JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

LLM Use Cases: Neural Topic Models

Overview

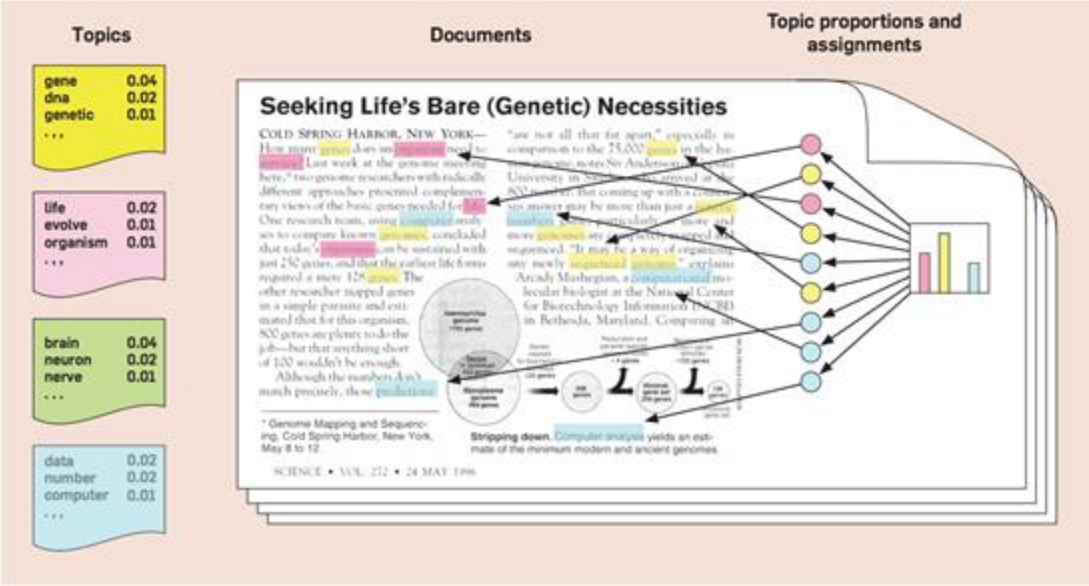
- Recap:
  - Last class: LLMs (MLMs) as classifiers and for metaphor detection

- Today:
  - Continuing LLM use cases, with a focus on Topic Modeling
    - Neural LDA (ProdLDA, CTM)
    - Instruction Tuning and Alignment
    - Beyond LDA (BERTtopic, TopicGPT, LLooM)

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Neural Topic Models

Recall: LDA Topic Model



- Unsupervised clustering
- Discover topics (themes, frames) inductively from the data
- Most common paradigm: LDA
  - Documents are mixtures of topics
  - Topics are mixtures of vocabulary

Recall: LDA Topic Model

- Goal: Estimate the posterior distribution

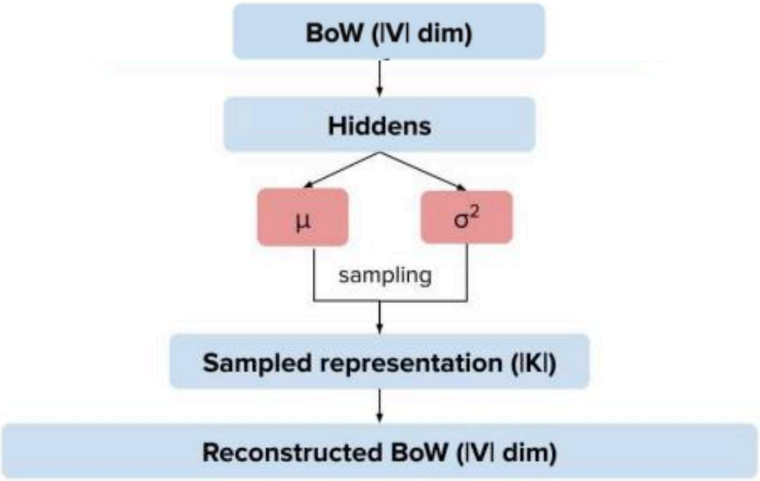
- Direct inference is intractable

- Instead we use:
  - Variational Inference
  - Gibbs Sampling

- Applying these inference methods to new topic models (remember STM) require re-deriving the inference methods

ProdLDA: Formulation

- Proposes an inference method for topic models: "Autoencoded Variational Inference for Topic Models"
  - Application of autoencoding variational Bayes (AEVB)
  - Trains a neural network (an encoder) that directly maps a document to an approximate posterior distribution
  - "Document" – BOW representation



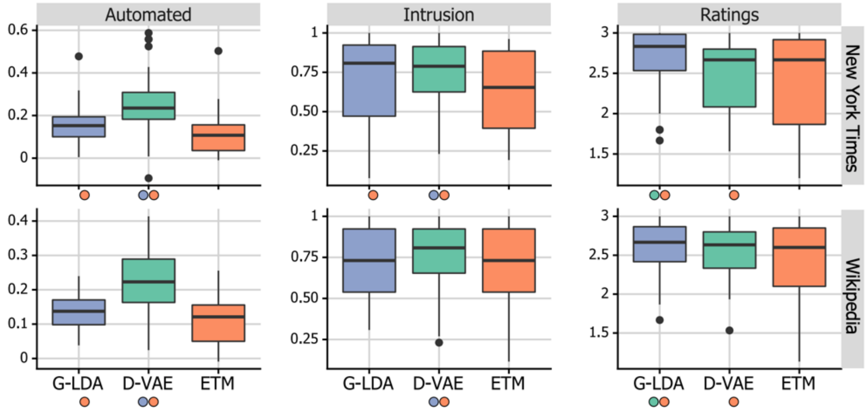
Srivastava, Akash, and Charles Sutton. "Autoencoding Variational Inference For Topic Models." International Conference on Learning Representations. 2017. https://openreview.net/pdf?id=BybtVK9lg
Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." *arXiv preprint arXiv:1312.6114* (2013). https://arxiv.org/abs/1312.6114

ProdLDA: Impact

- Improves over classic LDA in 3 ways:
  - Topic coherence: ProdLDA consistently scores better on automated metrics than LDA, even when LDA is trained using Gibbs sampling.
  - Computational efficiency: fast and efficient at both training and inference
  - Black box: AVITM does not require rigorous mathematical derivations to handle changes in the model, and can be easily applied to a wide range of topic models
    - Demonstrated with ProdLDA (Product-of-Experts LDA), in which the distribution over individual words is a product of experts rather than the mixture model used in LDA

Are neural topic models (Dirichlet-VAEs) actually better?

- Topic model evaluation metrics showed that automated metrics (NPMI) are correlated with human judgements of topic coherence
  - BUT these experiments were done with LDA-style models. Are they still correlated for neural models?

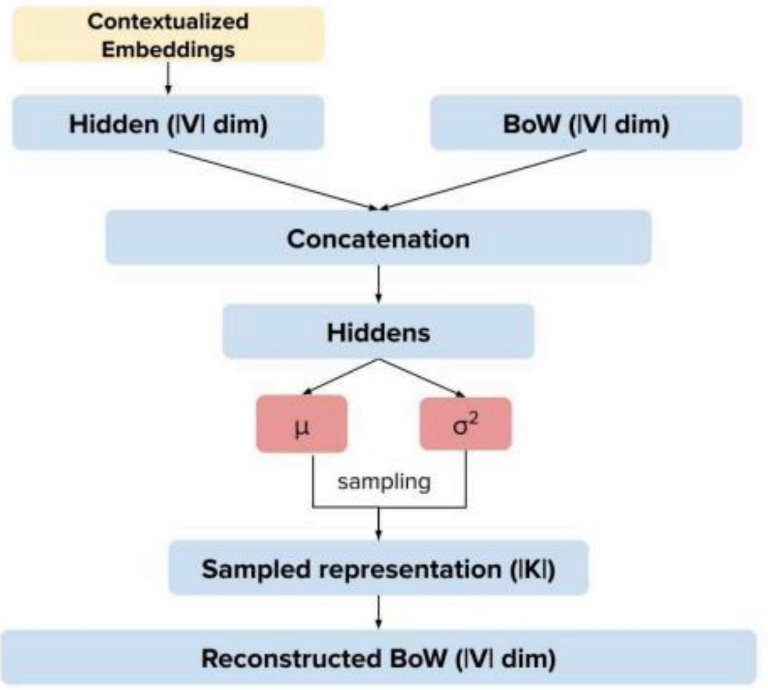


- Automated metrics (NPMI) suggest the VAE-based model is better, but human judgements do not

Hoyle, Alexander, et al. "Is automated topic model evaluation broken? the incoherence of coherence." *Advances in neural information processing systems* 34 (2021): 2018-2033.

