Carnegie Mellon University
School of Computer Science

More is Less?
Non-parametric Language Models and Efficiency

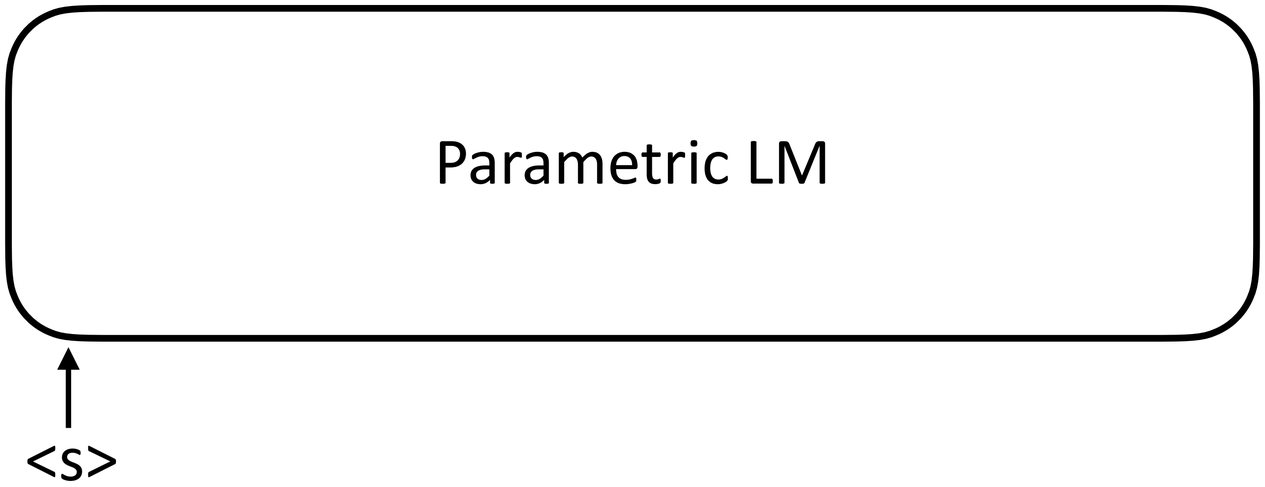
Graham Neubig
Carnegie Mellon University

