Automatic Text Summarization of COVID-19 Medical Research Articles using BERT and GPT-2

2020/12/29

김수영

Automatic Text Summarization of COVID-19 Medical Research Articles using BERT and GPT-2

Bowen Tan*
Laboratory of Molecular Genetics
Rockefeller University
New York, NY 10065
btan@rockefeller.edu

Virapat Kieuvongngam†
Laboratory of Membrane Biology and Biophysics
Rockefeller University
New York, NY 10065
vkieuvongn@rockefeller.edu

Yiming Niu‡
Laboratory of Molecular Neurobiology and Biophysics
Rockefeller University
New York, NY 10065
yniu@rockefeller.edu

Abstract

With the COVID-19 pandemic, there is a growing urgency for medical community to keep up with the accelerating growth in the new coronavirus-related literature. As a result, the COVID-19 Open Research Dataset Challenge has released a corpus of scholarly articles and is calling for machine learning approaches to help bridging the gap between the researchers and the rapidly growing publications. Here, we take advantage of the recent advances in pre-trained NLP models, BERT and OpenAI GPT-2, to solve this challenge by performing text summarization on this dataset. We evaluate the results using ROUGE scores and visual inspection. Our model provides abstractive and comprehensive information based on keywords extracted from the original articles. Our work can help the the medical community, by providing succinct summaries of articles for which the abstract are not already available.

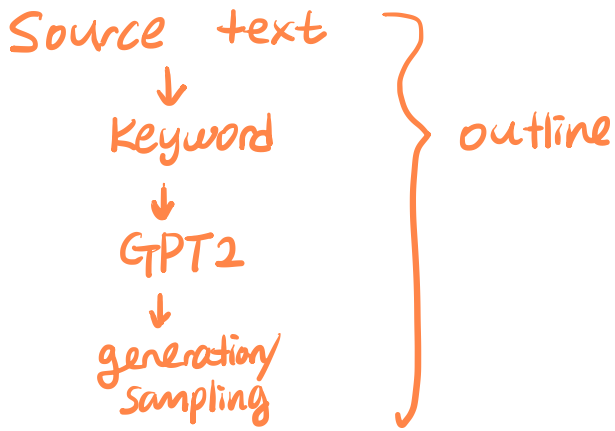
Low resource challenge

(D, S)

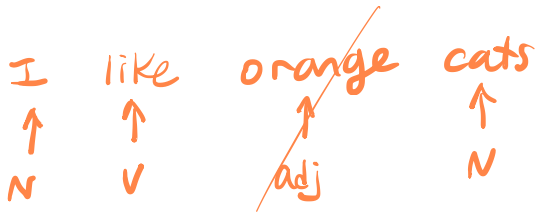
- CNN/Daily Mail dataset (286k)
- COVID-19 related literature, as of April 8 2020 (35k)

- The scientific terminology found in the literature can often be esoteric
- Not used in the mainstream text where the pre-training was performed

General Outline



- Abstractive Summary (novel model)
  - Extract a set of keywords from the source text using token classification tools
    - Use BERT that is fine-tuned for part of speech tagging
    - Extract the noun tokens and verb tokens from the sentences, and discard other words or parts of speech, such as adjectives, adverbs, determinants, etc.
  - Then use GPT2 that is specifically fine-tuned for making an abstractive summary from keywords



Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two*	TO	"to"	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential 'there'	*there*	VB	verb base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb gerund	*eating*
JJ	adjective	*yellow*	VBN	verb past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, sing.	*IBM*	$	dollar sign	*$*
NNPS	proper noun, plural	*Carolinas*	#	pound sign	*#*
PDT	predeterminer	*all, both*	"	left quote	' or "
POS	possessive ending	*'s*	"	right quote	' or "
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one's*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

