy, vegetables, grains, meat and beans, and fruits. Milk and cheese are dairy foods, green beans are vegetables, bread is a grain, beef is a meat, and chocolate is a fat and sugar food. To stay healthy, kids need to eat a certain number of servings from each food group every day.
Instruction	Summarize the text.
Target Text	Feeding children a variety of foods from the five fundamental food groups (dairy, vegetables, grains, meat, and fats/sweets) is essential for their overall health and development, including their physical, emotional, and mental well-being.

Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Yinxiao Liu, Simon Tong, Jindong Chen and Lei Meng. RewriteLM: An Instruction-Tuned Large Language Model for Text Rewriting, AAAI 2024.

OpenRewriteEval: A Benchmark for Text Rewriting

Human annotators attach appropriate instructions to each source text and then rewrite them accordingly.

Dataset	Size	Data Source	Instruction Examples
$D_{\text{Formality}}$	200	See Appendix A.1	Too conversational, rephrase it to be more formal? Make the text more formal. Rephrase it to be more formal?
$D_{\text{Paraphrase}}$	102	See Appendix A.1	Paraphrase this. Reword this text. Use different wording.
D_{Shorten}	102	See Appendix A.1	Make wording more concise. Improve accuracy, clarity, and conciseness of language. Rephrase for clarity and conciseness.
$D_{\text{Elaborate}}$	102	See Appendix A.1	Elaborate on advantages of JavaScript. Add more details about fighting styles. Describe more about what the third page does.
$D_{\text{MixedWiki}}$	606	Wiki	Attempt to make the text sound less like an advertisement. Change to have a consistent past tense throughout the paragraph. Rewrite text in the present tense. Give a detailed and concise description of the Wollyleaf bush. Rewrite for clarity and encyclopedic tone.
$D_{\text{MixedOthers}}$	517	C4, Human	Make it more personal and friendly. Rewrite to haiku. Change the name to Horton Beach throughout the text. Make it more motivational for parents of age 50. Create bullet points from text.
All	1629		

Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Yinxiao Liu, Simon Tong, Jindong Chen and Lei Meng. RewriteLM: An Instruction-Tuned Large Language Model for Text Rewriting, AAAI 2024.

OpenRewriteEval - Training Data

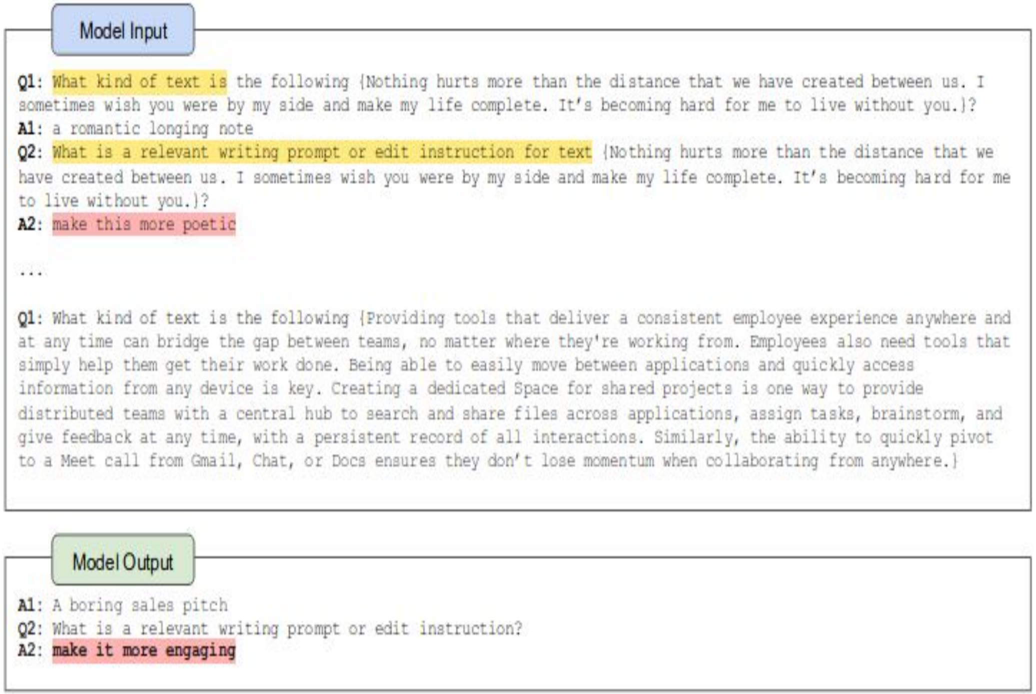
Wiki Instruction Dataset

- From Wikipedia revision history: Edit summary as a proxy for instruction (filtered and improved)

Synthetic Instruction Dataset

- Generated using CoT prompting
- Target text is generated from source and instruction

Check for hallucination using NLI and remove hallucinating instances



Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Yinxiao Liu, Simon Tong, Jindong Chen and Lei Meng. RewriteLM: An Instruction-Tuned Large Language Model for Text Rewriting, AAAI 2024.