Based on Work by Junxian He w/ Taylor Berg-Kirkpatrick

Language
Technologies
Institute

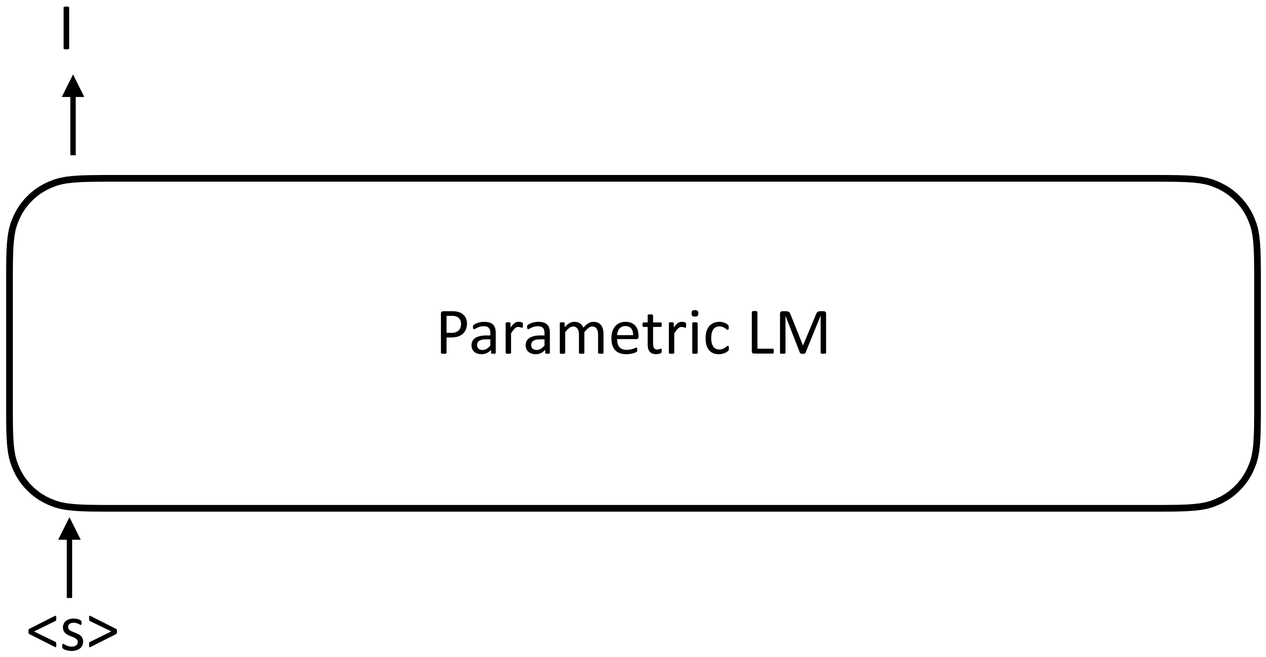
Language
Technologies
Institute

Parametric Language Models



