Zero-shot Prompting

- GPT-3/4/ChatGPT can handle lots of existing tasks based purely on incidental exposure to them in pre-training
  - Example from summarization: the token “tl;dr” (“too long; didn’t read”) is an indicator of summaries in the wild
- We’ll discuss two paradigms: **zero-shot prompting**, where no examples are given to a model (just a text specification), and **few-shot prompting**, where a few examples are given in-context
- Both paradigms can theoretically handle classification, text generation, and more!

Zero-shot Prompting

- Single unlabeled datapoint ***x***, want to predict label *y*

  x = *The movie's acting could've been better, but the visuals and directing were top-notch.*

- Wrap ***x*** in a template we call a verbalizer ***v***

Review: The movie's acting could've been better, but the visuals and directing were top-notch.
Out of positive, negative, or neutral, this review is

$\downarrow$

GPT-3

$\downarrow$

neutral

Zero-shot Prompting

- Single unlabeled datapoint ***x***, want to predict label *y*

  x = *The movie's acting could've been better, but the visuals and directing were top-notch.*

- Wrap ***x*** in a template we call a verbalizer ***v***

Review: The movie's acting could've been better, but the visuals and directing were top-notch.
On a 1 to 4 star scale, the reviewer would probably give this movie

↓

GPT-3

↓

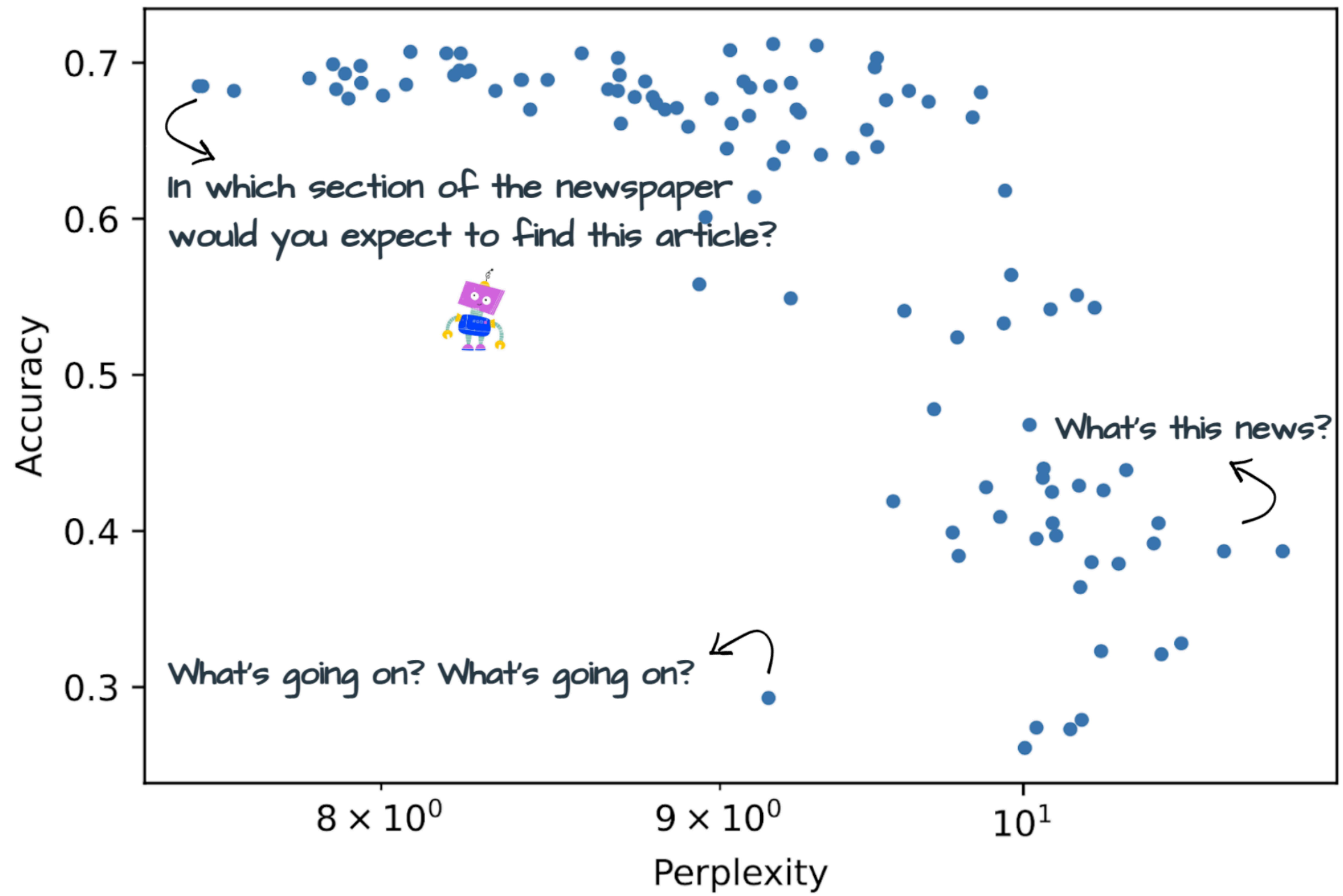
3 stars

Zero-shot Classification: Approaches

- **Approach 1:** Generate from the model and parse the generation
  - What if you ask for a star rating and it doesn't give you a number of stars but just says something else?
- **Approach 2:** Compare probs: "*Out of positive, negative, or neutral, this review is _*". Compare P(*positive* | ***x***), P(*neutral* | ***x***), P(*negative* | ***x***)
  - This constrains the model to only output a valid answer, and you can normalize these probabilities to get a distribution
- How much difference does changing the prompt make?