CTM: Combined Topic Model

- ProdLDA is a neural topic model, but:
  - it's an approximation of "vanilla" LDA, still using BOW simplifying assumption
  - we want to take advantage of pre-trained models like BERT that are very successful at language tasks in general

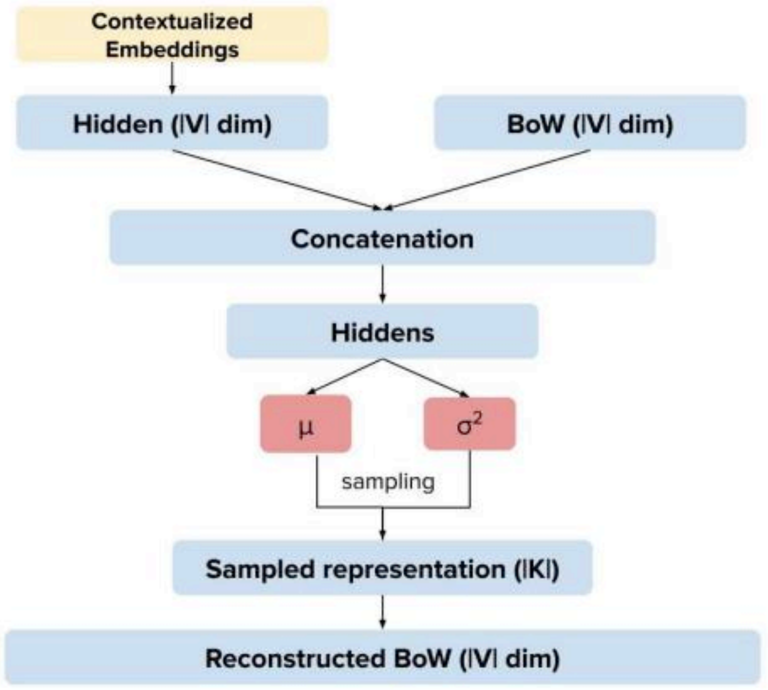


Bianchi, F., Terragni, S., & Hovy, D. (2021). *Pre-training is a Hot Topic: Contextualized Document Embeddings Improve Topic Coherence.* ACL. https://aclanthology.org/2021.acl-short.96/
Bianchi, F., Terragni, S., Hovy, D., Nozza, D., & Fersini, E. (2021). *Cross-lingual Contextualized Topic Models with Zero-shot Learning.* EACL. https://www.aclweb.org/anthology/2021.eacl-main.143/

CTM: Combined Topic Model

- Embedding source:
  - sBERT: modified variant of BERT/RoBERTa that is trained to produce semantically meaningful embeddings

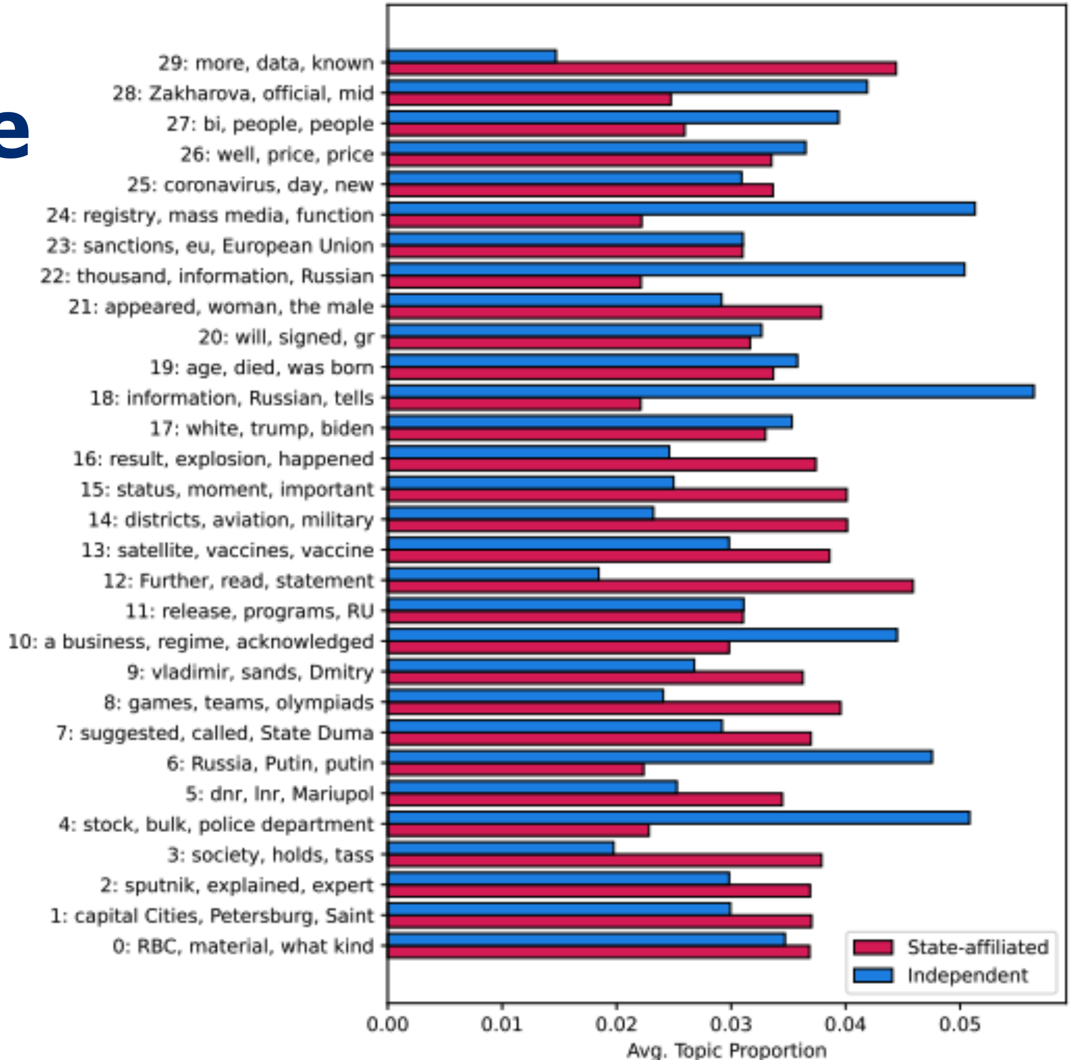
- Evaluation:
  - Automated metrics for topic coherence (nPMI and word embeddings)



Reimers, Nils, and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks." EMNLP. 2019.

Example Use Case

- 30-topic CTM output
- Social media posts by Russia-government affiliated news outputs and independent news outputs about the Russia-Ukraine war



20-topic STM output over Russia/Ukraine social media posts

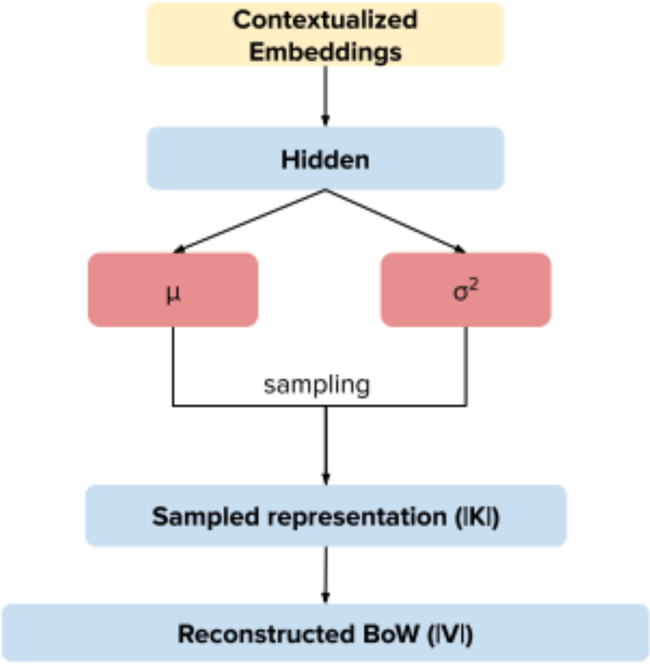
told, further, interview, our, andrey, chief, expert
read, own, stated, commented, words, expert, opportunity
russia, news, situations, russia, statement, about, gas
because of, earlier, media, case, data, person, reported
httpsliferup, video, areas, photo, look, houses, tass
countries, head, informed, sergey, mid, new, countries
children, became, told, result, known, died, one
authorities, court, decision, moscow, communications, may, s
foreign, agent, function, performer, information, Russian, for
more, stated, russia, putin, vladimir, president, alexander
rubles, thousands, bulk, new, january, deeds, shares
Russians, around, coronavirus, world, thousand, country, day
also, which, can, will, companies, yet, which
Moscow, days, became, February, April, we tell, our
usa, against, president, putin, believes, Russian, security
coronavirus, covid, day, new, latest, russia, coronavirus
years, years, year, day, multiple, life, year
this is, why, people, people, very, tells, his
ukraine, ukraine, russian, defence, russian, russian, details
time, which, according to, which, which, Michael, his