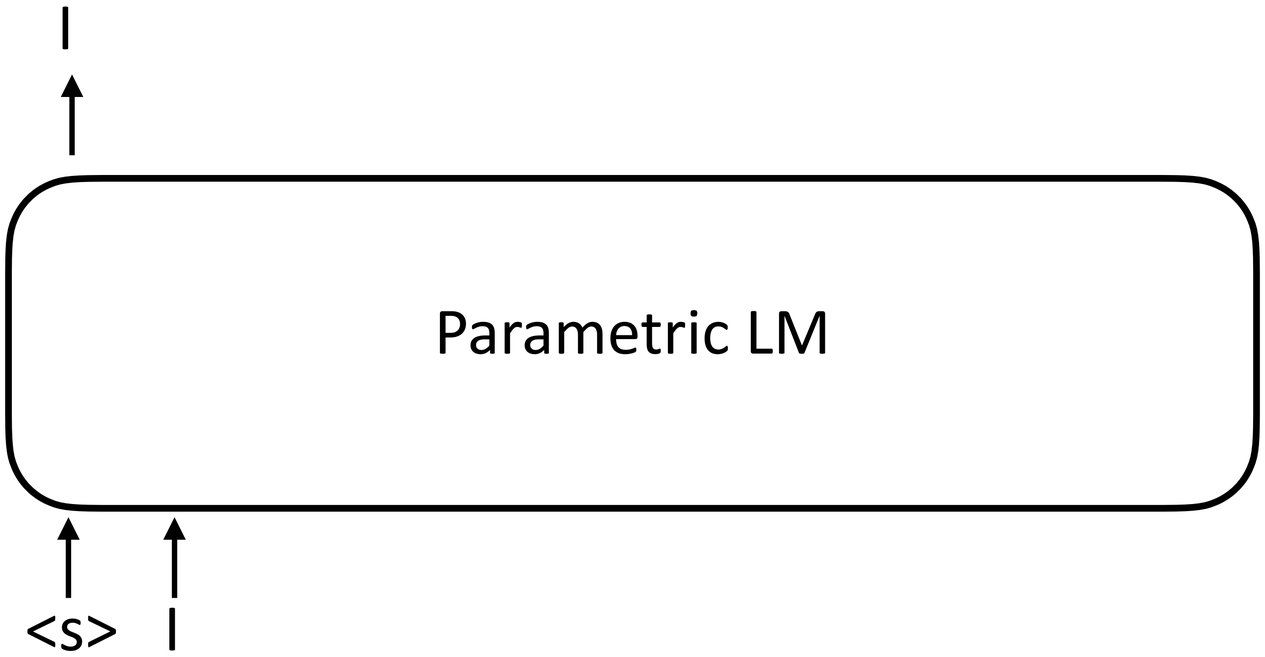
Language
Technologies
Institute

Parametric Language Models



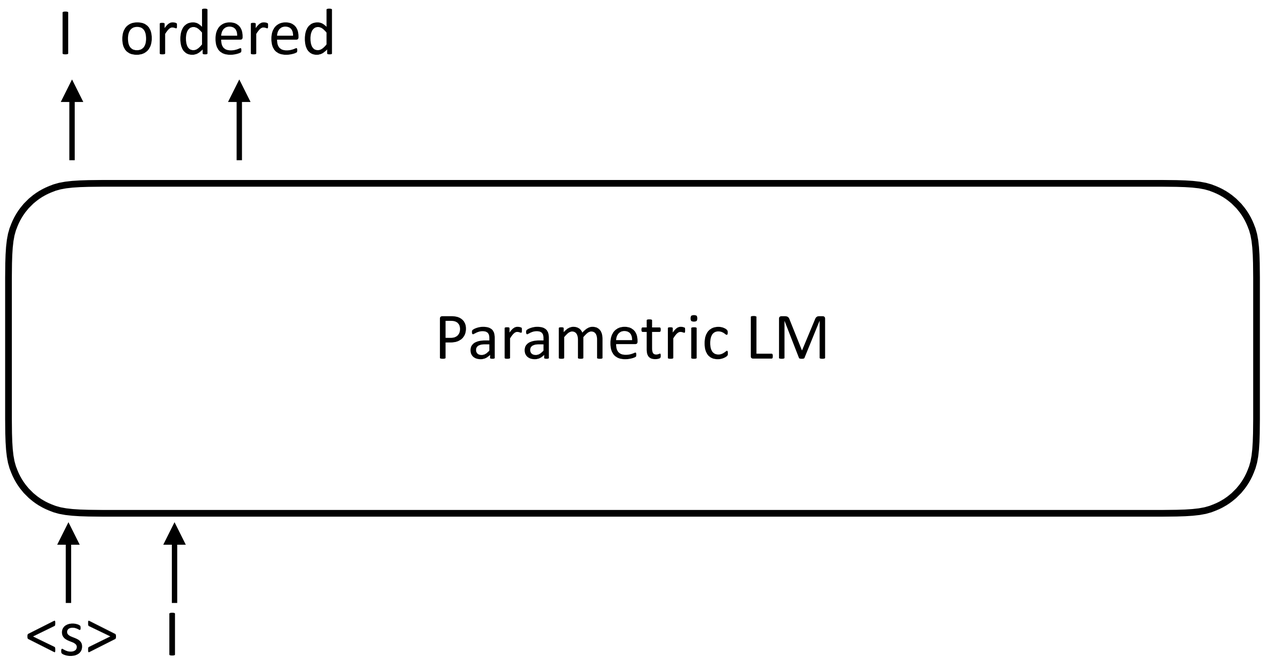
Language
Technologies
Institute

Parametric Language Models



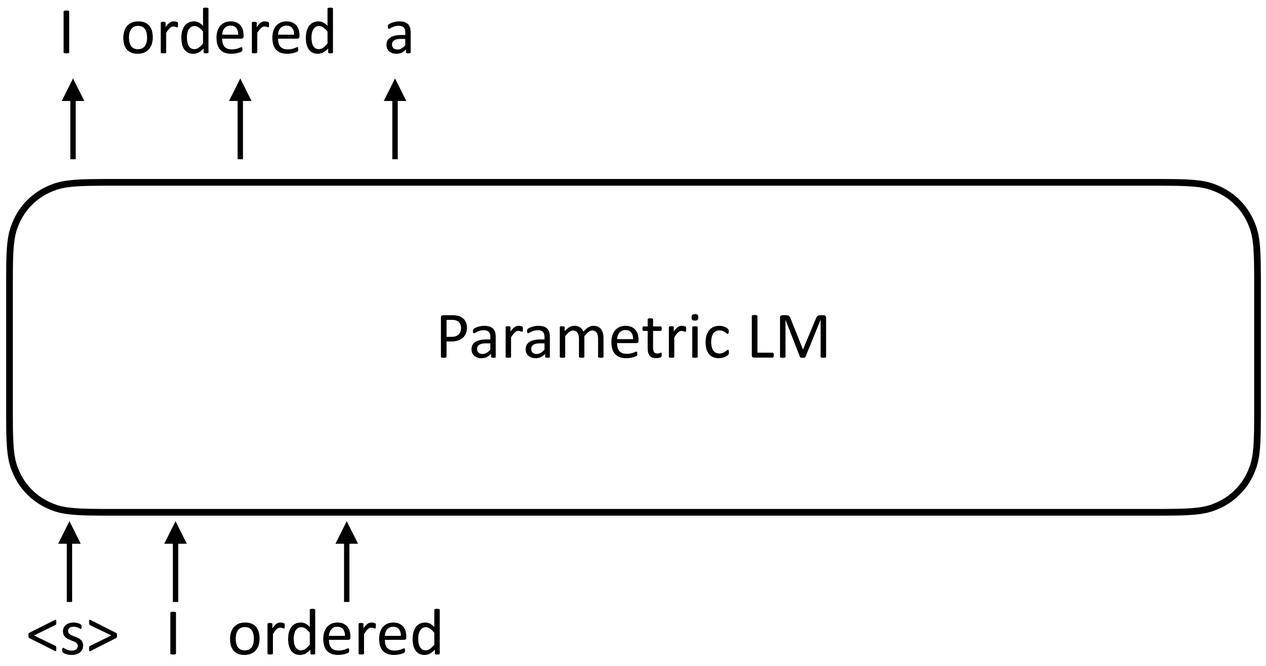