Variability in Prompts

- Plot: large number of prompts produced by {manual writing, paraphrasing, backtranslation}



y-axis: task performance

x-axis: perplexity of the prompt. How natural is it? How much does it appear in the pre-training data?

- Caveat: a little bit of prompt engineering will usually get you to a decent performance point

Gonen et al. (2022)

Variability in Prompts

Task	Perplexity-score corr.		Perplexity-acc corr.		Avg Acc	Acc 50%
	Pearson	Spearman	Pearson	Spearman		
Antonyms	**-0.41	**-0.53	–	–	–	–
GLUE Cola	-0.15	-0.14	-0.04	-0.02	47.7	57.1
Newspop	*-0.24	**-0.26	*-0.20	-0.18	66.4	72.9
AG News	**-0.63	**-0.68	**-0.77	**-0.81	57.5	68.7
IMDB	**0.35	**0.40	0.14	*0.20	86.2	91.0
DBpedia	**-0.50	**-0.44	**-0.51	**-0.42	46.7	55.2
Emotion	-0.14	-0.19	**-0.30	**-0.32	16.4	23.0
Tweet Offensive	*-0.19	0.07	0.18	*0.23	51.3	55.8

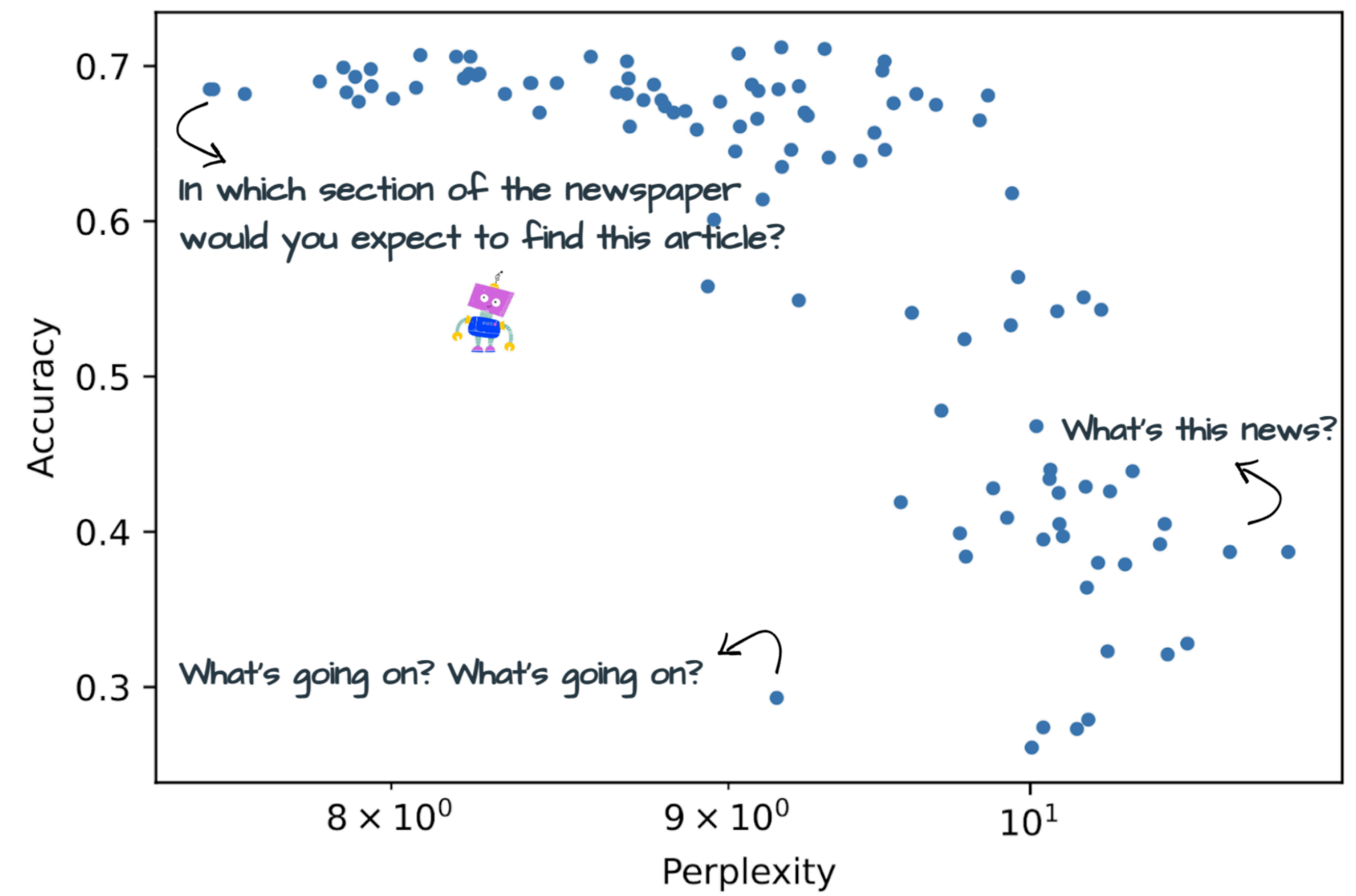
- OPT-175B: average of best 50% of prompts is much better than average over all prompts

Gonen et al. (2022)

Prompt Optimization

- A number of methods exist for searching over prompts (either using gradients or black-box optimization)

- Most of these do not lead to dramatically better results than doing some manual engineering/ hill-climbing (and they may be computationally intensive)



- RLHF models like ChatGPT are also better at "understanding" prompts, so less engineering is needed

Gonen et al. (2022)