RBC, material, what, life, we tell, understand, read, forbes, business, often
capital, petersburg, st, capital, moscow, afternoon, morning, friends, degrees, expected
sputnik, explained, expert, radio, ru, bi, told, told, interview, si
society, conducts, tass, economy, ruptly, politics, reuters, premier, world, michael
actions, Navalny, OVD, info, protest, support, detainees, aleksey, actions, new
DPR, LPR, Mariupol, peaceful, residents, Ukrainian, folk, news, Donbass, Mariupol
Russia, Putin, Putin, this, functions, foreign, agent, political scientist, doing, why
proposed, named, State Duma, warned, deputies, offer, access, draft law, new, offers
games, teams, olympiads, teams, olympics, athletes, championship, gold, team, victory
Vladimir, Sands, Dmitry, President, Putin, Zelensky, Secretary, Kremlin, negotiations, Lukashenko
case, regime, admitted, verdict, freedom, years, accusation, deprivation, former, threatens
release, programs, ru, watch, programme, show, utm, russia, air, TV channel
further, read, statement, did, important, accepted, did, did, accepted, applied
satellite, vaccine, vaccine, coronavirus, vaccine, vaccine, vaccination, who, omicron, health
county, air, military, military, navy, exercise, su, servicemen, enemy, forces
status, moment, important, continues, refused, said, exit, going to, by the time, leadership
result, explosion, occurred, board, killed, injured, accident, preliminary, were, fire
white, trump, baiden, trump, baiden, usa, joe, administration, antony, whites
information, Russian, tells, message, mass, material, functions, foreign, foreign, agent
age, died, born, deceased, life, ussr, birth, actor, roles, soviet
will, signed, qr, may, payments, government, support, law, must, may
appeared, woman, man, girl, instagram, summer, woman, mother, child, inhabitant
thousand, information, Russian, rubles, message, mass, million, about, material, million
sanctions, eu, eu, against, eu, regarding, package, Russian, ban, diplomats
registry, media, function, wrote, performs, requires, foreign agents, nco, foreign agent, law
coronavirus, days, new, last, number, cases, dead, cases, cases, max
rate, price, value, up, up, prices, up, tesla, up, price
bi, people, people, this, si, which, warriors, powers, several, time
Zakharova, official, mid, maria, representative, information, Russian, message, mass, material
more, data, known, commented, applied, situations, speak, became, appreciated, reacted

30-topic CTM output→

Zero-shot cross-lingual topic model

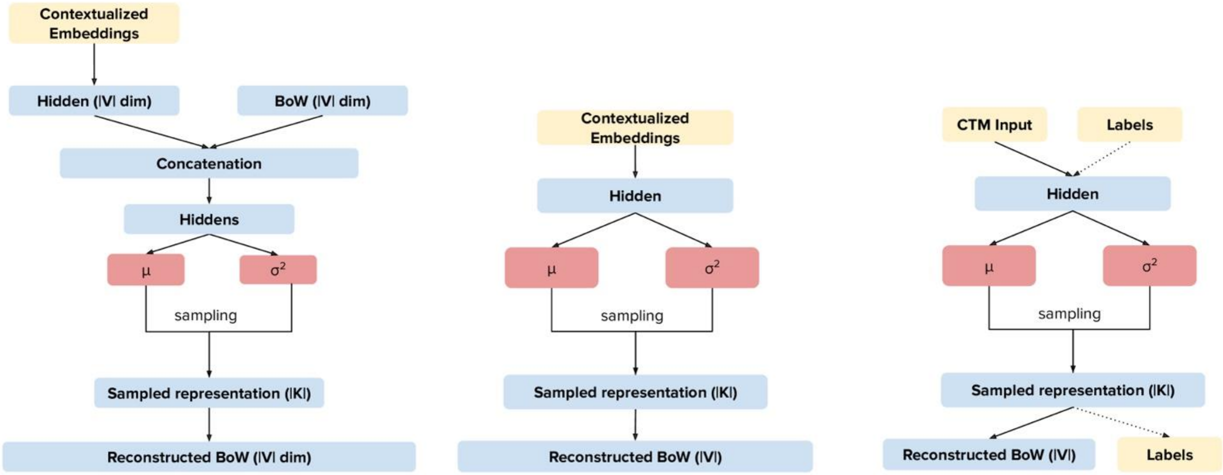
- Replace the input BOW with contextualized embeddings (instead of concatenation)
- We can train model on one language and apply it on a different language (if we use contextualized embeddings from a multilingual model)



CTM Python package

contextualized-topic-models 2.5.0

```
pip install contextualized-topic-models
```



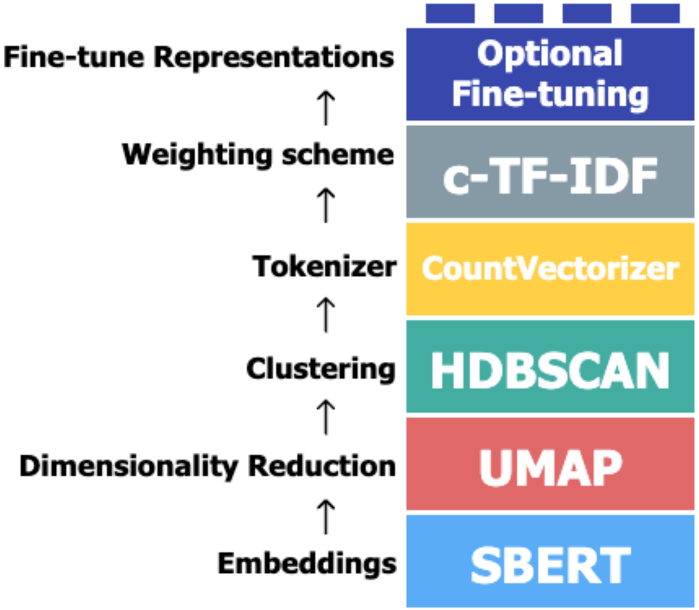
CombinedTM　　ZeroShotTM　　SuperCTM

Thinking higher level: Goals of topic modeling

- LDA became popular because it turned out to be pretty good at identifying trends in data

- Do we actually want better LDA?
  - Not really, goal of topic model is unsupervised investigation of text corpora
  - Example: We'd probably prefer for topics to be coherent descriptions than lists of words

BERTopic: Neural topic modeling with a class-based TF-IDF procedure



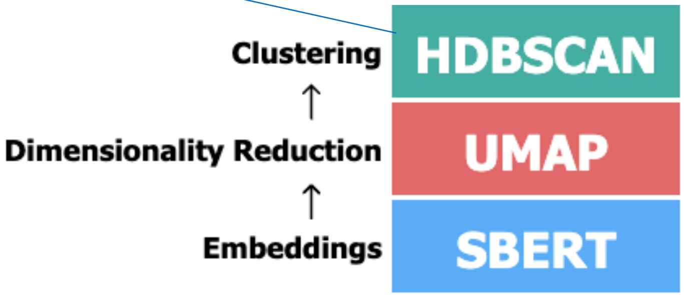
Grootendorst, Maarten. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure." arXiv preprint arXiv:2203.05794 (2022).

BERTopic: Neural topic modeling with a class-based TF-IDF procedure

Assumption: documents about the same topic will be semantically similar (will have similar semantic embeddings)

Hierarchical soft clustering to group common documents

Reduce dimensionality: could use PCA or t-SNE but UMAP has been shown to well-preserve local and global structure



Represent documents as embeddings (sBERT, same as CTM, but can be a different model)

BERTopic: Neural topic modeling with a class-based TF-IDF procedure

- Clustering embeddings is relatively straightforward
- We need some meaningful way to describe what a ”topic” is – what do the documents in a cluster have in common?
- How can we describe words that are more common in each cluster?
  - PMI, log-odds, etc.