Figure 9.1 Penn Treebank part-of-speech tags (including punctuation).

```
list_to_pick =
['NN','NNP','NNPS','NNS','VBD','VB','VBZ','VBP']
```

full text → BERT POS Keywords → Summarize

Source text

[" 'Two months after it was firstly reported, the novel coronavirus", 'disease COVID-19 has already spread worldwide. However, the vast', 'majority of reported infections have occurred in China. To assess the', 'effect of early travel restrictions adopted by the health authorities', 'in China, we have implemented an epidemic metapopulation model that is', 'fed with mobility data corresponding to 2019 and 2020. This allows to', 'compare two radically different scenarios, one with no travel', 'restrictions and another in which mobility is reduced by a travel ban.', 'Our findings indicate that i) travel restrictions are an effective', 'measure in the short term, however, ii) they are ineffective when it', 'comes to completely eliminate the disease.']

Keywords

['months was coronavirus disease has majority infections have china', 'assess effect travel restrictions health authorities china have', 'epidemic metapopulation model is mobility data allows compare', 'scenarios travel restrictions mobility is travel ban findings indicate', 'travel restrictions are measure term they are comes eliminate disease']

Input: <BOS> + $K_1, \ldots K_N$ + <Summarize> + abstract (ground truth) + <EOS> + <PAD>

In [ ]:
```
title = 'A data-driven assessment of early travel restrictions related to the spreading of the novel COVID-19 w
ithin mainland China '
```

In [ ]:
```
#@title GPT2 input preparation

GPT2_input = tokenizer_GPT2.encode(
      '<|startoftext|> ' +title + list_keywords_str + ' <|summarize|> ')
GPT2_input_torch = torch.tensor(GPT2_input, dtype=torch.long)

print("the keyword input :")
wrapper.wrap(tokenizer_GPT2.decode(GPT2_input_torch))
```

```
the keyword input :
```

Out[ ]:
```
['<|startoftext|>  A data-driven assessment of early travel restrictions',
 'related to the spreading of the novel COVID-19 within mainland China',
 'months was coronavirus disease has majority infections have china',
 'assess effect travel restrictions health authorities china have',
 'epidemic metapopulation model is mobility data allows compare',
 'scenarios travel restrictions mobility is travel ban findings indicate',
 'travel restrictions are measure term they are comes eliminate disease',
 '<|summarize|>']
```

, <EOS>

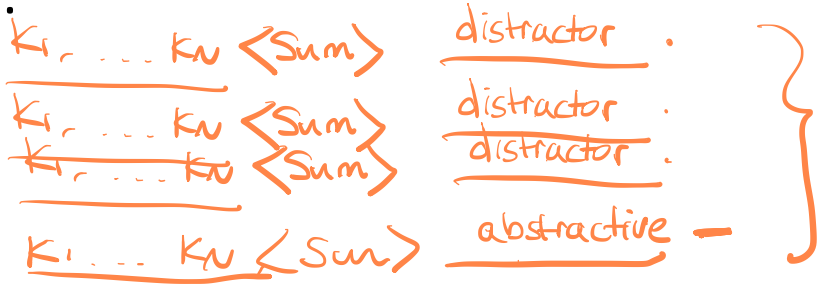
Abstractive Summary (cont'd)

loss = w_1 lm + w_2 mc
↑ 2 ↑ 1

- We use GPT2DoubleHead model, meaning, it has 2 heads: one head for causal language modeling (lm), and the other for multiple choice (mc) answering. So for training we have 2 losses to optimize, one is lm loss and the other one is mc loss. The total loss is a weighted sum of the two losses at ratio of 2:1 lm loss to mc loss.

