Explaining Datasets in Words: Statistical Models with Natural Language Parameters

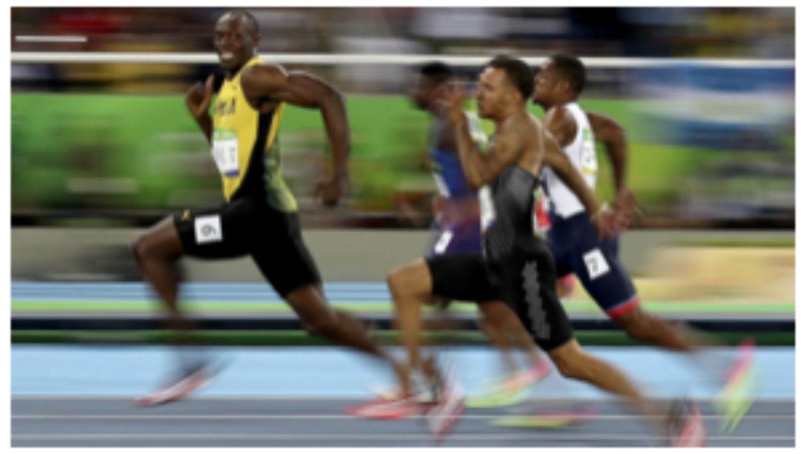
Ruiqi Zhong, Heng Wang, Dan Klein, Jacob Steinhardt

Motivation

- Imagine the following scenario: It is the year 3024. We are historians trying to understand what happened between 2016 and 2024, by looking at how Twitter topics changed across that time period.

Motivation

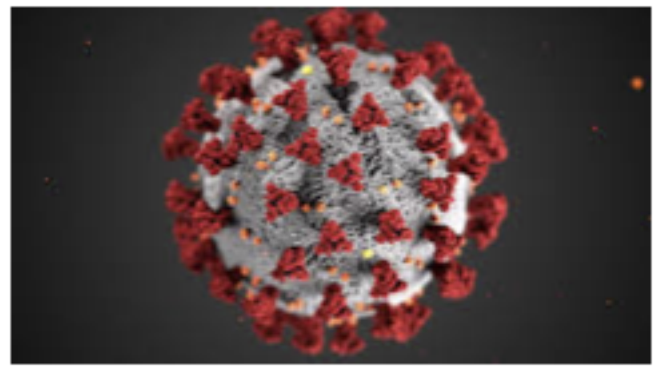
- Imagine the following scenario: It is the year 3024. We are historians trying to understand what happened between 2016 and 2024, by looking at how Twitter topics changed across that time period.
- We are given a dataset of user-posted images sorted by time, and our goal is to find trends in this dataset to help interpret what happened.



x_{2016}

...

x_{2019}



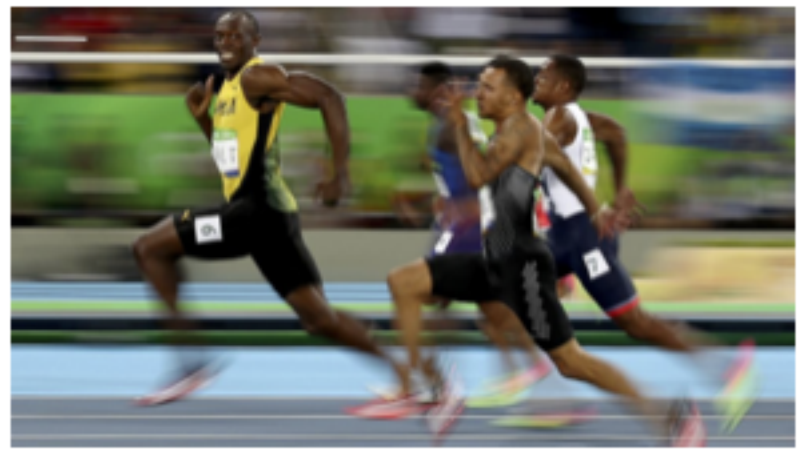
x_{2020}

...