Language
Technologies
Institute

Parametric Language Models



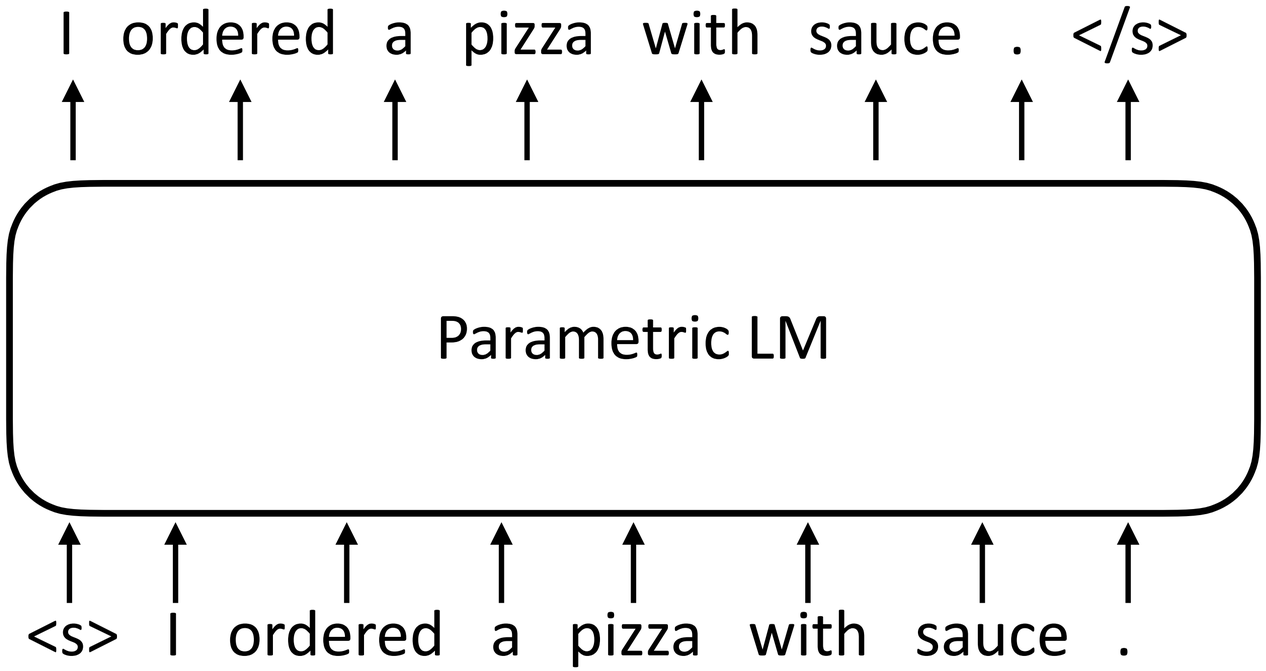
Language
Technologies
Institute

Parametric Language Models



Language
Technologies
Institute