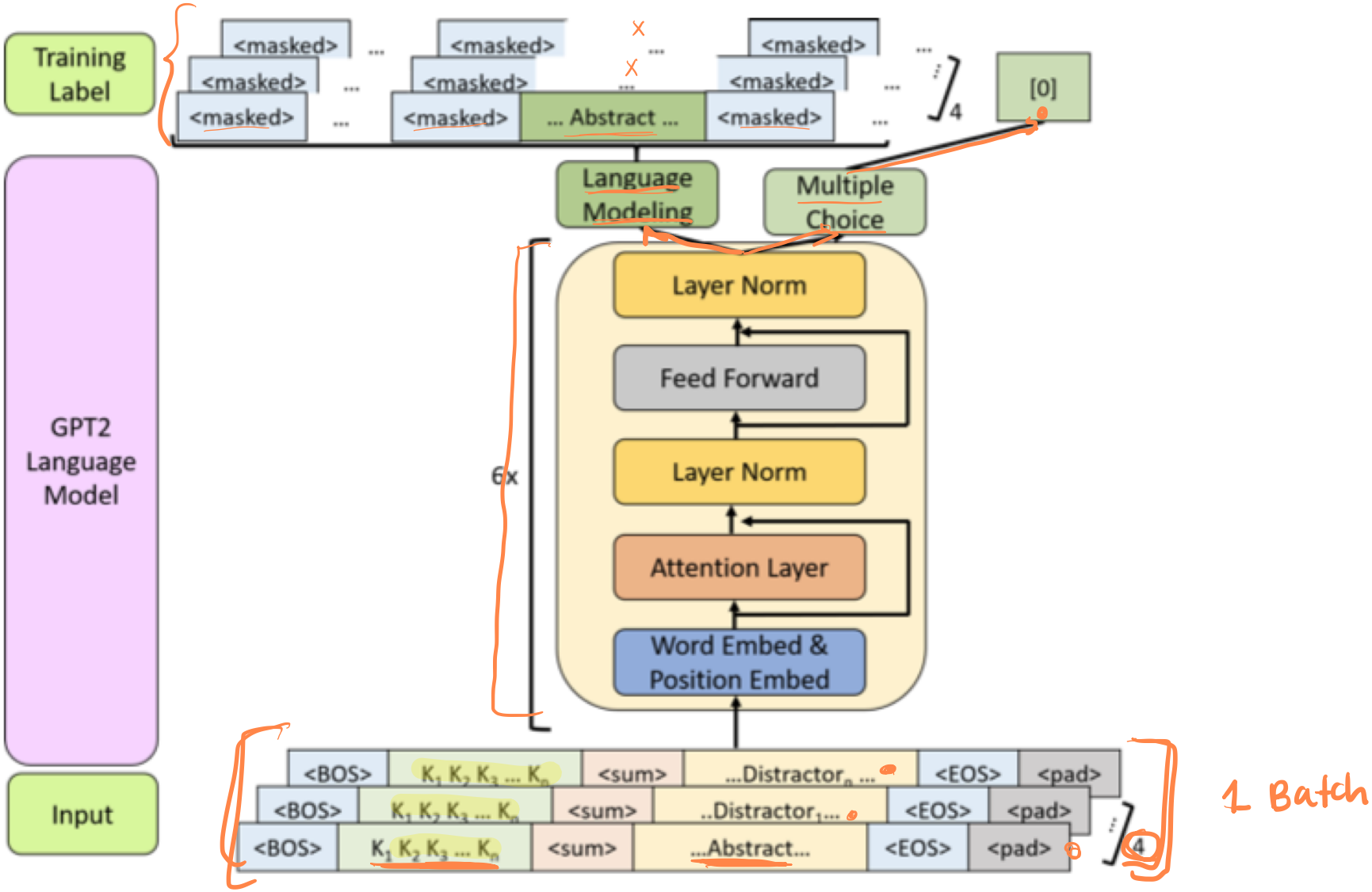
- For the lm task, the model predicts a next word token given previous tokens and context.
- For the mc task, given a set of keywords, the model choose the correct gold summary from summary choices.

- The rationale is that by optimizing two losses, we will force the model to learn both local context used to generate a next token and global semantic meaning for answering multiple choice question.