x_{2024}

Motivation

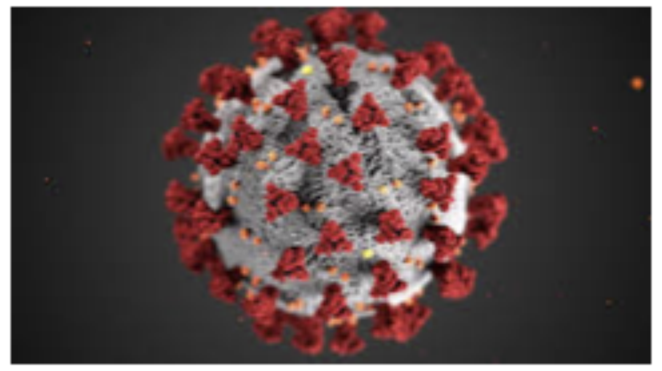
- If we successfully achieve our goal, we would discover, for instance, (1) a recurring spike of images depicting athletes every four years for the Olympics, and (2) a large increase in images containing medical concepts during and after the COVID-19 pandemic.



x_{2016}

...

x_{2019}



x_{2020}

...

x_{2024}

Motivation

- One common approach is to fit a time series model to predict how the features evolve and then interpret the learned model.
- However, the parameters are high-dimensional and uninterpretable, undermining the goal of extracting explainable trends.

Motivation

- One common approach is to fit a time series model to predict how the features evolve and then interpret the learned model.
- However, the parameters are high-dimensional and uninterpretable, undermining the goal of extracting explainable trends.
- We address this with a framework for statistical modeling where parameters are expressed as *natural language predicates* rather than abstract numerical values

Framework

- The foundation of the framework rests on:

$$p(x|\vec{\phi}, w) \propto e^{w^T[\vec{\phi}](x)}$$

- Learning two sets of latent parameters: *natural language parameters* **ϕ** (the learned features) and real-valued parameters **w**

Framework

- The foundation of the framework rests on:

$$p(x|\vec{\phi}, w) \propto e^{w^T [\vec{\phi}](x)}$$

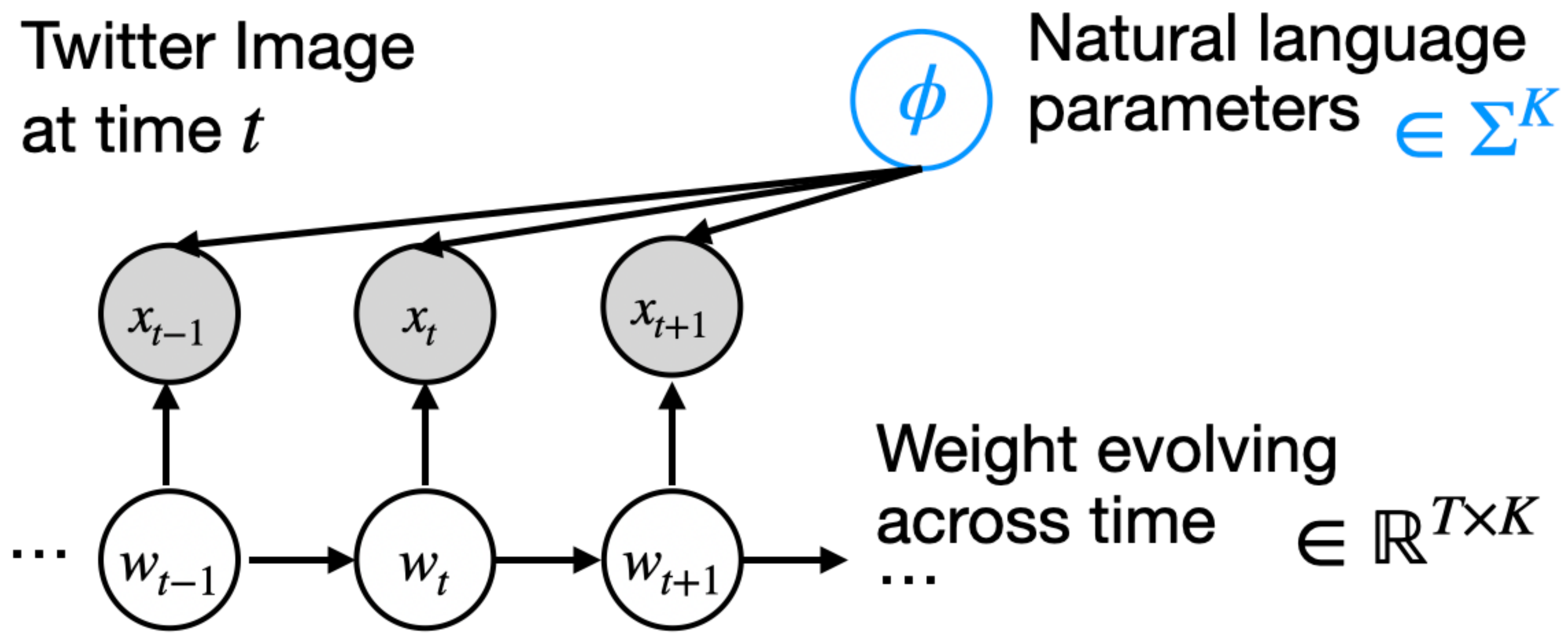
- Learning two sets of latent parameters: *natural language parameters* ϕ (the learned features) and real-valued parameters **w**

- ϕ are natural language predicts (e.g. "asks about the U.S. Election" or "discusses pandemics.'')