- TF-IDF weighting

Recall: TF-IDF weighting

- TF-IDF incorporates two terms that capture these conflicting constraints:
  - **Term frequency (tf):** frequency of the word t in the document

$$tf_{t,d} = \log(count(t,d) + 1)$$

  - Document frequency (df): number of documents that a term occurs in
  - **Inverse document frequency (idf):**

$$idf_t = \log(\frac{N}{df_t})$$

  - (N) is the number of documents in the corpus

Higher for terms that occur in fewer documents

BERTopic: Neural topic modeling with a class-based TF-IDF procedure

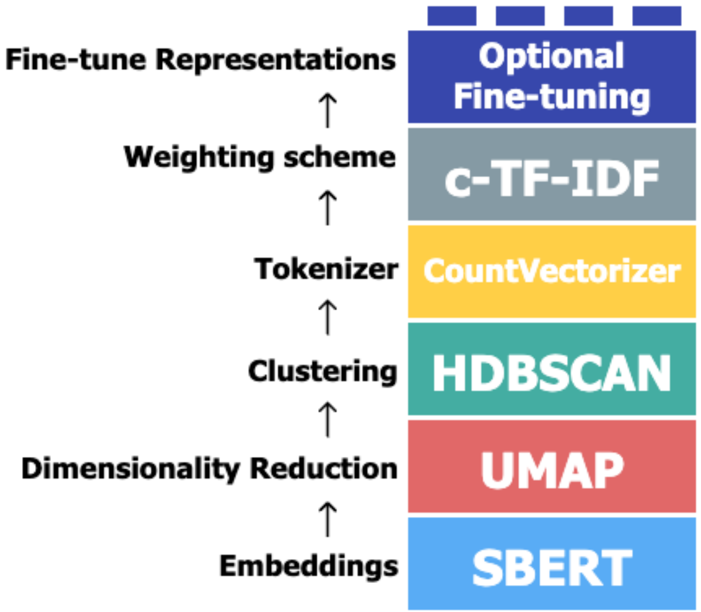
Count of term in the ”class” (or cluster)

Average number of words per class

$$W_{t,c} = tf_{t,c} * \log(1 + \frac{A}{tf_t})$$

Frequency of term across all classes

BERTopic: Neural topic modeling with a class-based TF-IDF procedure



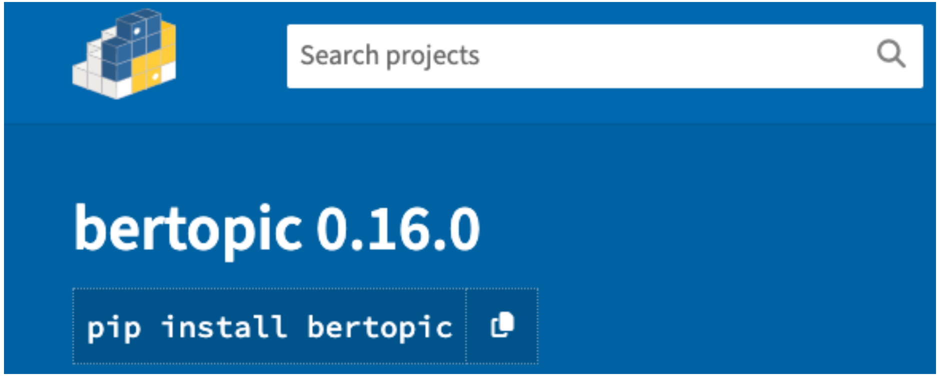
- Variants of computing topic representations (e.g. use GPT to generate human-readable representations)

- Can optionally merge uncommon topics with their most similar ones
- Can compute common words over subsets of a cluster rather than the whole cluster (e.g. divide documents based on time to allow topics to vary over time)

Additional notes

- Automated evaluation for topic *coherence* and *diversity*

- Limitations:
  - *Not* a mixture model – documents get assigned to 1 topic
  - Still using bag-of-words for assigning topic representations (in the original model)
  - What else?



LLM: Prompting

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Background

- So far, we've been talking about how to use pre-trained language models in two primary ways:
  - Fine-tuning them for downstream classification tasks
  - Leveraging pre-trained model characteristics (embeddings, MLM adaptation)

- What about chatbot-style LLMs like GPT? How can they be used for this kind of task?

- First, a little more background on how we build a GPT-style model

Language Models are not trained to do what you want

PROMPT	*Explain the moon landing to a 6 year old in a few sentences.*
COMPLETION	GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

There is a mismatch between LLM pre-training and user intents.

Adapting Language Models

A model that is pre-trained on massive amounts of data cannot do general-purpose tasks without further adaptation—it only complete sentences.

We need a few more steps:

Pre-train → Instruct-tune → RLHF

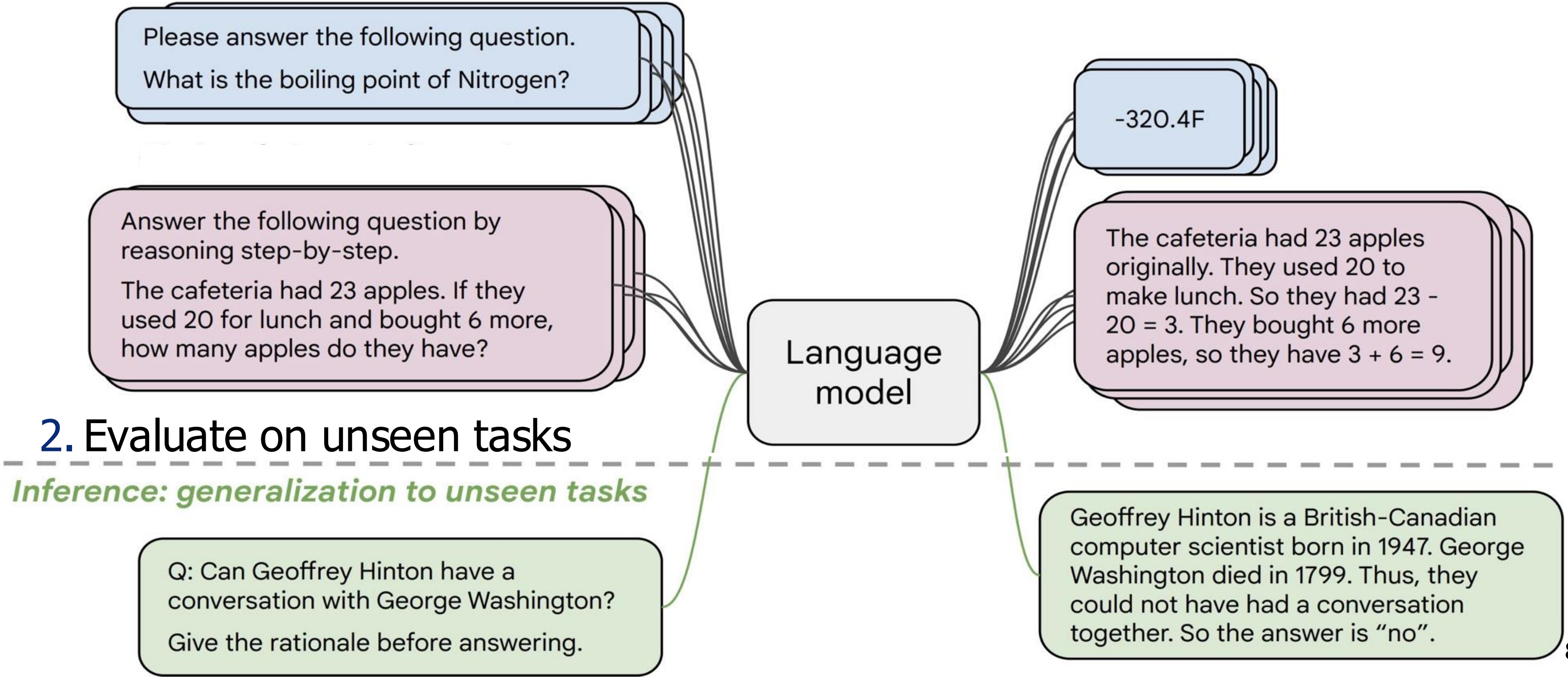
Instruction-tuning

- Finetuning language models on a collection of datasets that involve mapping language instructions to their corresponding desirable generations.

