https://arxiv.org/pdf/2206.04624.pdf

Paper review

Factuality Enhanced Language Models for Open-Ended Text Generation (NeurIPS 2022)

Presentation: **Jeiyoon Park**
6th Generation, TAVE

12/27/2022

TAVE Research

Outline

1. Background
2. Factuality Prompts
3. Evaluation Metrics
4. Factuality Analysis of Pretrained LMs
5. Factual-Nucleus Sampling
6. Factuality-Enhanced Continued Training
7. Discussion

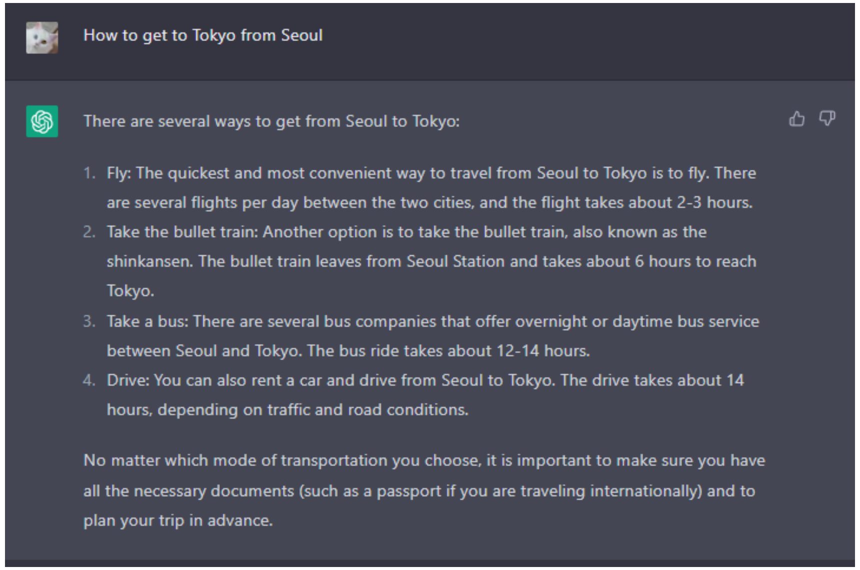
Outline

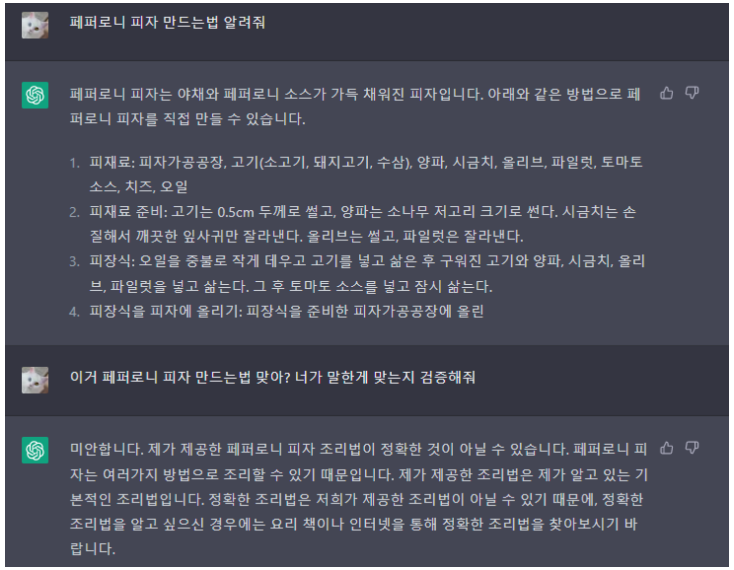
1. Background
2. Factuality Prompts
3. Evaluation Metrics
4. Factuality Analysis of Pretrained LMs
5. Factual-Nucleus Sampling
6. Factuality-Enhanced Continued Training
7. Discussion

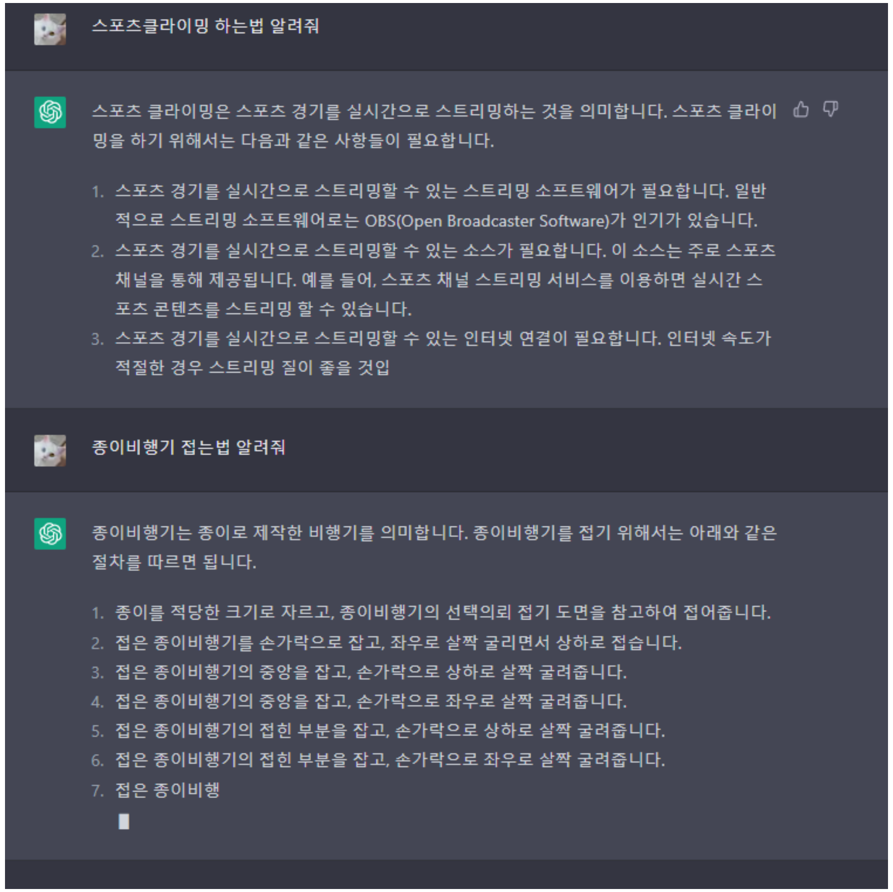
Why This Paper?

⚠ Are sentences generated by the LMs always true?

- e.g., ChatGPT (OpenAI; Nov 30, 2022)







Why This Paper?

⚠ Are sentences generated by the LMs always true?

- e.g., An example of continuations
- Red represents nonfactual, green stands for factual, and ~~strikethrough~~ is repetition

[Factual Prompt] One of lung cancer's symptoms is shortness of breath.

[Greedy] The American Cancer Society estimates that more than 228,000 people will be diagnosed with lung cancer in the United States this year.\n\n ~~The American Cancer Society estimates that more than 228,000 people will be diagnosed with lung cancer in the United States this year.\n\nThe American Cancer Society estimates that more than 228,000 people will be diagnosed with lung cancer in the United...~~

[p=0.9] That would make an oxygen mask one of the more popular treatments for this devastating disease. It helps ease breathing and give patients back their strength. Learn more here.<|endoftext|>"

[Ours] Other symptoms include coughing (including coughing up blood), chest pain, hoarseness, weight loss, and loss of appetite. The symptoms of lung cancer may resemble those of a bad cold or pneumonia. A chest x - ray is necessary to confirm the diagnosis. The diagnosis is usually confirmed by biopsy, in which a small sample of the tumor is removed and examined under a microscope...