- [[ϕ]](x) denotes the binary evaluation of whether predicate ϕ is true for sample x

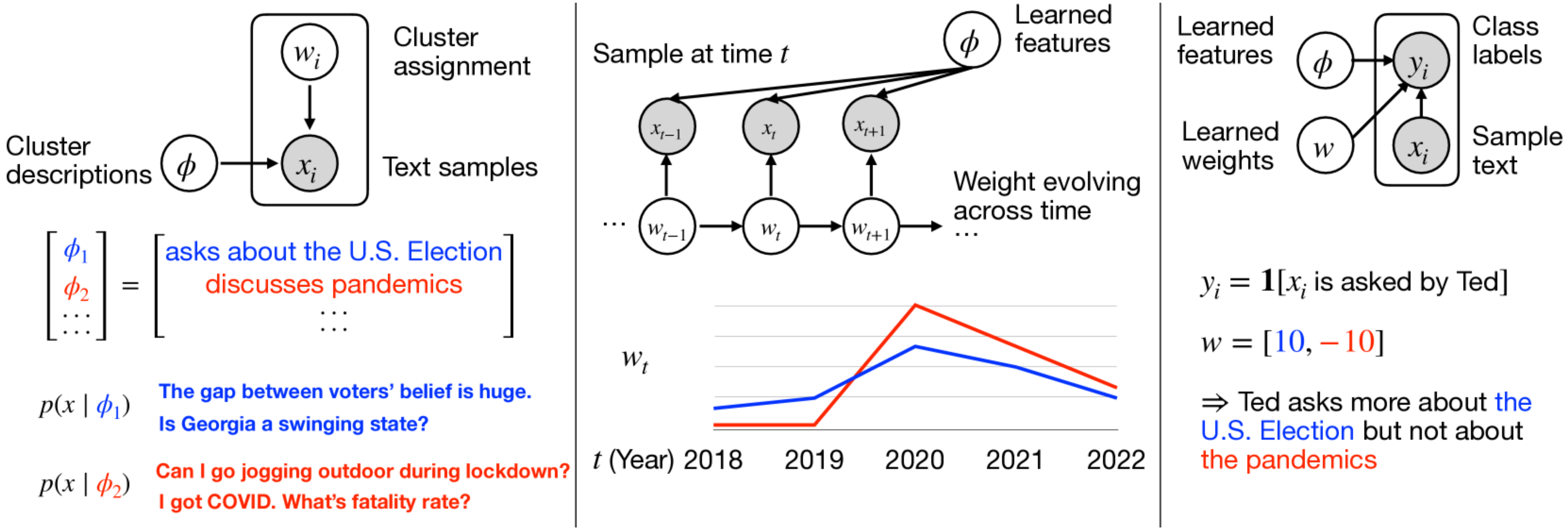
Framework

- In the time-series case, $p(x|\vec{\phi}, w) \propto e^{w^T[\vec{\phi}](x)}$
  - **ϕ**: the natural language descriptions of K different topics, e.g. "depicts athletes competing''
  - **w_t**: the frequency of each of the K topics at the time t



Framework

- The framework can also be applied to other statistical models, such as clustering and classification



Learning the Natural Language Parameters

- The central technical challenge involves optimizing discrete natural language parameters that cannot be directly differentiated
- Developed a three-step algorithmic approach

Learning the Natural Language Parameters

- Step 1: Continuous Relaxation
  - Each discrete predicate ϕ_k is relaxed to a continuous vector $\tilde{\phi}_k$ in embedding space. The binary evaluation $[[\phi_k]](x)$ is approximated by the dot product $\tilde{\phi}_k \cdot e_x$, where e_x is the neural embedding of x