Instruction-tuning

[Weller et al. 2020. Mishra et al. 2021; Wang et al. 2022, Sanh et al. 2022; Wei et al., 2022, Chung et al. 2022, many others]

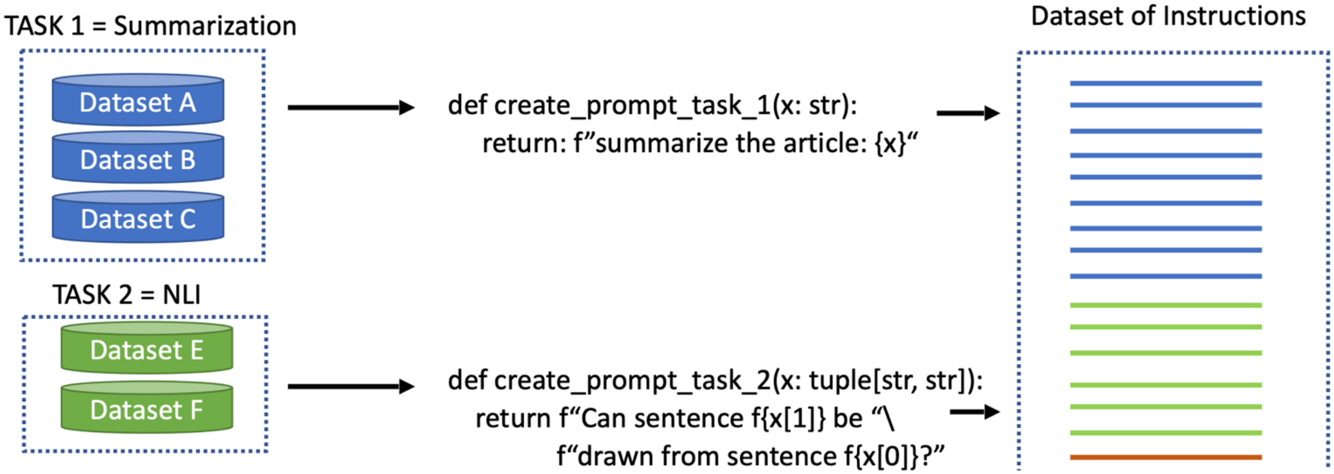
1. Collect examples of (instruction, output) pairs across many tasks and finetune an LM



2. Evaluate on unseen tasks

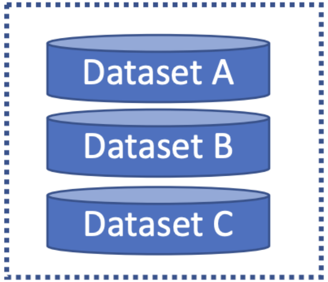
Instruction-tuning: Data

- High quality **labeled** data representing a variety of potential use cases
- Useful trick: we can leverage existing data that was collected for specific tasks



Diversity-inducing via Task Prompts

TASK 1 = Summarization



"Write highlights for this article:\n\n{text}\n\nHighlights: {highlights}"
"Write a summary for the following article:\n\n{text}\n\nSummary: {highlights}"
"{text}\n\nWrite highlights for this article. {highlights}"
"{text}\n\nWhat are highlight points for this article? {highlights}"
"{text}\nSummarize the highlights of this article. {highlights}"
"{text}\nWhat are the important parts of this article? {highlights}"
"{text}\nHere is a summary of the highlights for this article: {highlights}"
"Write an article using the following points:\n\n{highlights}\n\nArticle: {text}"
"Use the following highlights to write an article:\n\n{highlights}\n\nArticle:{text}"
"{highlights}\n\nWrite an article based on these highlights. {text}"

Summary Thus Far

- **Instruction-tuning:** Training LMs with annotated input instructions and their output.
  - Improves performance of LM's zero-shot ability in following instructions.
  - Scaling the instruction tuning data size improves performance.
  - Diversity of prompts is crucial.
  - Compared with pretraining, instruction tuning has a minor cost (Typically consumes <1% of the total training budget)
- Cons:
  - It's expensive to collect ground-truth data for tasks.
  - This is particularly difficult for open-ended creative generation have no right answer.
  - Prone to hallucinations.

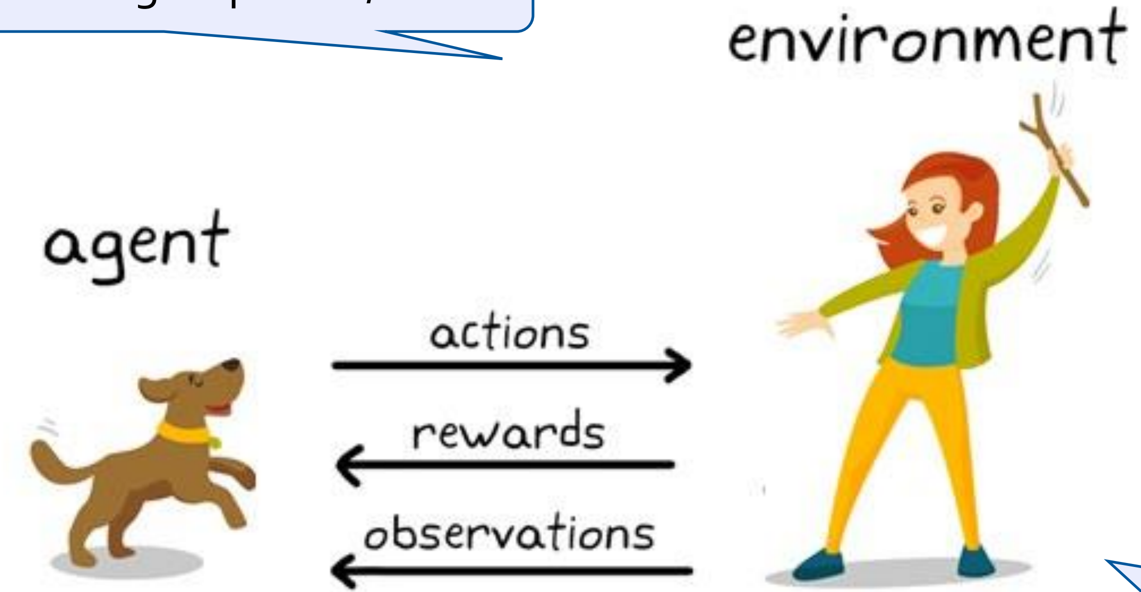
[Weller et al. 2020. Mishra et al. 2021; Wang et al. 2022, Sanh et al. 2022; Wei et al., 2022, Chung et al. 2022, many others]

Reinforcement Learning from Human Feedback

Pre-train → Instruct-tune → RLHF

Reinforcement Learning: Intuition

Action here: generating responses/token



Reward here: whether humans liked the generation (sequence of actions=tokens)

[figure credit]

Intuition

Human
I have to come up with a conversation with an AI assistant for the purposes of showing it off in a demo. What sorts of conversation topics would you recommend?

Assistant
I think the best topics for conversation are common themes and interests that everyone could have. Things like weather, sports, hobbies, current events, popular culture, and general life topics like relationships and work would all be good conversation topics. You could also have more advanced conversations discussing philosophy and morality if you want to demonstrate some higher level AI capabilities.