[Non-Factual Prompt] Cancer can be prevented by drinking too much alcohol.

[Greedy] \nCancer can be prevented by eating too much red meat.\n\n~~Cancer can be prevented by eating too much~~ sugar.~~\n\nCancer can be prevented by eating too much~~ fat.~~\n\nCancer can be prevented by eating too much~~ processed food.~~\n\nCancer can be~~...

[p=0.9] More than 90% of hepatocellular carcinoma is associated with alcohol use.\n\nAbout 9 percent of cancers can be prevented if we don't eat more than 1.5 ounces of processed meat each day.\n\nDiet is believed to play a role in 7% of cancers and...

[Ours] Cancer is often treated with some combination of radiation therapy, surgery, chemotherapy and targeted therapy. Pain and symptom management are an important part of care. Palliative care is particularly important in people with advanced disease. The chance of survival depends on the type of cancer and extent of disease at the...

Background

1. The Four categories of knowledge

1) Factual Knowledge

- The basic elements that students must know to be
- e.g., The earth is round, Water boils at 100 degrees (STP)

2) Conceptual Knowledge

- The interrelationships among the basic elements within a large structure that enable them to function together.
- e.g., Water is a scarce resource.
- e.g., Water is significant to people for economic, spiritual and health reasons.
- e.g., TruthfulQA (Lin et al., ACL 2022)

Background

1. The Four categories of knowledge

3) Procedural Knowledge

- How to do something; methods of inquiry, and criteria for using skills, algorithms, techniques, and methods
- e.g., Sewage treatment process, Riding a bicycle.

4) Metacognitive Knowledge

- Knowledge of cognition in general as well as awareness and knowledge of one's own cognition.

Background

2. Factuality in NLP

- Factuality refers to being coherent to provide ground-truth knowledge sources in NLP.
- In this study, the scope of ground-truth knowledge is confined to Wikipedia for simplifying the evaluation setup

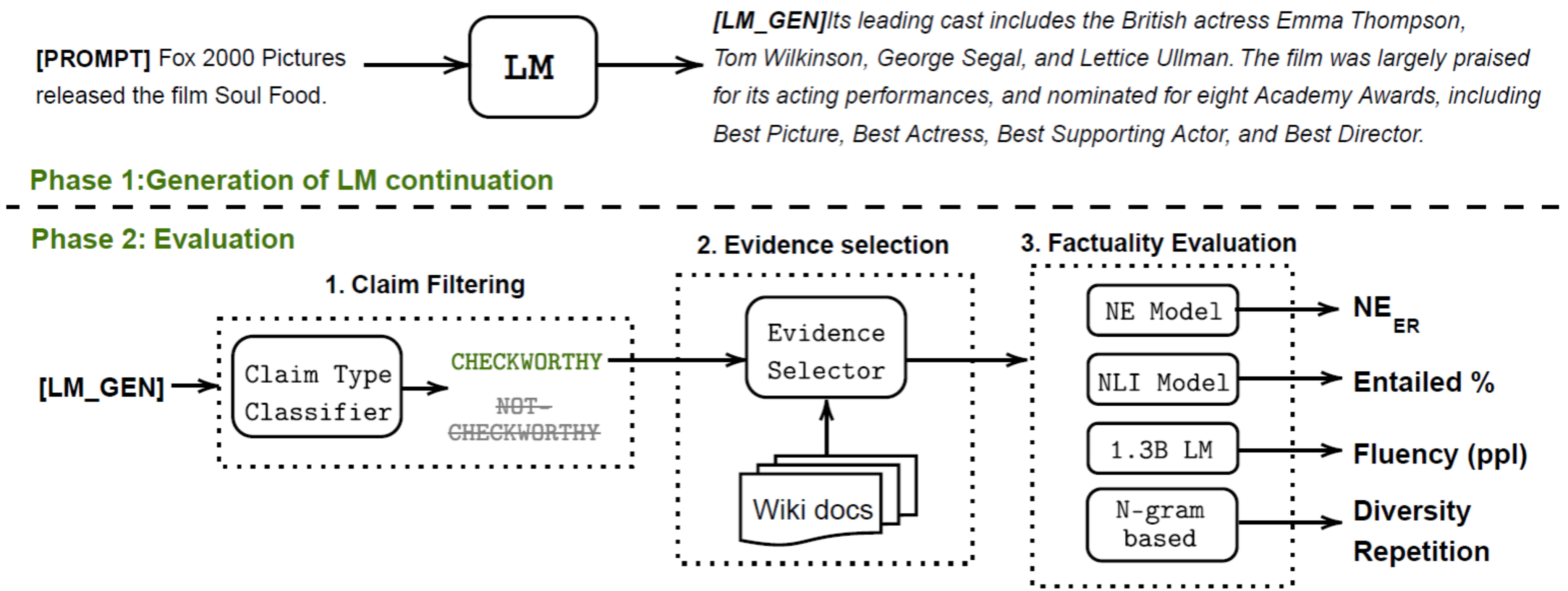
Outline

1. Background
2. Factuality Prompts
3. Evaluation Metrics
4. Factuality Analysis of Pretrained LMs
5. Factual-Nucleus Sampling
6. Factuality-Enhanced Continued Training
7. Discussion