Parametric Language Models



Language Technologies Institute

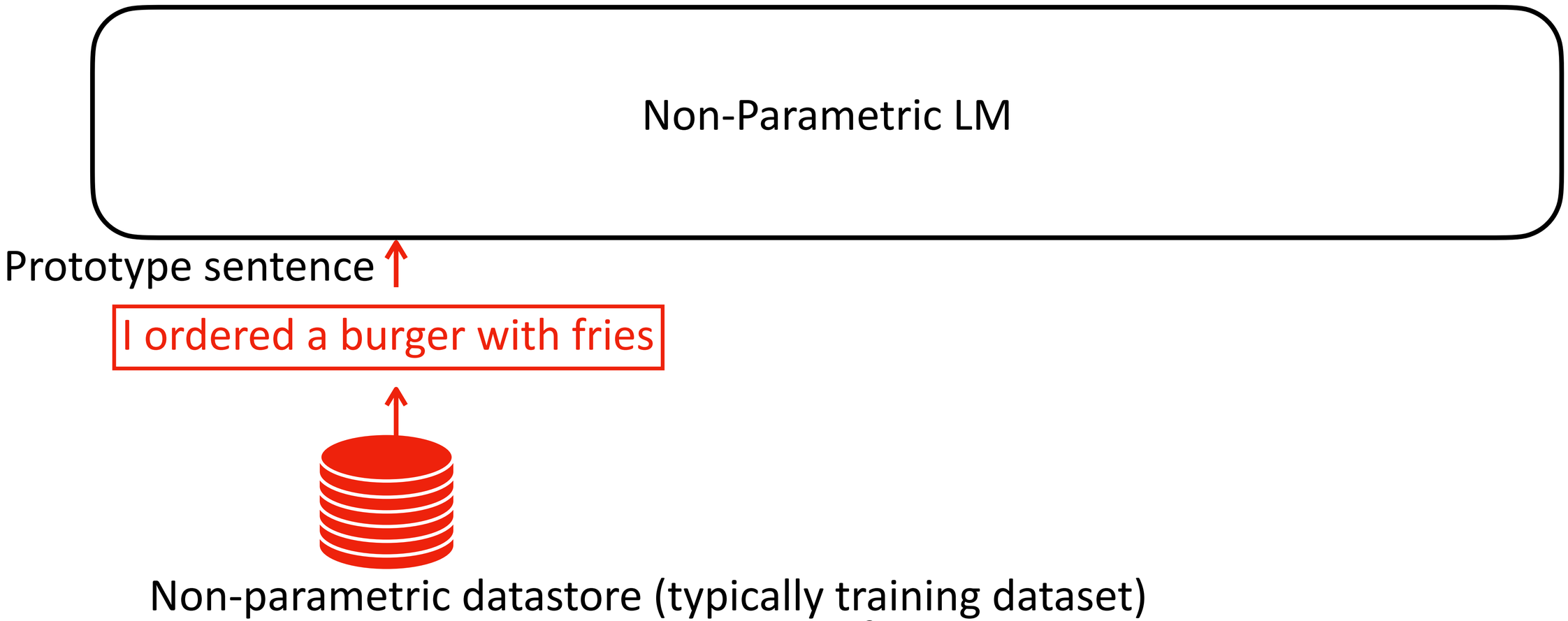
Non-Parametric Language Models

Non-Parametric LM

Non-parametric datastore (typically training dataset)