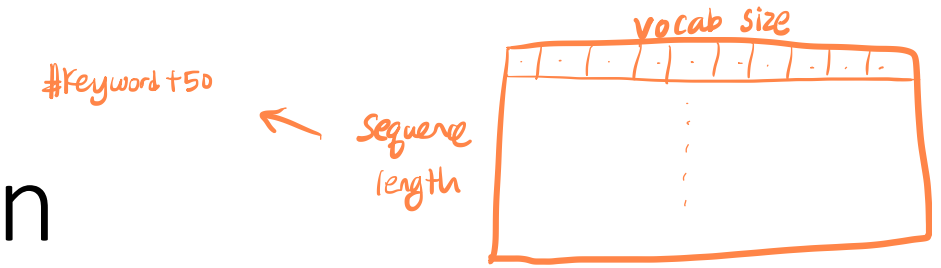


GPT2 multi-loss training

I like <mask> drinking water.



Sentence Generation



- The language modeling output of the GPT2 is a tensor of size [sequence length, vocab size]. This is a tensor of likelihood distribution over all words before the softmax.
- Currently, the two most promising candidates to succeed beam-search/greedy decoding are ***top-k*** and ***top-p sampling***. The general principle of these two methods is to sample from the next-token distribution after having filtered this distribution to keep only the top k tokens (*top-k*) or the top tokens with a cumulative probability just above a threshold (*nucleus/top-p*).

Sentence Generation (cont'd)

- We sample words from this distribution in the word-by-word manner.
  - To obtain the ith word, we consider the conditional probability of previous i – 1 words.

$$P(x) = \Pi_{i=1} P(X_i | x_i, ..., x_{i-1})$$

Sentence Generation (cont'd)

- Firstly, before the sampling, we can apply a scaling factor, so-called temperature (t), to the likelihood (u) to reshape the skews likelihood distribution before the softmax
- Temperature is a scaling factor (always positive real number) apply to the likelihood before softmax. higher temperature (>1) shrink all the likelihood together, making high likelihood and low likelihood closer; this result in word sequence appear more random (and sounds creative) than before. Low temperature sharpening the distribution, increasing the high likelihood and decreasing the likelihood of low probability word, making the result more deterministic.

$$p(x) = \frac{exp(u/t)}{\Sigma_{l*} exp(u_l^*/t)}$$

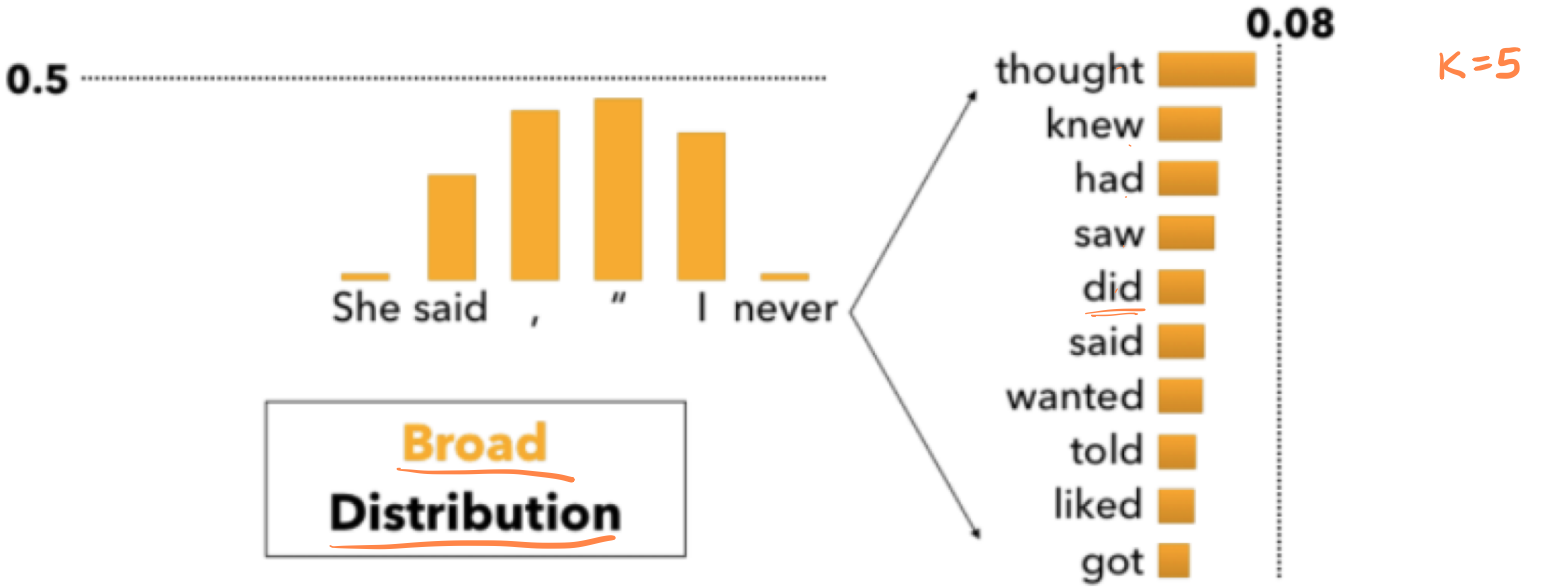
Sentence Generation (cont'd)