Factuality Prompts

1. Evaluation Framework

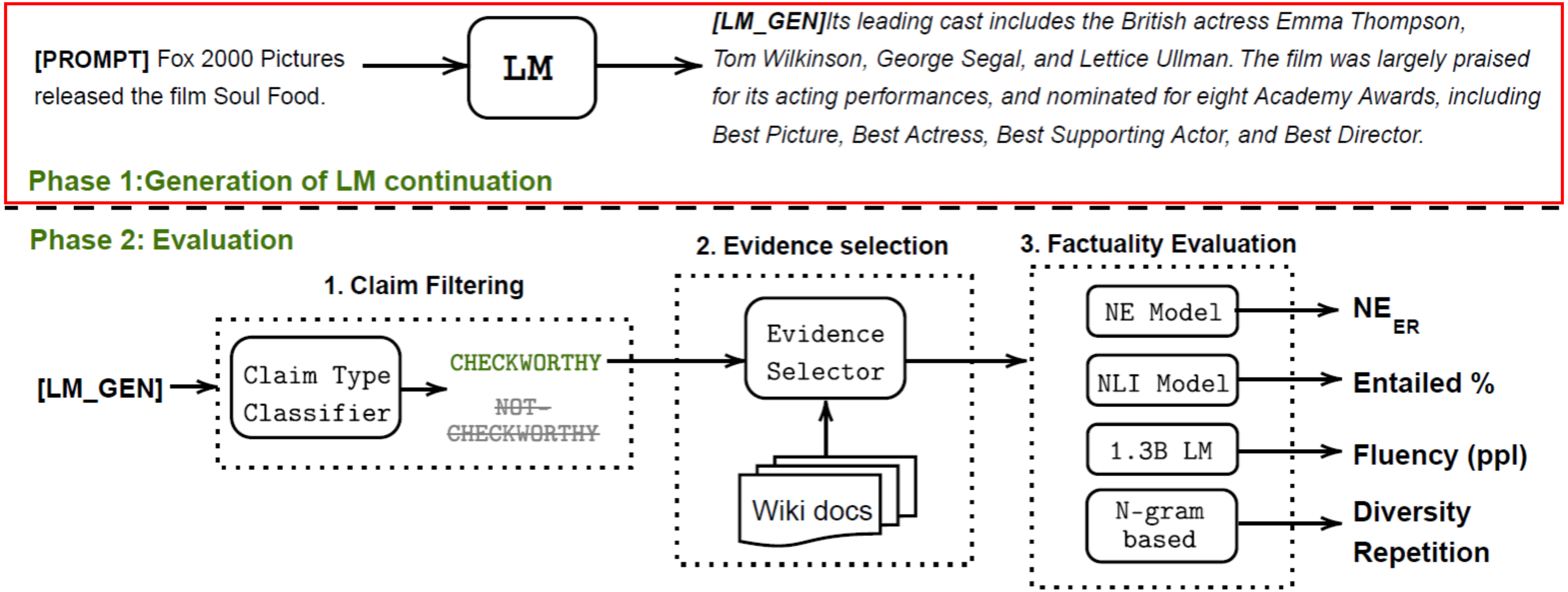
- It consists of two phases



Factuality Prompts

1. Evaluation Framework

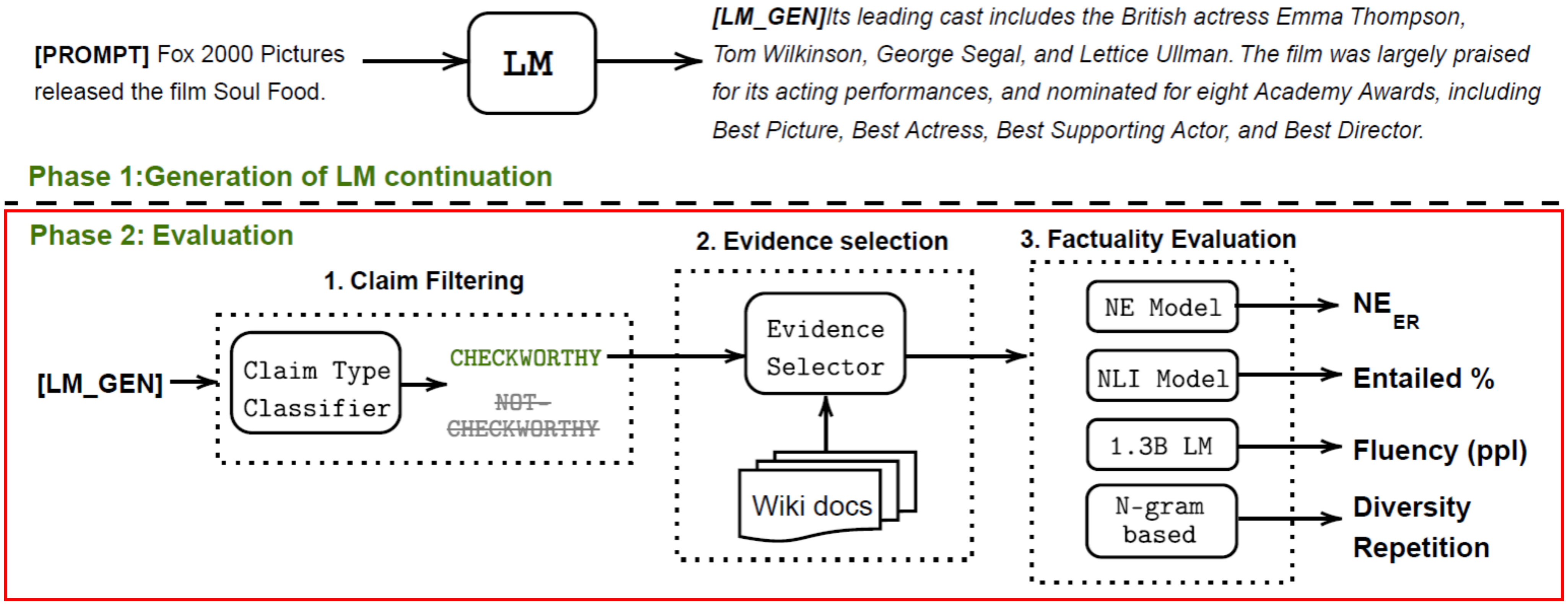
- It consists of two phases
- In Phase 1, LM creates the continuations from the provided test prompts



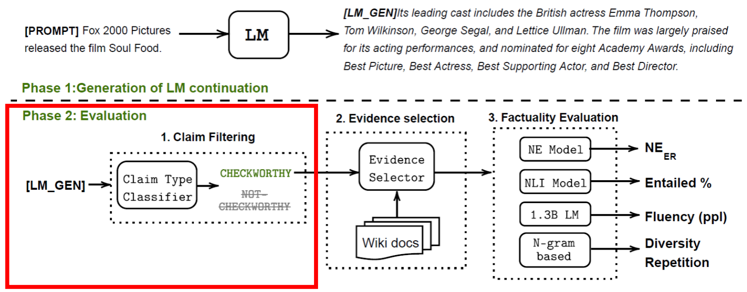
Factuality Prompts

1. Evaluation Framework

- In Phase 2, (1) Identify check-worthy continuations, (2) Prepare relevant ground-truth knowledge required for factual verification, (3) Factuality evaluation



Factuality Prompts



1. Evaluation Framework

- Claim Filtering; Identify check-worthy continuations

- It filters out "non-checkworthy" sentences that contain any of the following characteristics:

- Contains no named entities, which are important building blocks of fact or information. E.g., "Check this out", "To say that a person is an example of something is absurd."
- Contains first-person pronouns (i.e., I, we, and us), which are strong signal for personal opinions or casual chitchat style of writing. E.g., "I think...", "I believe..."
- Contains question mark. E.g., "Do you want to hear something interesting?", "Did you know?", "What are your thoughts?"

Factuality Prompts

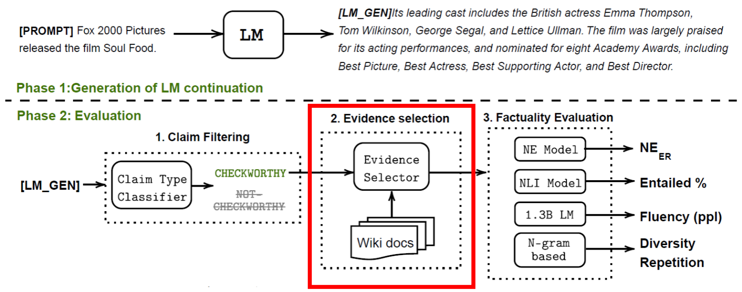
2. Factuality Prompts Testset

- A similar setup as REALTOXICITYPROMPTS (Gehman et al., EMNLP 2020)
- FACTUALITYPROMPTS is composed of *factual* and *nonfactual* prompts
- It exploits FEVER (Thorne et al., 2018 NAACL) dataset to construct both prompts
- FEVER is a fact-checking dataset consisting of claims that are SUPPORTED, REFUSED, and NOTENOUGHINFO by Wikipedia documents
- FACTUALITYPROMPTS leverages the SUPPORTED and REFUSED claims from FEVER validation set as *factual* and *nonfactual* prompts