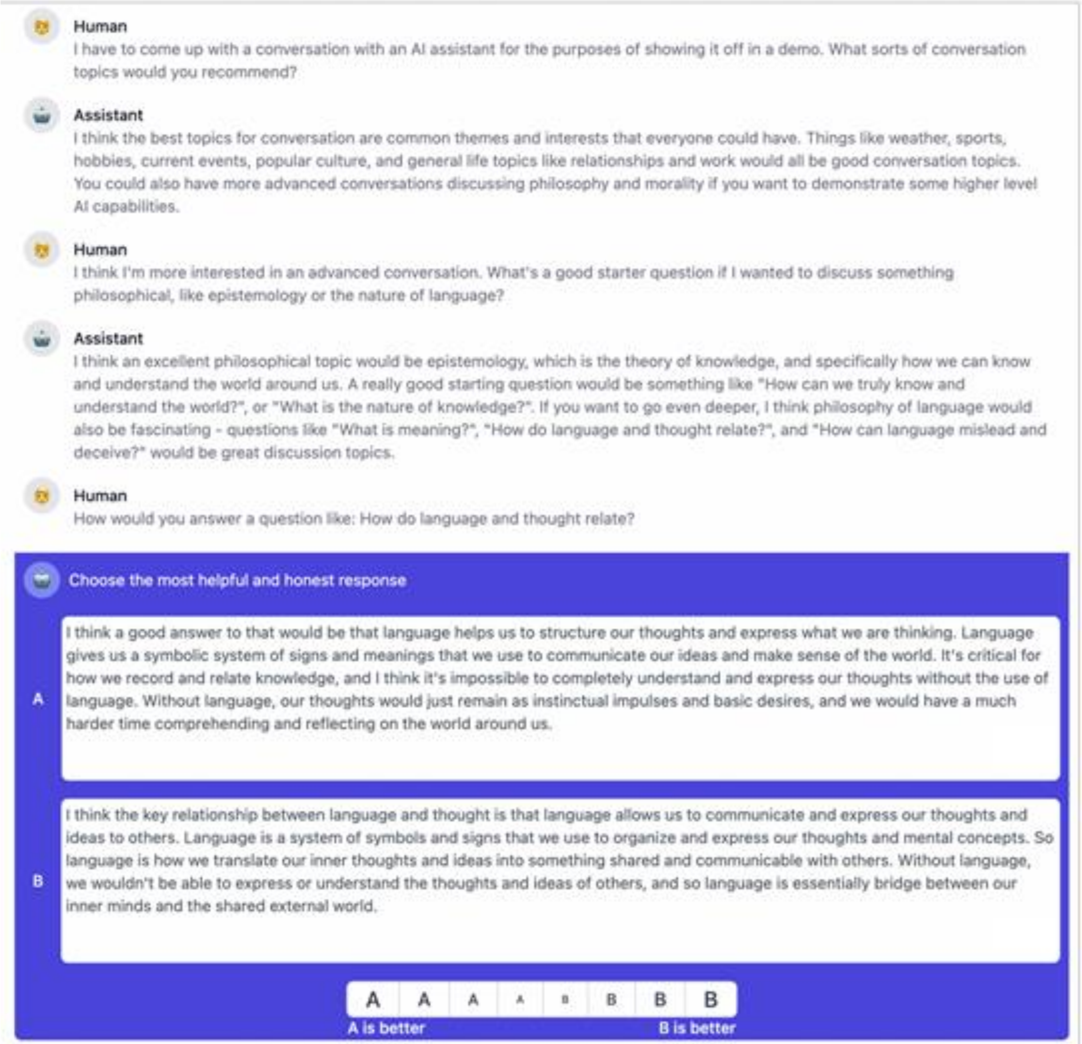
Human
I think I'm more interested in an advanced conversation. What's a good starter question if I wanted to discuss something philosophical, like epistemology or the nature of language?

Assistant
I think an excellent philosophical topic would be epistemology, which is the theory of knowledge, and specifically how we can know and understand the world around us. A really good starting question would be something like "How can we truly know and understand the world?", or "What is the nature of knowledge?". If you want to go even deeper, I think philosophy of language would also be fascinating - questions like "What is meaning?", "How do language and thought relate?", and "How can language mislead and deceive?" would be great discussion topics.

human has conversation with the LLM

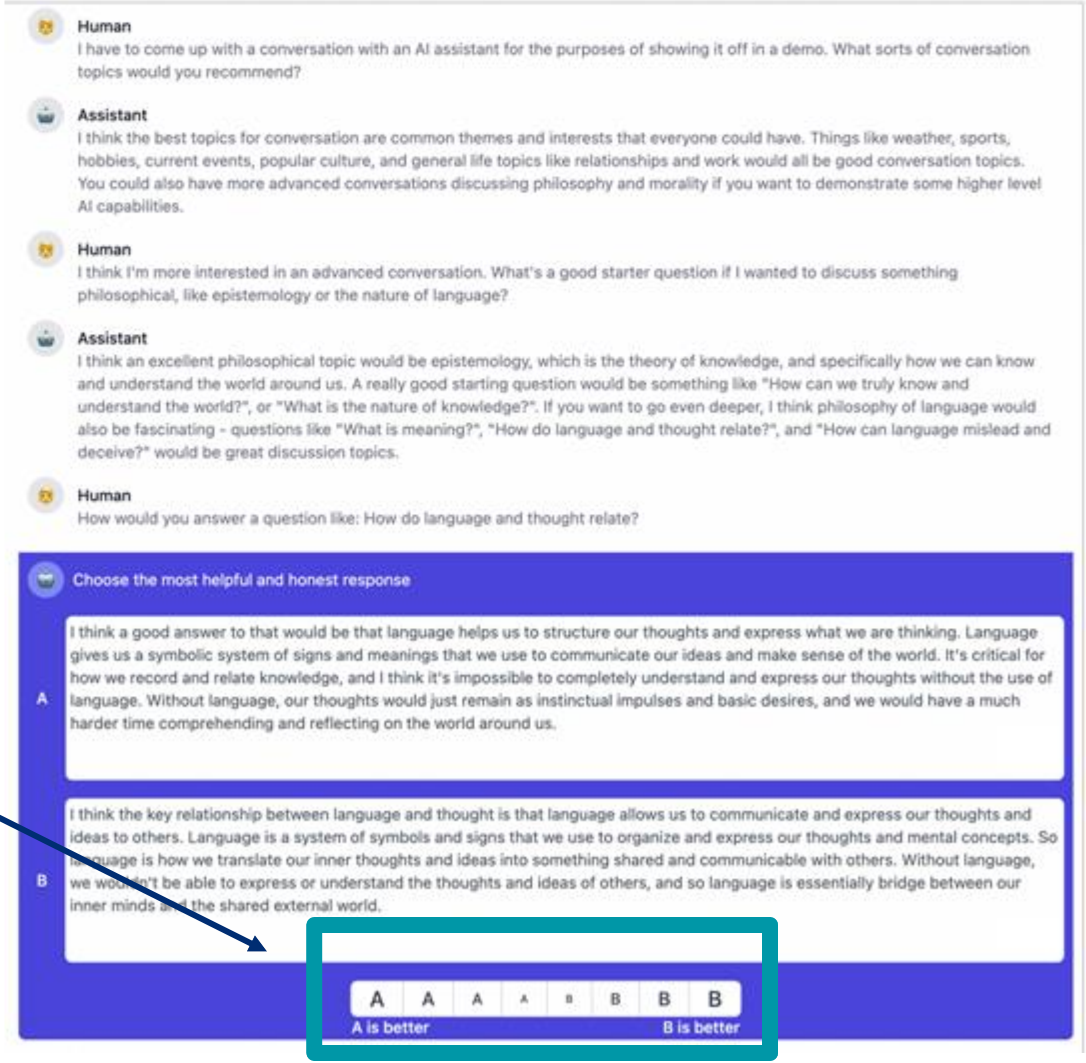
Intuition

LLM provides two options for next responses

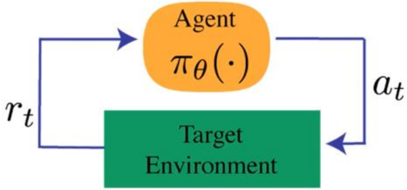


Intuition

human rates better response (Likert scale or rankings)



Reinforcement Learning from Human Feedback



- We can't use actual human feedback all of the time – too expensive!
- Instead, define a reward function: $R(s; \text{prompt}) \in \mathbb{R}$ for any output s to a prompt, where the reward is higher when humans prefer the output
- Good generation is equivalent to finding reward-maximizing outputs:
  - $\mathbb{E}_{\hat{s} \sim p_\theta}[R(\hat{s}; \text{prompt})]$
- What we need to do:
  - (1) Estimate the reward function $R(s; \text{prompt})$.
  - (2) Find the best generative model p_θ that maximizes the expected reward:

$$\hat{\theta} = \text{argmax}_\theta \mathbb{E}_{\hat{s} \sim p_\theta}[R(\hat{s}; \text{prompt})]$$

[Slide credit: Jesse Mu]