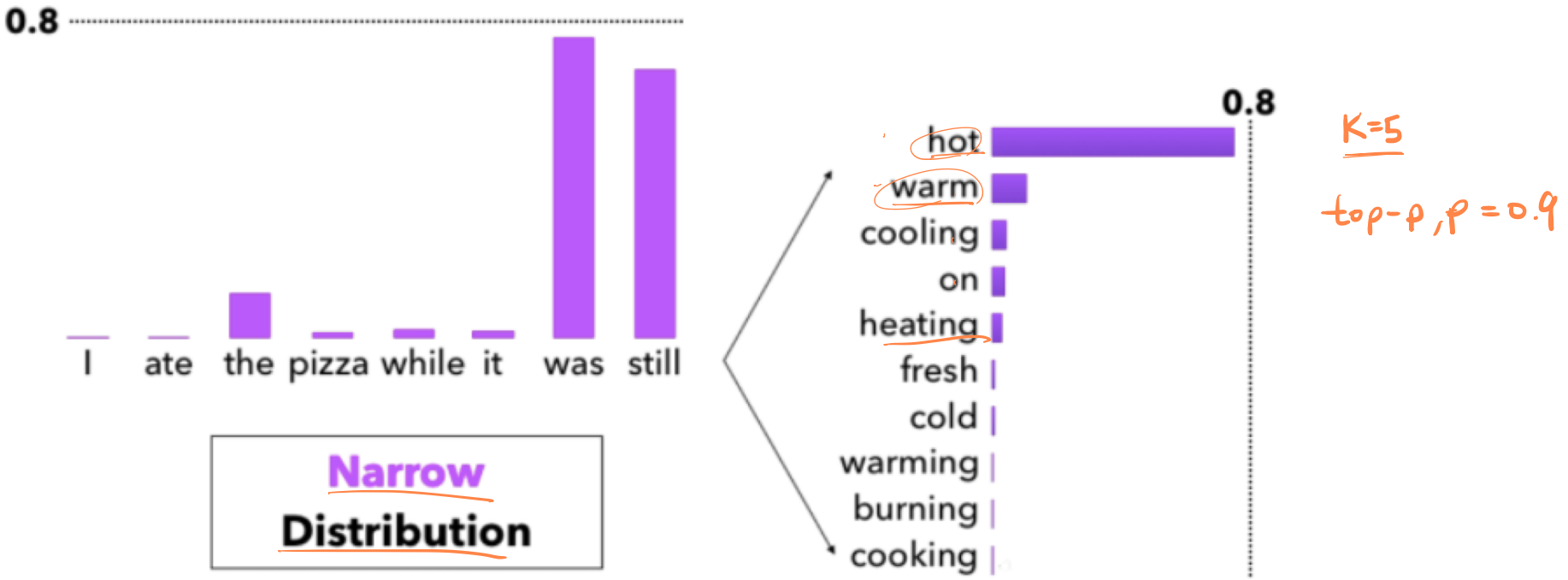
- top-k = the top k word candidates to consider when doing the sampling. For example, top-k = 5 will consider top 5 words when doing the sampling. Higher top-k value means considering many low probability words, thus, there is some chance that these words will be used.
- top-p = top p sampling choose top n word candidates such that the set of these n words have cumulative probability > p.