Table 7: Data statistics of FACTUALITYPROMPTS

	Factual Prompts	Nonfactual Prompts
# Prompts	8000	8000
Avg # Tokens	9.77	9.48

Factuality Prompts

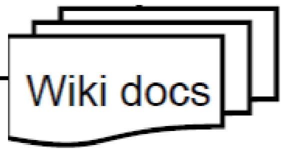


3. Ground-Truth Knowledge Preparation

- The required ground-truth knowledge can be either document-level or sentence-level, depending on the type of factuality metrics
- For document-level, It directly employs Wikipedia document annotation from FEVER
- For sentence-level, It automatically selects sentence by using two different methods (TF-IDF or Contextual representation (i.e., Sentence Transformer))
- Cosine similarity

[LM_GEN]*Its leading cast includes the British actress Emma Thompson, Tom Wilkinson, George Segal, and Lettice Ullman. The film was largely praised for its acting performances, and nominated for eight Academy Awards, including Best Picture, Best Actress, Best Supporting Actor, and Best Director.*

$\longrightarrow q \xleftrightarrow{\text{Cosine Similarity}} C = \{c_1, \dots, c_N\} \longleftarrow$



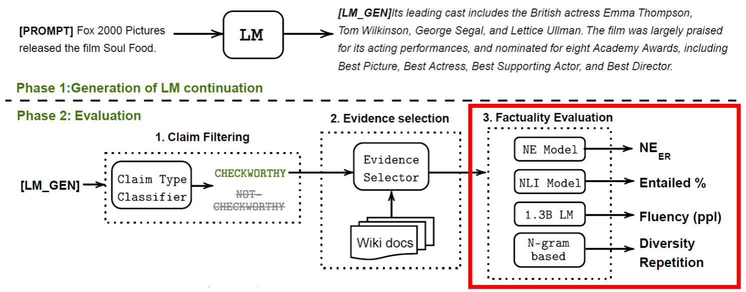
q: query ; generated text

C: Wikipedia sentences as candidates

Outline

1. Background
2. Factuality Prompts
3. Evaluation Metrics
4. Factuality Analysis of Pretrained LMs
5. Factual-Nucleus Sampling
6. Factuality-Enhanced Continued Training
7. Discussion

Evaluation Metrics



1. Hallucinated Named Entity (NE) Error

- A model is hallucinating (i.e., making factual error) if it generates a NE that does not appear in the ground-truth knowledge:

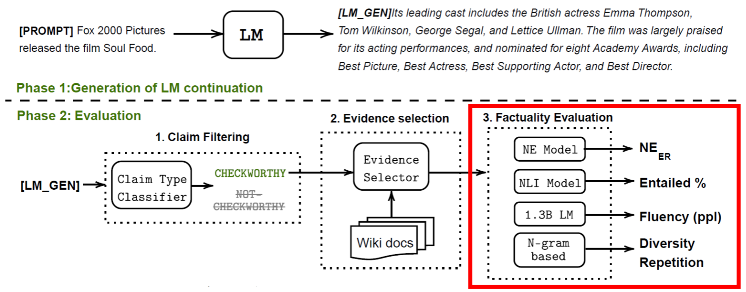
$$\mathrm{NE_{ER}} = |\mathrm{HALLU_{NE}}| \;/\; |\mathrm{ALL_{NE}}|$$

, where $\mathrm{ALL_{NE}}$ is the set of all the NEs, detected in LM generation, and $\mathrm{HALLU_{NE}}$ is a subset of $\mathrm{NE_{ALL}}$ that does not appear in the ground-truth Wikipedia document

- As NEs consist of multiple words, partial n-gram overlaps are also treated as a "match"
- e.g., Barack Hussein Obama vs. Obama
- Stopwords (e.g., the, a) are not considered in the partial n-gram overlaps
- NE detection model: Spacy.io

Evaluation Metrics

Example	English Translation	Label
P: 저는, 그냥 알아내려고 거기 있었어요. H: 이해하려고 노력하고 있었어요.	I was just there just trying to figure it out. I was trying to understand.	Entailment
P: 저는, 그냥 알아내려고 거기 있었어요. H: 나는 처음부터 그것을 잘 이해했다.	I was just there just trying to figure it out. I understood it well from the beginning.	Contradiction
P: 저는, 그냥 알아내려고 거기 있었어요. H: 나는 돈이 어디로 갔는지 이해하려고 했어요.	I was just there just trying to figure it out. I was trying to understand where the money went.	Neutral



2. Entailment Ratio

- Textual Entailment is a task of determining whether a hypothesis is entailed by, refused by, or neutral to a given promise

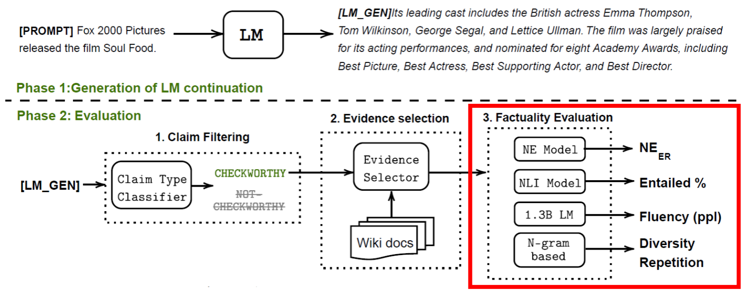
- Entailment Ratio:

$$\mathrm{Entail_R} = |\mathrm{ENTAIL_{gen}}| \,/\, |\mathrm{ALL_{gen}}|$$

, where $\mathrm{ALL_{gen}}$ is the set of all generations, and $\mathrm{ENTAIL_{gen}}$ is a set of generations that are entailed by an entailment model

- Pretrained entailment model: https://pytorch.org/hub/pytorch_fairseq_roberta/

Evaluation Metrics



3. Generation Quality Evaluation

1) PPL

- Mean perplexity of generated continuations
- *Fluency*

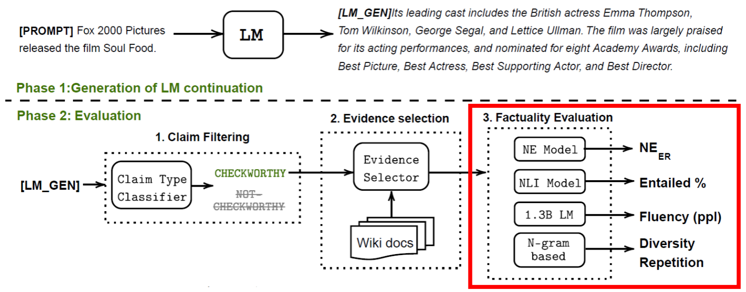
2) Div.

- Mean number of distinct 4-grams, normalized by the length of text among 10 generations for each prompt (In total 160,000 generations)
- *Diversity*
- Any problem?: Tevet et al., ACL 2021

3) Rep.

- Exploited model: https://github.com/ari-holtzman/degen
- *Repetition*

Evaluation Metrics



4. Correlation with Human Judgement

- Human annotations for 200 randomly chosen LM continuations are obtained

- 1 = Factual (If supporting Wikipedia evidence is founded)
- 0 = Nonfactual (Otherwise)

- Two types of fact-checking annotation
- Expert: One of the authors
- Majority-voting: English speaking workers on *Appen.com* platform

Table 2: Pearson correlation coefficients between human factuality annotation and our factuality metrics. p-values for all results are 0.00.