RewriteLM - Model

1. Fine-tune Decoder M_{base} on Wiki- and Synthetic Instruction Dataset (M_{sft})
2. Train a reward model on synthetic data
    a. Generate target text using M_{base} and M_{sft}
    b. Rank the two outputs using NLI, edit distance and length ratio
    c. Transformer with a linear regression output layer
    d. Computes the predicted score difference $\sigma(r_{good} - r_{bad})$.
    e. The training loss is the entropy of the normalized score difference
3. Optimize M_{sft} using Reinforcement Learning (M_{rewite})

Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Yinxiao Liu, Simon Tong, Jindong Chen and Lei Meng. RewriteLM: An Instruction-Tuned Large Language Model for Text Rewriting, AAAI 2024.

RewriteLM - Results on OpenRewriteEval

		Edit Ratio	NLI (s-p)	NLI (p-s)	SARI	GLEU	Update-R
Pretrained LLMs							
PaLM (Chowdhery et al., 2022)	62B	0.31	0.25	0.11	28.24	0.74	11.99
PaLM 2 (Passos et al., 2023)	M	**1.22**	0.63	0.37	28.62	0.48	8.14
LLaMA (Touvron et al., 2023)	65B	0.71	0.83	0.83	27.98	2.10	21.35
Instruction-Tuned LLMs							
Alpaca (Taori et al., 2023)	13B	0.11	0.90	0.85	36.12	6.81	34.88
Vicuna (Chiang et al., 2023)	13B	0.23	0.89	0.77	39.05	6.84	33.31
Flan-PaLM (Chung et al., 2022)	62B	0.12	0.58	0.42	24.52	1.87	6.23
RewriteLMs							
Rewrite-PaLM	62B	0.14	0.88	0.76	37.02	7.40	36.68
Rewrite-PaLM 2	M	0.25	0.93	0.79	40.92	**9.64**	39.36
Rewrite-RL-PaLM 2	M	0.27	0.94	0.81	**40.97**	9.43	39.36
Rewrite-$RL_{r/w}$-PaLM 2	M	0.29	**0.96**	**0.87**	40.66	**9.64**	**40.10**

Edit ratio - proportion of the source text that is edited.
SARI - how close a generated text is to the source and target text
GLEU - BLEU customized to penalize only the changed n-grams in the target
Updated-R - the recall of n-grams between the model's prediction and the reference. Computes ROUGE-L on the updated sentences rather than the full text.

Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Yinxiao Liu, Simon Tong, Jindong Chen and Lei Meng. RewriteLM: An Instruction-Tuned Large Language Model for Text Rewriting, AAAI 2024.

RewriteLM - Results on OpenRewriteEval

- Pretrained LLMs have poor performance
- Instruction-tuned LLMs have better performance but still have room for improvement.
- RewriteLMs outperform
    - their corresponding foundation models
    - Instruction-tuned LLMs
- Applying reinforcement learning further improves its performance.

		Edit Ratio	NLI (s-p)	NLI (p-s)	SARI	GLEU	Update-R
Pretrained LLMs							
PaLM (Chowdhery et al., 2022)	62B	0.31	0.25	0.11	28.24	0.74	11.99
PaLM 2 (Passos et al., 2023)	M	**1.22**	0.63	0.37	28.62	0.48	8.14
LLaMA (Touvron et al., 2023)	65B	0.71	0.83	0.83	27.98	2.10	21.35
Instruction-Tuned LLMs							
Alpaca (Taori et al., 2023)	13B	0.11	0.90	0.85	36.12	6.81	34.88
Vicuna (Chiang et al., 2023)	13B	0.23	0.89	0.77	39.05	6.84	33.31
Flan-PaLM (Chung et al., 2022)	62B	0.12	0.58	0.42	24.52	1.87	6.23
RewriteLMs							
Rewrite-PaLM	62B	0.14	0.88	0.76	37.02	7.40	36.68
Rewrite-PaLM 2	M	0.25	0.93	0.79	40.92	**9.64**	39.36
Rewrite-RL-PaLM 2	M	0.27	0.94	0.81	**40.97**	9.43	39.36
Rewrite-$RL_{r/w}$-PaLM 2	M	0.29	**0.96**	**0.87**	40.66	**9.64**	**40.10**

Lei Shu, Liangchen Luo, Jayakumar Hoskere, Yun Zhu, Yinxiao Liu, Simon Tong, Jindong Chen and Lei Meng. RewriteLM: An Instruction-Tuned Large Language Model for Text Rewriting, AAAI 2024.