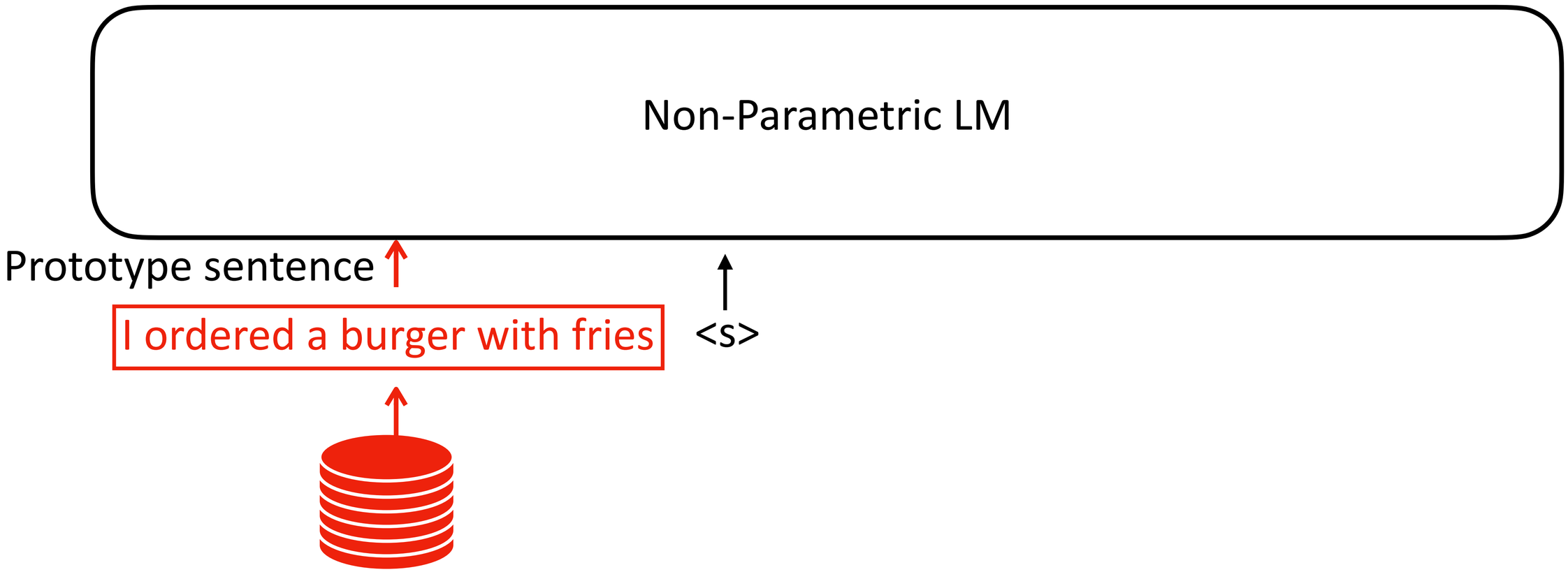
Language Technologies Institute

Non-Parametric Language Models



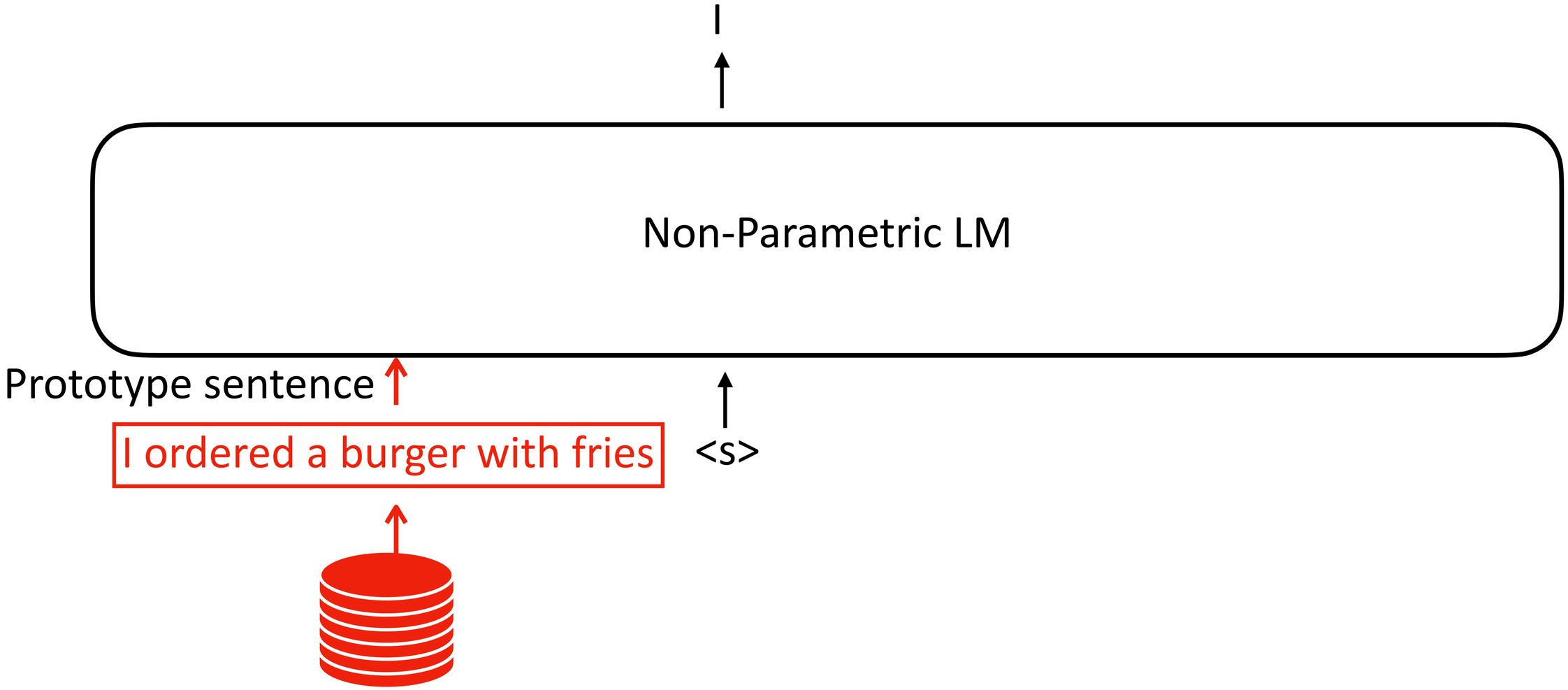
Language Technologies Institute

Non-Parametric Language Models



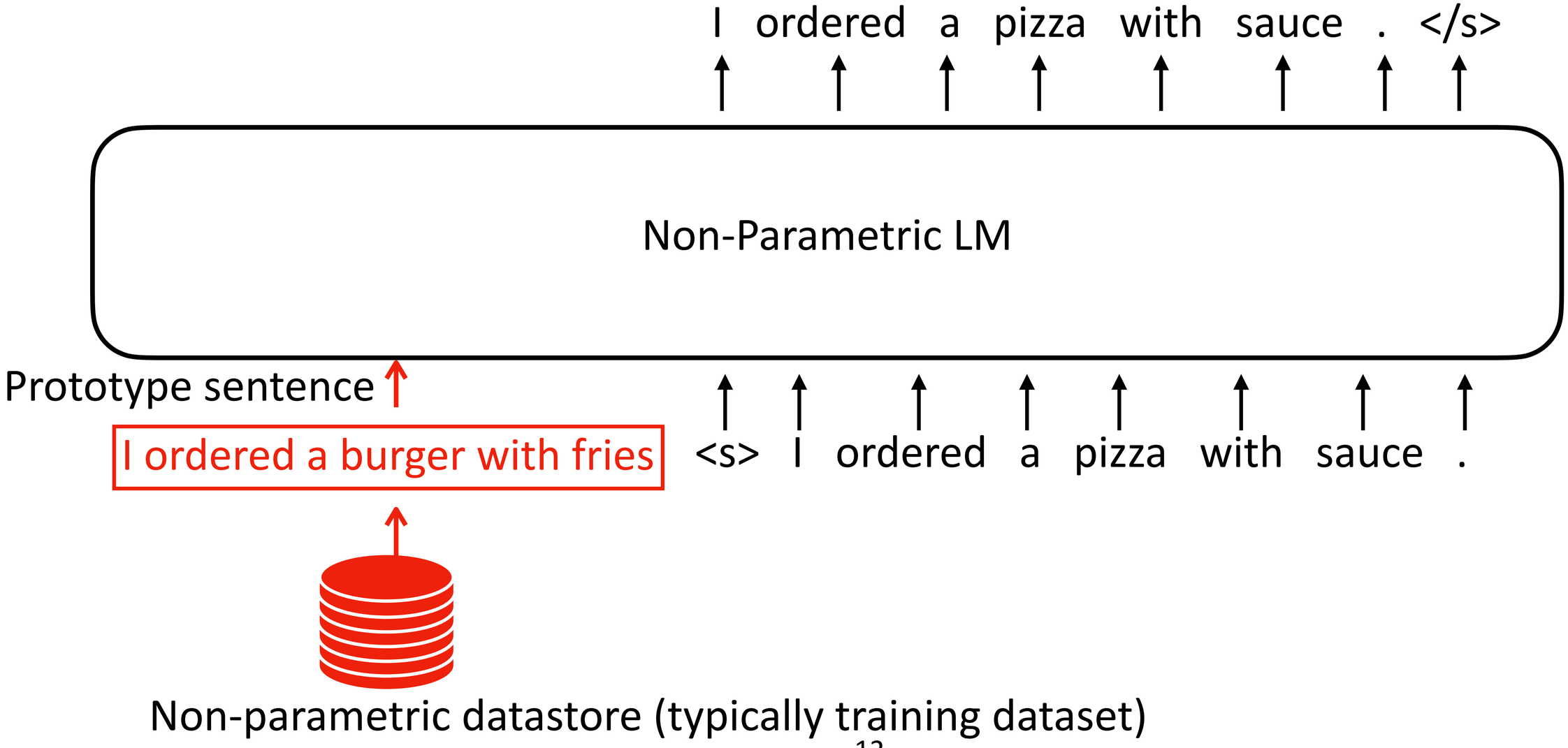
Language Technologies Institute

Non-Parametric Language Models



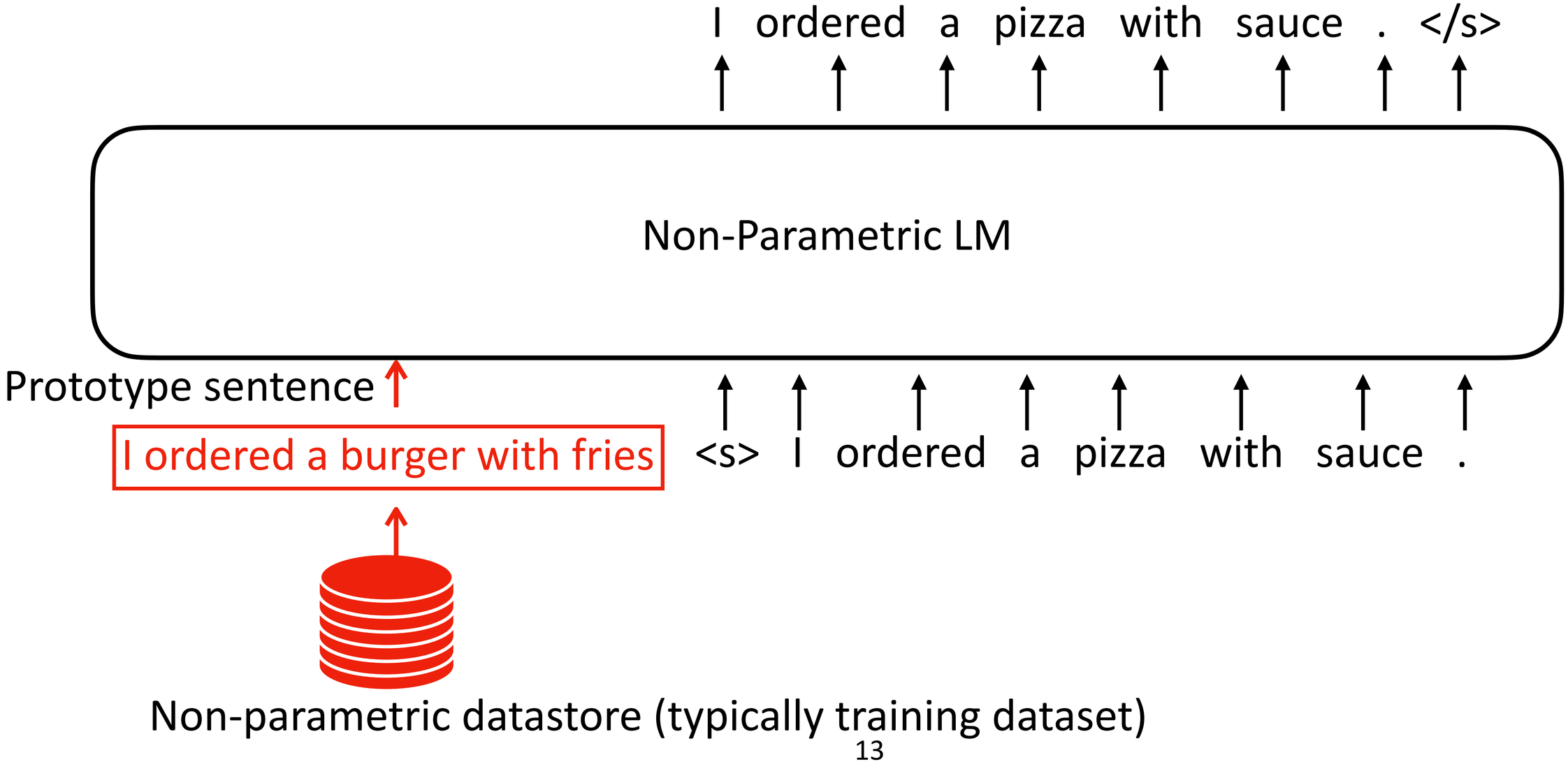
Language Technologies Institute

Non-Parametric Language Models



Language Technologies Institute

Non-Parametric Language Models



Language
Technologies
Institute

A Human Processing Analogy

Language Technologies Institute

A Human Processing Analogy

Our Language Ability/ Knowledge

Language Technologies Institute

A Human Processing Analogy