Annotation	Entail_{R}	NE_{ER}
Expert	0.81	-0.77
Majority-voting	0.47	-0.46

Negative Correlation
(The lower, the better)

Outline

1. Background
2. Factuality Prompts
3. Evaluation Metrics
4. Factuality Analysis of Pretrained LMs
5. Factual-Nucleus Sampling
6. Factuality-Enhanced Continued Training
7. Discussion

Factuality Analysis of Pretrained LMs

1. Factuality analysis of LMs from three aspects

1) Model size
- 126M, 357M, 1.3B, 8.3B, and 530B

2) Prompt type
- Both Factual and Nonfactual

3) Decoding algorithm
- Greedy, nucleus sampling (a.k.a., top-p)

Detail about pre-trained LMs All LMs with different sizes are pre-trained on the same corpus, following the experimental details in [77].

Table 6: Architecture details of pre-trained LMs.

Models (#/parameters)	#/layers	#/hidden size	#/ attention heads
126M	12	768	12
357M	24	1024	16
1.3B	24	2048	32
8.3B	40	4096	64
530B	105	20480	128

LMs: Megatron LM (https://arxiv.org/pdf/1909.08053.pdf)

Factuality Analysis of Pretrained LMs

1. Factuality analysis of LMs from three aspects

4) Result Analysis:
- Model size?
- Factual? Nonfactual?
- Greedy? Top-p?

Table 3: The factuality of LMs with different parameter size from 12M to 530B. NE_{ER} refers to the named-entity error, $Entail_R$ refers to entailment ratio, Div. refers to distinct 4-grams, and Rep. refers to repetition. ↑ means the higher the better, and ↓ means the lower the better.

Size	Decode	Factual Prompt				Nonfactual Prompt			
		NE_{ER}↓	$Entail_R$↑	Div.↑	Rep.↓	NE_{ER}↓	$Entail_R$↑	Div.↑	Rep.↓
126M	p=0.9	63.69%	0.94%	0.90	0.58%	67.71%	0.76%	0.90	0.38%
	greedy	48.55%	8.36%	0.03	59.06%	54.24%	6.25%	0.03	59.90%
357M	p=0.9	56.70%	2.01%	0.87	0.55%	60.80%	1.42%	0.88	0.35%
	greedy	43.04%	14.25%	0.03	45.18%	46.79%	9.89%	0.04	46.30%
1.3B	p=0.9	52.42%	2.93%	0.88	0.24%	56.82%	2.04%	0.89	0.25%
	greedy	39.87%	12.91%	0.05	33.13%	45.02%	8.75%	0.05	36.20%
8.3B	p=0.9	40.59%	7.07%	0.90	0.11%	47.49%	3.57%	0.91	0.08%
	greedy	28.06%	22.80%	0.07	19.41%	32.29%	15.01%	0.07	13.26%
530B	p=0.9	33.30%	11.80%	0.90	0.13%	40.49%	7.25%	0.92	0.08%
	greedy	**20.85%**	**31.94%**	0.08	15.88%	27.95%	19.91%	0.08	16.28%

5) Analysis of error types
- 530B LM & Greedy

- **Named Entity Mix-up**: Mixing up similar types of the named entity. For example, LM generated "*The movie is based on the novel of the same name by Gayle Forman.*" about a film called "*The Best of Me*". However, the correct author's name is "Nicholas Sparks", not "Gayle Forman". Note that Gayle Forman is also an American young adult fiction author who writes similar type of novels as Nicholas Sparks.
- **Fabricated Fact:** Fabricating some random facts. For example, "*Samuel Witwer's father is a Lutheran minister.*" Note that, the pretraining corpus contains non-factual or fictional information, which can also contribute to such fabricated facts.

Factuality Analysis of Pretrained LMs

1. Factuality analysis of LMs from three aspects

4) Result Analysis:
- Model size?
- Factual? Nonfactual?
- Greedy? Top-p?

5) Analysis of error types
- 530B LM & Greedy

Table 3: The factuality of LMs with different parameter size from 12M to 530B. $\mathrm{NE}_{\mathrm{ER}}$ refers to the named-entity error, $\mathrm{Entail}_{\mathrm{R}}$ refers to entailment ratio, Div. refers to distinct 4-grams, and Rep. refers to repetition. ↑ means the higher the better, and ↓ means the lower the better.

Size	Decode	Factual Prompt				Nonfactual Prompt			
		$\mathrm{NE}_{\mathrm{ER}}$↓	$\mathrm{Entail}_{\mathrm{R}}$↑	Div.↑	Rep.↓	$\mathrm{NE}_{\mathrm{ER}}$↓	$\mathrm{Entail}_{\mathrm{R}}$↑	Div.↑	Rep.↓
126M	p=0.9	63.69%	0.94%	0.90	0.58%	67.71%	0.76%	0.90	0.38%
	greedy	48.55%	8.36%	0.03	59.06%	54.24%	6.25%	0.03	59.90%
357M	p=0.9	56.70%	2.01%	0.87	0.55%	60.80%	1.42%	0.88	0.35%
	greedy	43.04%	14.25%	0.03	45.18%	46.79%	9.89%	0.04	46.30%
1.3B	p=0.9	52.42%	2.93%	0.88	0.24%	56.82%	2.04%	0.89	0.25%
	greedy	39.87%	12.91%	0.05	33.13%	45.02%	8.75%	0.05	36.20%
8.3B	p=0.9	40.59%	7.07%	0.90	0.11%	47.49%	3.57%	0.91	0.08%
	greedy	28.06%	22.80%	0.07	19.41%	32.29%	15.01%	0.07	13.26%
530B	p=0.9	33.30%	11.80%	0.90	0.13%	40.49%	7.25%	0.92	0.08%
	greedy	**20.85%**	**31.94%**	0.08	15.88%	27.95%	19.91%	0.08	16.28%

Any problem?: Trade-offs between factuality and diversity/repetition

- **Named Entity Mix-up**: Mixing up similar types of the named entity. For example, LM generated "*The movie is based on the novel of the same name by Gayle Forman.*" about a film called "*The Best of Me*". However, the correct author's name is "Nicholas Sparks", not "Gayle Forman". Note that Gayle Forman is also an American young adult fiction author who writes similar type of novels as Nicholas Sparks.
- **Fabricated Fact:** Fabricating some random facts. For example, "*Samuel Witwer's father is a Lutheran minister.*" Note that, the pretraining corpus contains non-factual or fictional information, which can also contribute to such fabricated facts.