Learning the Natural Language Parameters

- Step 1: Continuous Relaxation
  - Each discrete predicate ϕ_k is relaxed to a continuous vector $\tilde{\phi}_k$ in embedding space. The binary evaluation $[[\phi_k]](x)$ is approximated by the dot product $\tilde{\phi}_k \cdot e_x$, where e_x is the neural embedding of x
- Step 2: Gradient-Based Optimization
  - With continuous relaxation, the objective becomes differentiable, enabling standard gradient descent optimization of $\tilde{\phi}_k$

Learning the Natural Language Parameters

- Step 3: Discretization
  - Sample texts are ranked by their dot product scores with $\tilde{\phi}_k$
  - A language model generates candidate predicates explaining the ranked patterns
  - Candidates are re-ranked based on correlation with the continuous representation

Discretization: Discretize($\tilde{\phi}$)

Here is a corpus of text samples, sorted from the lowest to the highest score.

Sample 0. “athlete demonstrated remarkable prowess.” (score: -0.2)
Sample 1. “see the player?” (score: -0.3)
…
… $x \sim U(x)$ $\cos(e_x, \tilde{\phi})$
Sample 9. “Wonderful painting …” (score: 0.4)

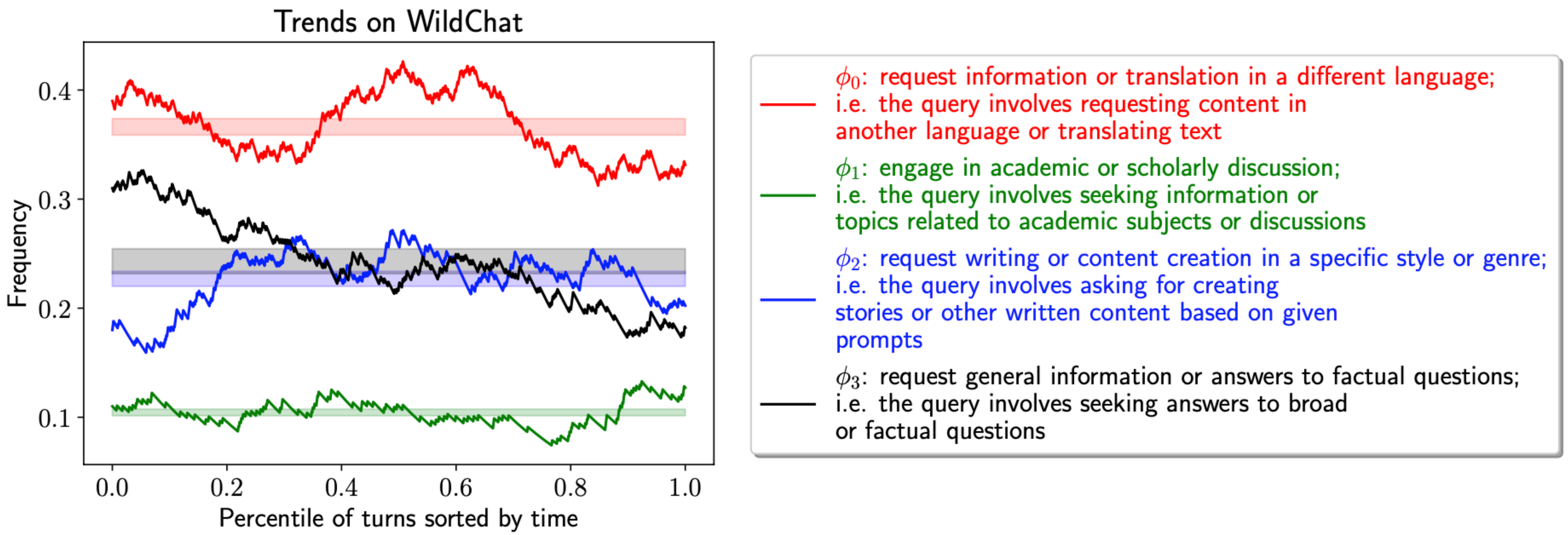
Please suggest predicates about the text samples that are more likely to achieve higher scores.

Your responses are:

- **“has a topic of art”**
- **“has a topic of sports”**
- ….

Borad Applications

- Application 1: Temporal Trend Analysis
  - For instance, given user queries to LLMs (e.g. ChatGPT), we can use our time series model defined above to identify trends in user queries.



Borad Applications

- Application 2: Taxonomizing text via clustering
  - For instance, when applied to chat dialogue, the method generates clear taxonomies of user intents, producing interpretable categories like "*requesting graphic design prompts*" rather than ambiguous word lists from traditional topic models.