Johns Hopkins
Whiting School of Engineering

Prompting+Topic Model: TopicGPT

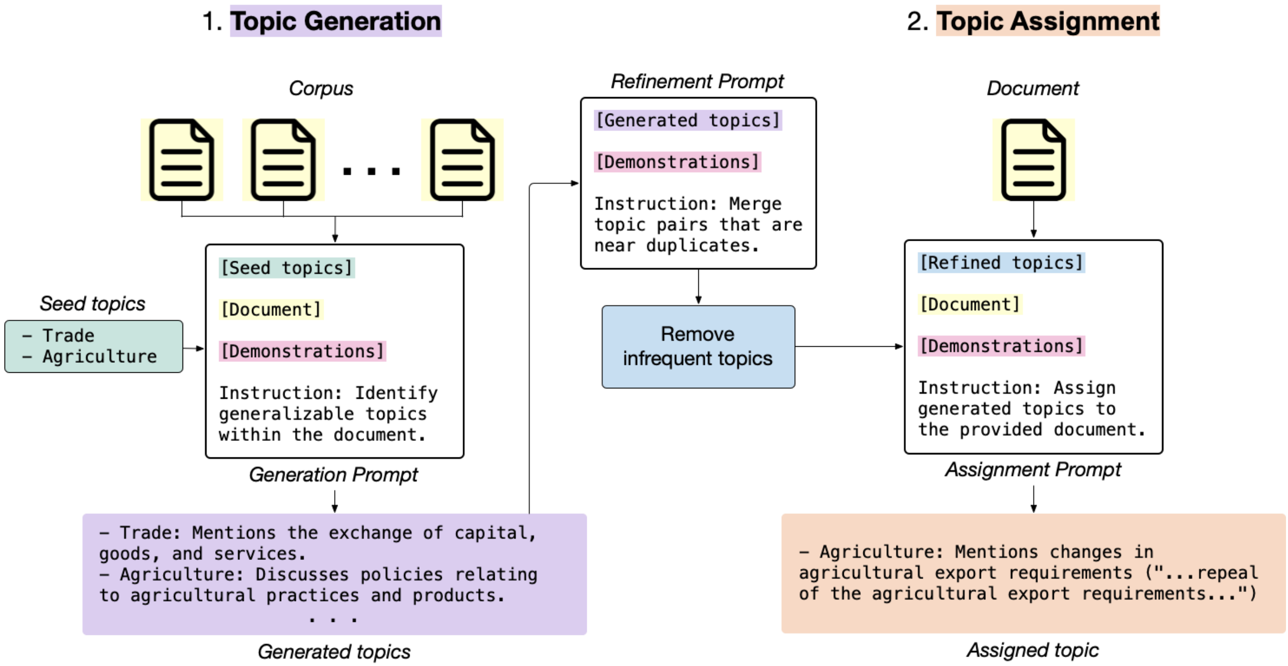
LLM-based ”topic models”

- Now we’ve built an LLM where we can give it arbitrary instructions and it’s potentially pretty good at following them

- How can we use this for “topic modeling” (or more generally, open-ended corpus analysis)?

- Two examples of LLM-based topic models: TopicGPT, LLooM

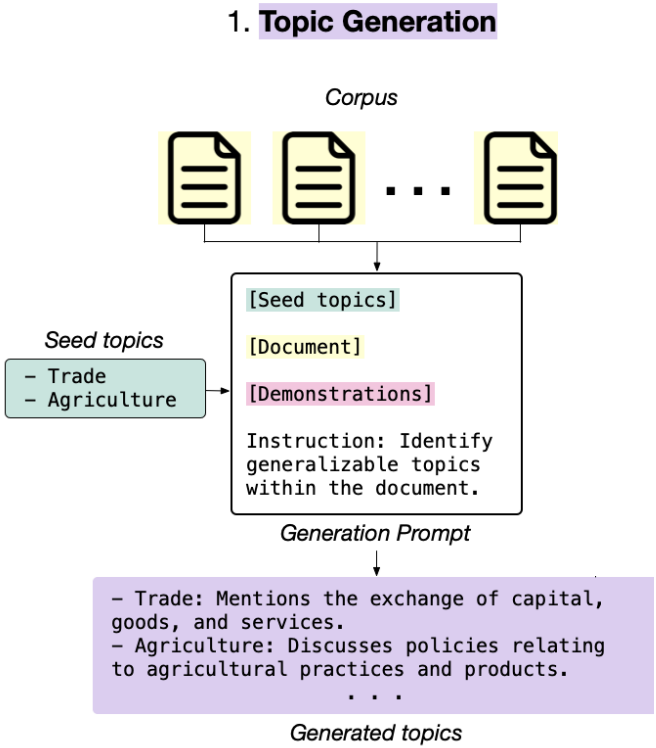
TopicGPT



Johns Hopkins Whiting School of Engineering

Pham, Chau Minh, et al. "TopicGPT: A prompt-based topic modeling framework." NAACL (2024).
https://arxiv.org/abs/2311.01449

TopicGPT: Generate Topics (Phase 1)



- Provide to AI model (GPT-4):
  - Seed topics (concise label and broad 1 sentence description)
  - Document *d*
- Prompt model to generate a topic assignment for *d*, either from the existing topics or generate a new one
- Conducted over a sample of documents from the corpus

TopicGPT: Refine Topics (Phase 1.5)

- Merge topics [Optional]
  - Provide model pairs of similar topics (determined using embedding similarity)
  - Prompt model to merge similar pairs

- Reduce topics
  - Drop topics with infrequent assignments

- Generate topic hierarchy
  - Provide the model with top level topic, the documents associated with the top-level topic t, and a list of seed subtopics S'
  - Instruct the LLM to generate subtopics that capture common themes among the provided documents.

TopicGPT: Assign Topics (Phase 2)

- Prompt model to assign a topic to a document given
  - Generated topics from step 1
  - 2-3 examples
  - The document

- Final output:
  - Assigned topic label
  - Document-specific topic description
  - Quote extracted from the document to support this assignment

- [Self-correction step to eliminated hallucinated topics or None/Error outputs]

Prompts are long and complicated

Prompt template for generating first-level/flat topics

You will receive a document and a set of top-level topics from a topic hierarchy. Your task is to identify generalizable topics within the document that can act as top-level topics in the hierarchy. If any relevant topics are missing from the provided set, please add them. Otherwise, output the existing top-level topics as identified in the document.

[Top-level topics]
{Example topics (containing "[1] Trade" in this example)}

[Examples]
Example 1: Adding "[1] Agriculture"
Document:
Saving Essential American Sailors Act or SEAS Act - Amends the Moving Ahead for Progress in the 21st Century Act (MAP-21) to repeal the Act's repeal of the agricultural export requirements that: (1) 25 of the gross tonnage of certain agricultural commodities or their products exported each fiscal year be transported on U.S. commercial vessels, and (2) the Secretary of Transportation (DOT) finance any increased ocean freight charges incurred in the transportation of such items. Revives and reinstates those repealed requirements to read as if they were never repealed.

Your response:
[1] Agriculture: Mentions policies relating to agricultural practices and products.

Example 2: Duplicate "[1] Trade", returning the existing topic
Document:
Amends the Harmonized Tariff Schedule of the United States to suspend temporarily the duty on mixtures containing Fluopyram.

Your response:
[1] Trade: Mentions the exchange of capital, goods, and services.

[Instructions]
Step 1: Determine topics mentioned in the document.
- The topic labels must be as GENERALIZABLE as possible. They must not be document-specific.
- The topics must reflect a SINGLE topic instead of a combination of topics.
- The new topics must have a level number, a short general label, and a topic description.
- The topics must be broad enough to accommodate future subtopics.
Step 2: Perform ONE of the following operations:
1. If there are already duplicates or relevant topics in the hierarchy, output those topics and stop here.
2. If the document contains no topic, return "None".
3. Otherwise, add your topic as a top-level topic. Stop here and output the added topic(s). DO NOT add any additional levels.

[Document]
{Document}

Please ONLY return the relevant or modified topics at the top level in the hierarchy.
[Your response]

Evaluation

- Topic *Alignment*
  - Use corpora with human-assigned labels
  - Assign each document to a single most-probable topic
  - Standard metrics for evaluating *cluster assignment* (this pays no attention to the label of the cluster)
    - Purity, Inverse Purity, Adjusted Rand Index, Normalized Mutual Information
- Topic *Stability*
  - Robustness to changes in prompts, different seed topics, etc
- Human evaluation of topic semantics

Evaluation

Dataset	Setting	TopicGPT			LDA			BERTopic		
		P_1	ARI	NMI	P_1	ARI	NMI	P_1	ARI	NMI
`Wiki`	Default setting ($k = 31$)	**0.73**	**0.58**	**0.71**	0.59	0.44	0.65	0.54	0.24	0.50
	Refined topics ($k = 22$)	**0.74**	**0.60**	**0.70**	0.64	0.52	0.67	0.58	0.28	0.50
`Bills`	Default setting ($k = 79$)	**0.57**	**0.42**	**0.52**	0.39	0.21	0.47	0.42	0.10	0.40
	Refined topics ($k = 24$)	**0.57**	**0.40**	**0.49**	0.52	0.32	0.46	0.39	0.12	0.34
	TopicGPT stability ablations, baselines controlled to have the same number of topics (k).									
	Different generation sample ($k = 73$)	**0.57**	**0.40**	**0.51**	0.41	0.23	0.47	0.38	0.08	0.38
	Out-of-domain prompts ($k = 147$)	**0.55**	**0.39**	**0.51**	0.31	0.14	0.47	0.35	0.07	0.41
`Bills`	Additional seed topics ($k = 123$)	**0.50**	**0.33**	**0.49**	0.33	0.15	0.46	0.36	0.07	0.40
	Shuffled generation sample ($k = 118$)	**0.55**	**0.40**	**0.52**	0.33	0.16	0.47	0.36	0.08	0.40
	Assigning with Mistral ($k = 79$)	**0.51**	**0.37**	**0.46**	0.39	0.21	0.47	0.42	0.10	0.40

Table 2: Topical alignment between ground-truth labels and predicted assignments. Overall, TopicGPT achieves the best performance across all settings and metrics compared to LDA and BERTopic. The number of topics used in each setting is specified as k. The largest values in each metric and setting are **bolded**.