Outline

1. Background
2. Factuality Prompts
3. Evaluation Metrics
4. Factuality Analysis of Pretrained LMs
5. Factual-Nucleus Sampling
6. Factuality-Enhanced Continued Training
7. Discussion

Factual-Nucleus Sampling

Size	Decode	Factual Prompt			
		$NE_{ER}\downarrow$	$Entail_R\uparrow$	Div.↑	Rep.↓
126M	p=0.9	63.69%	0.94%	0.90	0.58%
	greedy	48.55%	8.36%	0.03	59.06%
357M	p=0.9	56.70%	2.01%	0.87	0.55%
	greedy	43.04%	14.25%	0.03	45.18%
1.3B	p=0.9	52.42%	2.93%	0.88	0.24%
	greedy	39.87%	12.91%	0.05	33.13%
8.3B	p=0.9	40.59%	7.07%	0.90	0.11%
	greedy	28.06%	22.80%	0.07	19.41%
530B	p=0.9	33.30%	11.80%	0.90	0.13%
	greedy	**20.85%**	**31.94%**	0.08	15.88%

1. Why Factual-Nucleus Sampling?

- Assumption: Randomness of sampling is more harmful to factuality when it is used to generate the latter part of a sentence than the beginning of a sentence
- As a generation proceeds, the premise become more determined, and fewer word choices can make the sentence factual

- e.g., “Samuel Witwer’s father is” [Factual]
- e.g., “Samuel Witwer’s father is a Lutheran minister” [Nonfactual]

- Factual-Nucleus Sampling dynamically adapts the “nucleus” p along the generation of each sentence

Size	Decode	Factual Prompt			
		$\mathrm{NE}_{\mathrm{ER}}\downarrow$	$\mathrm{Entail}_{\mathrm{R}}\uparrow$	Div.↑	Rep.↓
126M	p=0.9	63.69%	0.94%	0.90	0.58%
	greedy	48.55%	8.36%	0.03	59.06%
357M	p=0.9	56.70%	2.01%	0.87	0.55%
	greedy	43.04%	14.25%	0.03	45.18%
1.3B	p=0.9	52.42%	2.93%	0.88	0.24%
	greedy	39.87%	12.91%	0.05	33.13%
8.3B	p=0.9	40.59%	7.07%	0.90	0.11%
	greedy	28.06%	22.80%	0.07	19.41%
530B	p=0.9	33.30%	11.80%	0.90	0.13%
	greedy	**20.85%**	**31.94%**	0.08	15.88%

Factual-Nucleus Sampling

1. Why Factual-Nucleus Sampling?

- Factual-Nucleus Sampling dynamically adapts the "nucleus" p along the generation of each sentence
- The nucleus probability p_t to generate the t-th token within each sentence is:

$$p_t = \max\{\omega, p \times \lambda^{t-1}\}$$

, where λ is the decay factor for top-p probability, and ω is lower bound

- **λ-decay**: Given that top-p sampling pool is selected as a set of subwords whose cumulative probability exceeds p, we gradually decay the p value with decay factor λ at each generation step to reduce the "randomness" through time.
- **p-reset**: The nucleus probability p can quickly decay to a small value after a long generation. So, we reset the p-value to the default value at the beginning of every new sentence in the generation (we identify the beginning of a new sentence by checking if the previous step has generated a full-stop). This reduces the unnecessary cost of diversity for any long generations.
- **ω-bound**: If λ-decay is applied alone, the p-value could become too small to be equivalent to greedy decoding and hurt diversity. To overcome this, we introduce a lower-bound ω to limit how far p-value can be decayed.

Factual-Nucleus Sampling

2. Results

- λ is 0.5? (~ Greedy)
- λ-decay?
- p-reset?
- ω-bound?

Table 4: **1.3B** LM results with different decoding algorithms. NE_{ER} refers to named-entity error, $Entail_R$ refers to entailed class ratio, Div. refers to distinct 4-grams, and Rep. refers to repetition. ↑ means the higher, the better, and ↓ means the lower, the better. For factual-nucleus sampling, p, λ and ω are nucleus probability, decay factor, and decay lowerbounds respectively. See more results with different hyperparameters in Figure 2a and 2b.

Decoding	**Factual Prompt**				**Nonfactual Prompt**			
	NE_{ER}↓	$Entail_R$↑	Div.↑	Rep.↓	NE_{ER}↓	$Entail_R$↑	Div.↑	Rep.↓
Greedy	39.9%	12.9%	0.05	33.1%	45.0%	8.8%	0.05	36.2%
Top-p 0.9	52.4%	2.9%	0.88	0.2%	56.8%	2.0%	0.89	0.3%
p | λ	Top-p + λ-decay							
0.9 | 0.9	41.1%	10.8%	0.43	30.7%	45.7%	6.8%	0.47	34.5%
0.9 | 0.5	39.9%	13.0%	0.08	33.1%	44.9%	9.1%	0.09	35.9%
p | λ	Top-p + λ-decay + p-reset							
0.9 | 0.9	41.5%	10.3%	0.52	10.3%	45.4%	6.3%	0.57	9.1%
0.9 | 0.5	39.3%	12.8%	0.34	17.8%	44.5%	8.4%	0.45	18.9%
p | λ | ω	Top-p + λ-decay + p-reset + ω-bound (*factual-nucleus sampling*)							
0.9 | 0.9 | 0.7	46.2%	5.0%	0.78	1.2%	52.2%	3.2%	0.80	0.5%
0.9 | 0.9 | 0.3	42.1%	10.1%	0.55	7.1%	46.5%	5.6%	0.59	6.4%
0.9 | 0.9 | 0.2	41.7%	9.9%	0.52	8.6%	45.6%	6.2%	0.56	7.6%
0.9 | 0.5 | 0.3	41.0%	12.2%	0.47	13.0%	46.0%	7.0%	0.51	12.7%
0.9 | 0.5 | 0.2	39.3%	12.8%	0.38	16.1%	45.2%	7.8%	0.42	16.9%

Factual-Nucleus Sampling

2. Results

- Top-p? (Div. & Rep.)

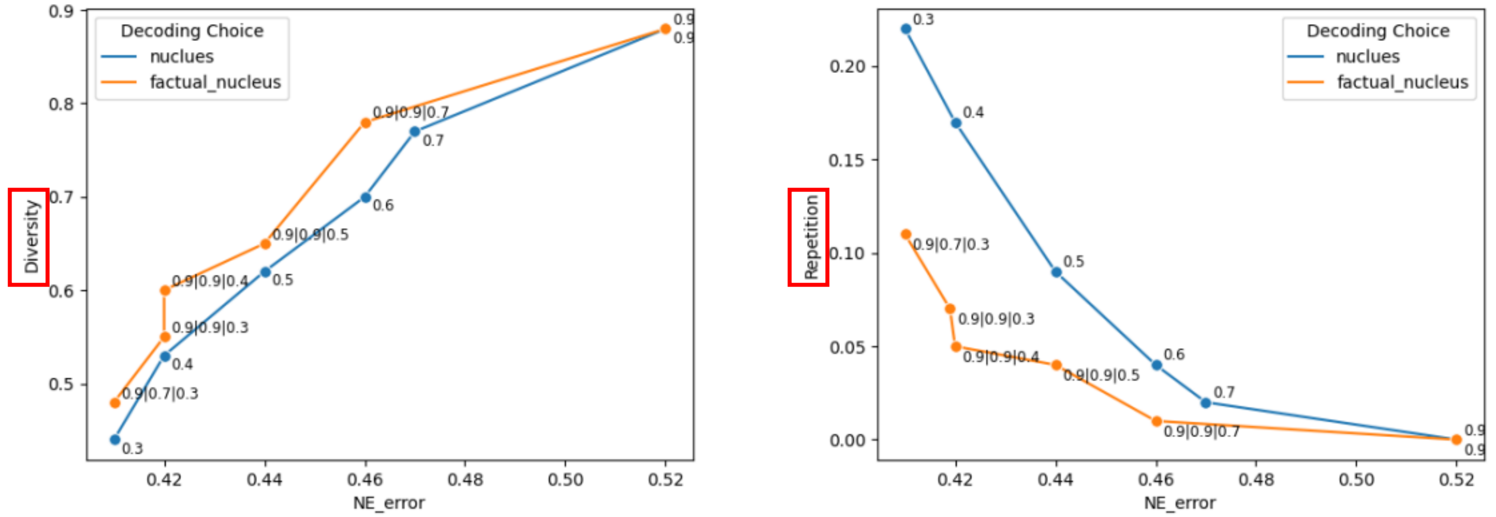
Table 4: **1.3B** LM results with different decoding algorithms. NE_{ER} refers to named-entity error, Entail_{R} refers to entailed class ratio, Div. refers to distinct 4-grams, and Rep. refers to repetition. ↑ means the higher, the better, and ↓ means the lower, the better. For factual-nucleus sampling, p, λ and ω are nucleus probability, decay factor, and decay lowerbounds respectively. See more results with different hyperparameters in Figure 2a and 2b.