$$\Sigma_x P(x|x_{1:i-1}) >= p$$

- using both top-k and top-p sampling together allows us to cap n to be <= k.
- the probability mass is redistributed among only those filtered words

- We empirically tested a few of the sampling parameters and found that temperature = 1, k = 50, and p = 0.8 yields a reasonable generations.


0.5
She said , " I never
Broad
Distribution
0.08
thought
knew
had
saw
did
said
wanted
told
liked
got
K=5


0.8
I ate the pizza while it was still
Narrow
Distribution
0.8
hot
warm
cooling
on
heating
fresh
cold
warming
burning
cooking
K=5
top-p, p=0.9

Experiments and Results

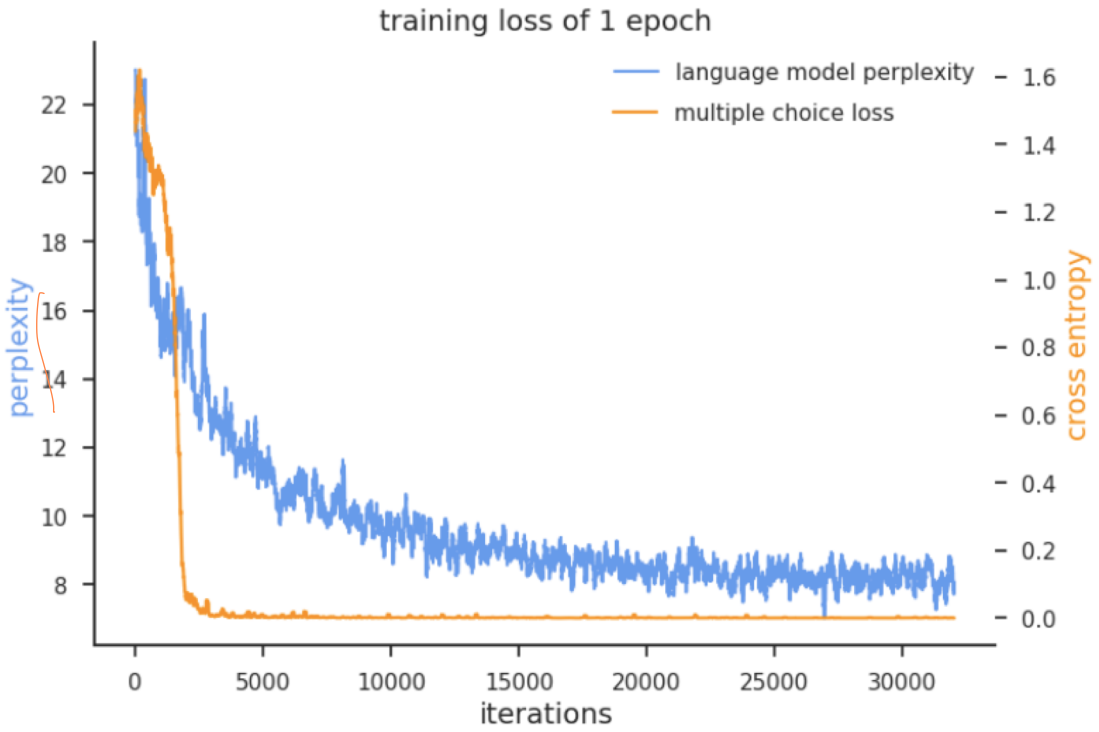


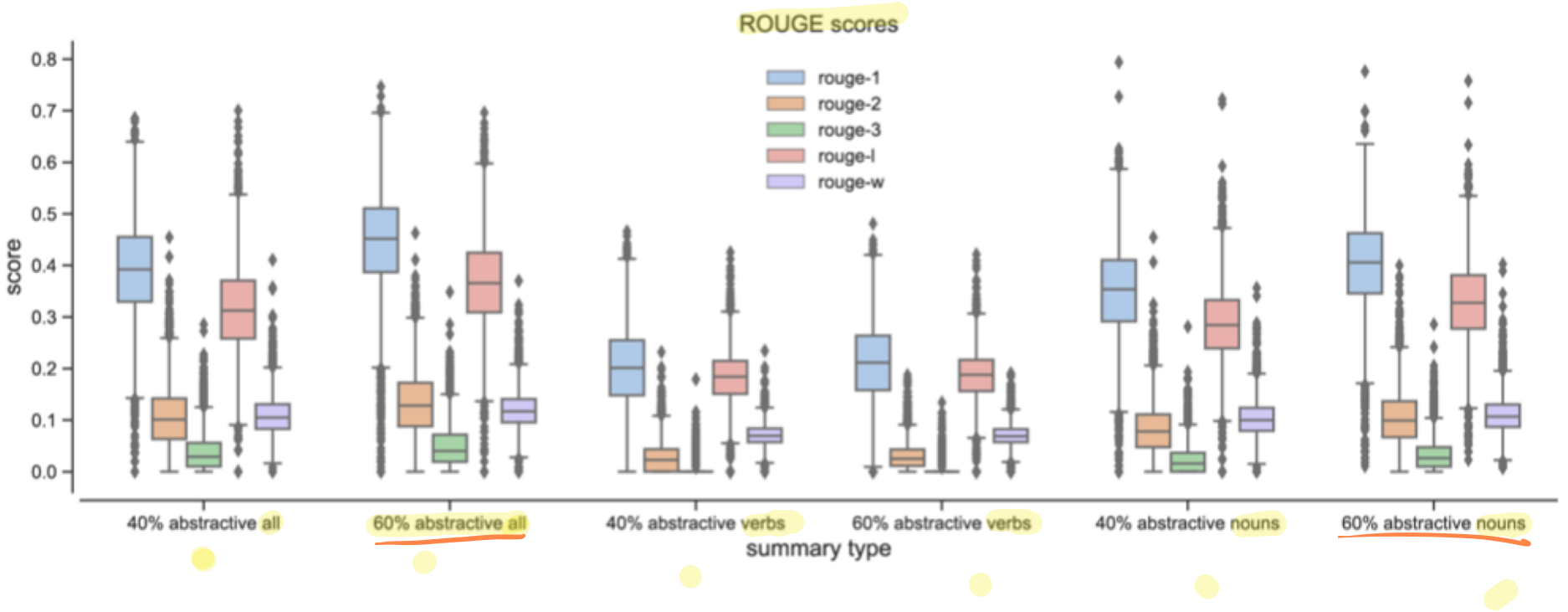
Figure 2: **Training results.** The two losses are shown during 1 epoch of training (32k iterations). The language model loss (blue) is shown in the exponentiated form of the cross-entropy loss, so-called the perplexity score (e^{lmloss}). The multiple choice loss (orange) is calculated from the cross entropy loss over all the multiple choices.