Evaluation

- How do we evaluate:
    - Actual topic assignments?
    - Comprehensiveness of generated topics?

Evaluation

- Hand-annotated topics in comparison to ground truth:
  - Out-of-scope topics: topics that are too narrow or too broad compared to the associated ground truth topic.
  - Missing topics: topics present in the ground truth but not in the generated outputs.
  - Repeated topics: topics that are duplicates of other topics.

Dataset	Setting	Out-of-scope	Missing	Repeated	Total
Wiki	LDA ($k = 31$)	46.3	4.3	11.9	62.4
	Unrefined ($k = 31$)	38.7	**0.0**	1.1	39.8
	Refined ($k = 22$)	**30.3**	**0.0**	**0.0**	**30.3**
Bills	LDA ($k = 79$)	56.1	2.1	22.0	80.2
	Unrefined ($k = 79$)	65.0	**1.3**	3.8	70.1
	Refined ($k = 24$)	**27.8**	4.2	**0.0**	**31.9**

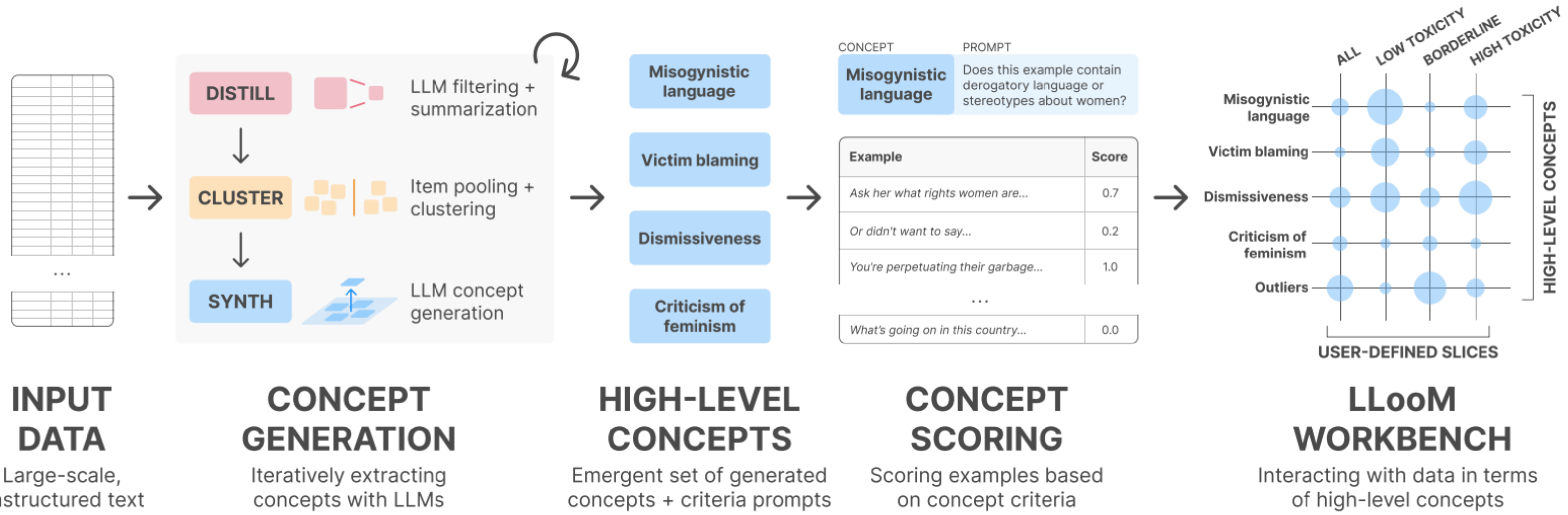
Limitations

- Evaluation is still difficult:
  - Do any of these metrics check if documents were assigned to the correct topic?
  - How do we evaluate multi-topic assignment?

- Need to provide seed topics

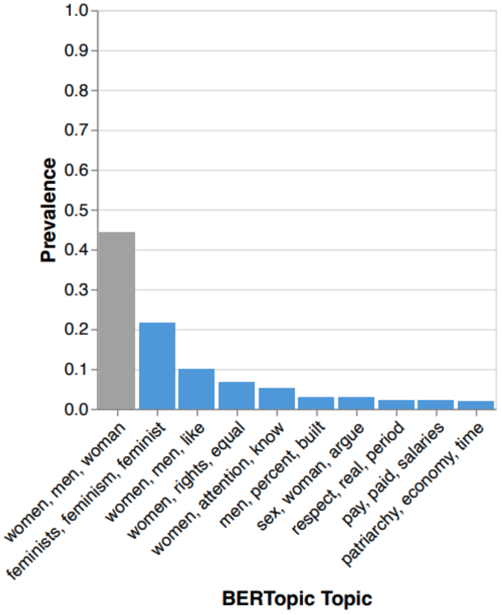
- Reliance on closed-source LLMs (paid APIs)
  - Open-source models are less good at topic generation in particular (they use GPT-4 for generation and GPT-3.5 for assignment)

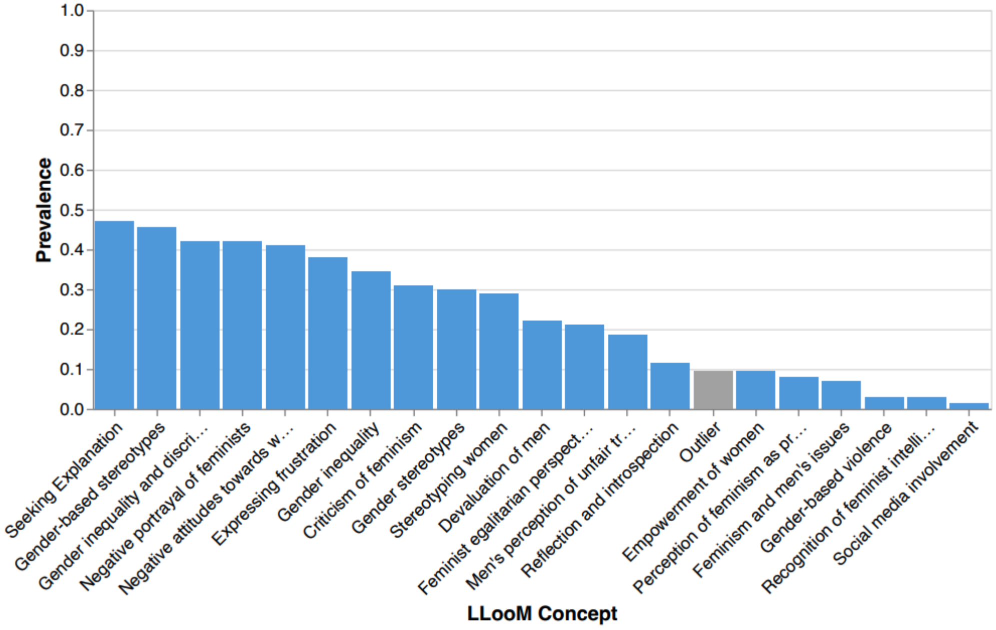
A different approach: LLooM



Lam, Michelle S., et al. "Concept induction: Analyzing unstructured text with high-level concepts using lloom." *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 2024.

Example evaluation





Comparison with TopicGPT

- Overall pipeline has some differences
- Lots of focus on usable interface with less model abalations and changes in pipeline (HCI vs. NLP)
- Which is better?

Recap

- Neural LDA (ProdLDA, CTM)
- Instruction Tuning and Alignment
- Beyond LDA (BERTopic, TopicGPT)

- Next class:
  - Prompting approaches