Decoding	**Factual Prompt**				**Nonfactual Prompt**			
	NE_{ER}↓	Entail_{R}↑	Div.↑	Rep.↓	NE_{ER}↓	Entail_{R}↑	Div.↑	Rep.↓
Greedy	39.9%	12.9%	0.05	33.1%	45.0%	8.8%	0.05	36.2%
Top-p 0.9	52.4%	2.9%	0.88	0.2%	56.8%	2.0%	0.89	0.3%
p | λ	Top-p + λ-decay							
0.9 | 0.9	41.1%	10.8%	0.43	30.7%	45.7%	6.8%	0.47	34.5%
0.9 | 0.5	39.9%	13.0%	0.08	33.1%	44.9%	9.1%	0.09	35.9%
p | λ	Top-p + λ-decay + p-reset							
0.9 | 0.9	41.5%	10.3%	0.52	10.3%	45.4%	6.3%	0.57	9.1%
0.9 | 0.5	39.3%	12.8%	0.34	17.8%	44.5%	8.4%	0.45	18.9%
p | λ | ω	Top-p + λ-decay + p-reset + ω-bound (*factual-nucleus sampling*)							
0.9 | 0.9 | 0.7	46.2%	5.0%	0.78	1.2%	52.2%	3.2%	0.80	0.5%
0.9 | 0.9 | 0.3	42.1%	10.1%	0.55	7.1%	46.5%	5.6%	0.59	6.4%
0.9 | 0.9 | 0.2	41.7%	9.9%	0.52	8.6%	45.6%	6.2%	0.56	7.6%
0.9 | 0.5 | 0.3	41.0%	12.2%	0.47	13.0%	46.0%	7.0%	0.51	12.7%
0.9 | 0.5 | 0.2	39.3%	12.8%	0.38	16.1%	45.2%	7.8%	0.42	16.9%

Factual-Nucleus Sampling

3. Comparison between nucleus and factual_nucleus

- Trade-off between NE_error and Div./Rep.



(a) Diversity vs. NE_{ER} (b) Repetition vs. NE_{ER}

Figure 2: Comparison between nucleus sampling (blue line) and factual-nucleus sampling (orange line). The x-axis is named entity error NE_{ER}. The y-axes are diversity and repetition in (a) and (b) respectively. The lower the repetition, the better. It is evident that factual-nucleus sampling has better trade-offs between factuality and diversity/repetition. For a reference, the diversity score of randomly sampled 5000 Wikipedia documents is 0.767.

Outline

1. Background
2. Factuality Prompts
3. Evaluation Metrics
4. Factuality Analysis of Pretrained LMs
5. Factual-Nucleus Sampling
6. Factuality-Enhanced Continued Training
7. Discussion

Factuality-Enhanced Continued Training

1. Prepending TOPICPREFIX

- Due to the GPU memory limit and computation efficiency, it is common to chunk documents in LM training corpus
- This causes the "fragmentation" of information and leads to wrong associations of entities that appear in independent documents
- TOPICPREFIX mainly utilizes name of Wikipedia document

The following Wikipedia paragraph about Barack Obama:

Barack Hussein Obama II (born August 4, 1961) is an American politician who served as the 44th president of the United States from 2009 to 2017. He was the first African-American president of the United States. A member of the Democratic Party, he previously served as a U.S. senator from Illinois from 2005 to 2008 and as an Illinois state senator from 1997 to 2004.

is transferred into:

Barack Obama ==> Barack Hussein Obama II (born August 4, 1961) is an American politician who served as the 44th president of the United States from 2009 to 2017. Barack Obama ==> He was the first African-American president of the United States. Barack Obama ==> A member of the Democratic Party, he previously served as a U.S. senator from Illinois from 2005 to 2008 and as an Illinois state senator from 1997 to 2004. .

Factuality-Enhanced Continued Training

2. Sentence Completion Loss

- To address the incorrect association learned between entities
- LM failed to accurately complete the sentence given the generated context

- e.g., "Samuel Witwer's father is Lutheran minister"

- Sentence completion loss makes the LM focus on predicting the subwords later in the sentence
- It determines a pivot t for each sentence, and then apply zero-masking for all token prediction loss before t
- Training only (No pivot needed during inference time)

Factuality-Enhanced Continued Training

2. Sentence Completion Loss

- Three strategies to determine the pivot t:

  - SC_{HALF}: pivot $t = 0.5\times$ sentence-length.
  - SC_{RANDOM}: random pivot, e.g., $t \sim \text{uniform}[0.25, 0.75]\times$ sentence-length.
  - SC_{ROOT}: pivot $t =$ position of ROOT (relation) from dependency parsing.

- Which strategy is the best?: SC_{HALF}
- Domain-Adaptive Training?: Training corpus consists of both factual and nonfactual text
- Nonfactual texts could be the problem
- So, Domain-Adaptive setting excludes nonfactual things during training

- Effect of TOPICPREFIX?
- Effect of SC Loss?

Table 5: Results for factuality enhanced training. The decoding settings are formatted as: nucleus probability p, decay rate λ, lower-bound ω.