Table 1: Example of summary result

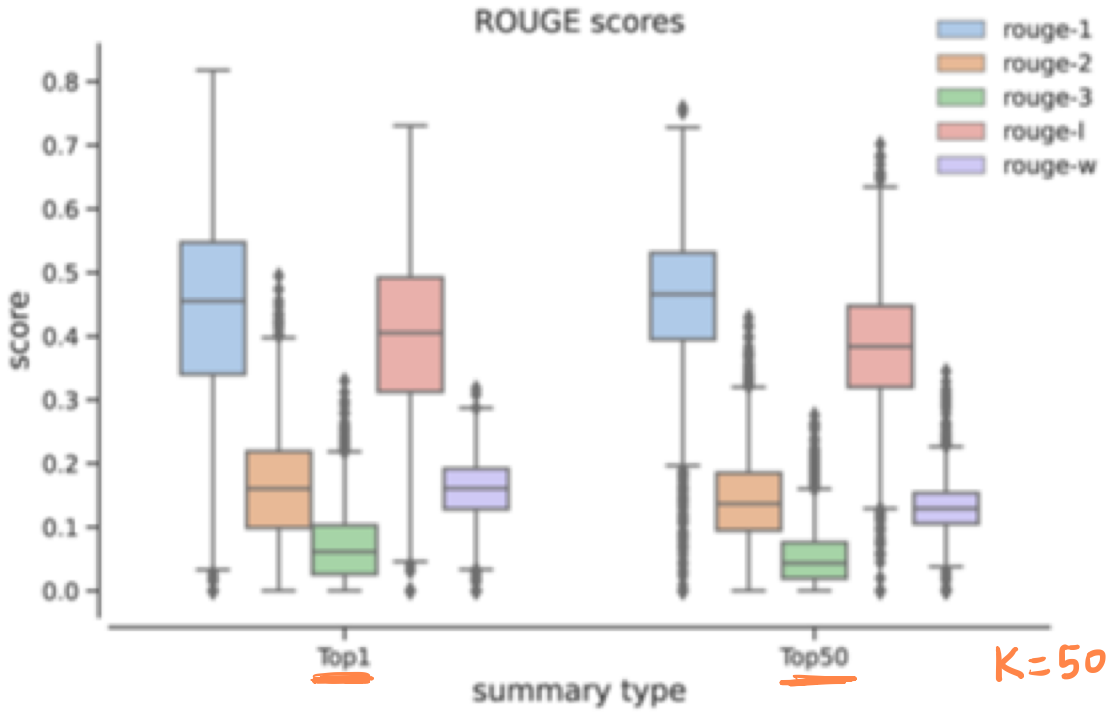
gold summary	keyword input	generated abstractive summary
	1 epoch training	
Influenza virus is the most frequently reported viral cause of rhabdomyolysis. A 7-year-old child is presented with rhabdomyolysis associated with parainfluenza type 2 virus. Nine cases of rhabdomyolysis associated with parainfluenza virus have been reported. Complications may include electrolyte disturbances, acute renal failure, and compartment syndrome.	<|startoftext|> Rhabdomyolysis Associated with Parainfluenza VirusInfluenza virus is cause rhabdomyolysis child is rhabdomyolysis parainfluenza type 2 virus cases rhabdomyolysis parainfluenza virus have Complications include electrolyte disturbances renal failure compartment syndrome <|summarize|>	Influenza virus is the most common cause of respiratory rhabdomyolysis in the child. It is believed that the rhabdomyolysis and parainfluenza type 2 virus cases with rhabdomyolysis in parainfluenza type 2 virus. Recent cases with rhabdomyolysis in parainfluenza virus have been described. Complications include electrolyte disturbances, kidney failure,and gastrointestinal compartment syndrome.
	5 epoch training	
	<|startoftext|> Rhabdomyolysis Associated with Parainfluenza', 'VirusInfluenza virus is cause rhabdomyolysis child is rhabdomyolysis', 'parainfluenza type 2 virus cases rhabdomyolysis parainfluenza virus', 'have Complications include electrolyte disturbances renal failure', 'compartment syndrome <|summarize|>	Influenza virus is a leading cause of rhabdomyolysis in child. However, several cases of rhabdomyolysis in the parainfluenza virus have been reported. Complications include electrolyte disturbances in renal failure of the normal renal compartment syndrome

[Full text , abstract]

Analysis: Compression rate, verb/noun



Analysis: Sampling method



"the readability and abstractive meanings are significantly worse in the greedy search group compared to the top-k group."