Decoding	**Factual Prompt**				**Nonfactual Prompt**			
($p \mid \lambda \mid \omega$)	$\text{NE}_{\text{ER}}\downarrow$	$\text{Entail}_{\text{R}}\uparrow$	Div.	Rep.	NE_{ER}	Entail_{R}	Div.	Rep.
Vanilla Pretrained LM (1.3B)								
0.9	52.4%	2.9%	0.88	0.2%	56.8%	2.0%	0.89	0.3%
0.9 | 0.9 | 0.3	42.1%	10.1%	0.55	7.1%	46.5%	5.6%	0.59	6.4%
Factual Domain-Adaptive Training with Wikipedia (1.3B)								
0.9	52.5%	2.8%	0.85	0.2%	55.8%	2.2%	0.86	0.1%
0.9 | 0.9 | 0.3	42.7%	7.1%	0.51	7.2%	48.2%	4.9%	0.56	6.0%
TopicPrefix (1.3B)								
0.9	34.4%	4.2%	0.84	0.3%	36.2%	2.7%	0.85	0.2%
0.9 | 0.9 | 0.3	27.6%	8.7%	0.43	8.0%	30.5%	6.1%	0.47	6.9%
TopicPrefix + SC_{ROOT} (1.3B)								
0.9	32.5%	6.7%	0.83	1.2%	34.3%	4.6%	0.84	1.1%
0.9 | 0.9 | 0.3	24.7%	15.8%	0.40	13.6%	27.6%	9.1%	0.44	13.7%
TopicPrefix + SC_{RANDOM} (1.3B)								
0.9	32.0%	7.9%	0.81	1.2%	34.2%	5.5%	0.83	1.1%
0.9 | 0.9 | 0.3	23.6%	17.6%	0.39	14.2%	26.9%	9.3%	0.42	13.2%
TopicPrefix + SC_{HALF} (1.3B)								
0.9	31.6%	7.6%	0.81	1.4%	33.5%	5.1%	0.83	1.5%
0.9 | 0.9 | 0.3	23.6%	17.4%	0.38	14.4%	27.2%	10.2%	0.42	13.1%
Vanilla Pretrained LM (530B)								
0.9	33.3%	11.8%	0.90	0.1%	40.5%	7.25%	0.92	0.1%
TopicPrefix + SC_{HALF} (530B)								
0.9	18.3%	19.3%	0.68	0.1%	21.7%	13.7%	0.68	0.1%
0.9 | 0.9 | 0.3	**14.5%**	**25.5%**	0.33	0.2%	**17.7%**	**20.0%**	0.33	0.1%

Outline

1. Background
2. Factuality Prompts
3. Evaluation Metrics
4. Factuality Analysis of Pretrained LMs
5. Factual-Nucleus Sampling
6. Factuality-Enhanced Continued Training
7. Discussion

Discussion

1. Any problems?

1) Diversity measure (Distinct n-gram)

- N-gram-based metrics perform poorly, indicating they do not measure content diversity well (Tevet et al., ACL 2021)

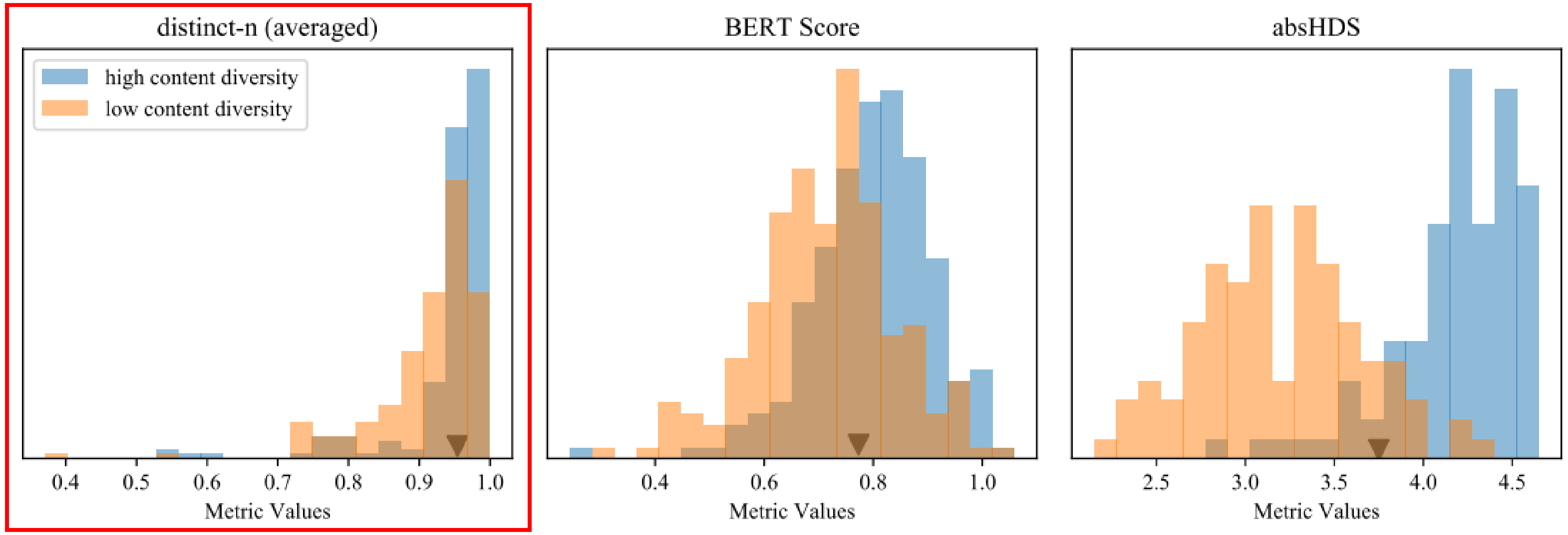


Figure 4: *conTest*: histograms of metric values of n-gram (distinct n-grams), neural (BERT-Score) and human (absHDS) metrics for *promptGen*. The **orange** histogram represents the distribution of the *low content diversity class*, the **blue** histogram represents the distribution of the *high content diversity class* and **brown** is the intersection between the two. Pointing down triangles represent the threshold η of the optimal classifiers. The histograms show how each metric separates the two classes.

Discussion

1. Any problems?

2) Assumption

- e.g., “Samuel Witwer’s father is” [Factual]
- e.g., “Samuel Witwer’s father is a Lutheran minister” [Nonfactual]

- What if..
- e.g., “Jason’s father is a Lutheran minister” [Factual]
- e.g., “Lutheran minister’s father is Jason” [Nonfactual]

Discussion

1. Any problems?

3) Trade-off

- That’s not make sense

Table 3: The factuality of LMs with different parameter size from 12M to 530B. NE_{ER} refers to the named-entity error, $Entail_R$ refers to entailment ratio, Div. refers to distinct 4-grams, and Rep. refers to repetition. ↑ means the higher the better, and ↓ means the lower the better.

Size	Decode	Factual Prompt				Nonfactual Prompt			
		NE_{ER}↓	$Entail_R$↑	Div.↑	Rep.↓	NE_{ER}↓	$Entail_R$↑	Div.↑	Rep.↓
126M	p=0.9	63.69%	0.94%	0.90	0.58%	67.71%	0.76%	0.90	0.38%
	greedy	48.55%	8.36%	0.03	59.06%	54.24%	6.25%	0.03	59.90%
357M	p=0.9	56.70%	2.01%	0.87	0.55%	60.80%	1.42%	0.88	0.35%
	greedy	43.04%	14.25%	0.03	45.18%	46.79%	9.89%	0.04	46.30%
1.3B	p=0.9	52.42%	2.93%	0.88	0.24%	56.82%	2.04%	0.89	0.25%
	greedy	39.87%	12.91%	0.05	33.13%	45.02%	8.75%	0.05	36.20%
8.3B	p=0.9	40.59%	7.07%	0.90	0.11%	47.49%	3.57%	0.91	0.08%
	greedy	28.06%	22.80%	0.07	19.41%	32.29%	15.01%	0.07	13.26%
530B	p=0.9	33.30%	11.80%	0.90	0.13%	40.49%	7.25%	0.92	0.08%
	greedy	**20.85%**	**31.94%**	0.08	15.88%	27.95%	19.91%	0.08	16.28%

Thank you

https://jeiyoon.github.io/