Our Language Ability/ Knowledge

Our Language Ability/ Knowledge + Wikipedia/Google

Language
Technologies
Institute

A (Very Incomplete) Survey of Neural Non-parametric Models for NLP

Language Technologies Institute

A (Very Incomplete) Survey of Neural Non-parametric Models for NLP

- **Count-based non-parametric data stores:**
  - *Neural Machine Translation with External Phrase Memory. Tang et al. 2016.*
  - *Incorporating Discrete Translation Lexicons into Neural Machine Translation. Arthur et al. 2016.*
  - *Generalizing and Hybridizing Count-based and Neural Language Models. Neubig and Dyer 2017.*

Language Technologies Institute

A (Very Incomplete) Survey of Neural Non-parametric Models for NLP

- **Count-based non-parametric data stores:**
  - *Neural Machine Translation with External Phrase Memory. Tang et al. 2016.*
  - *Incorporating Discrete Translation Lexicons into Neural Machine Translation. Arthur et al. 2016.*
  - *Generalizing and Hybridizing Count-based and Neural Language Models. Neubig and Dyer 2017.*
- **Bias probabilities based on count-based aggregation of retrieved sentences:**
  - *Guiding Neural Machine Translation with Retrieved Translation Pieces. Zhang et al. 2018.*
  - *Deep Weighted Averaging Classifiers. Card et al. 2019.*
  - *Nearest Neighbor Machine Translation. Khandelwal et al. 2020.*

Language Technologies Institute

A (Very Incomplete) Survey of Neural Non-parametric Models for NLP

- **Count-based non-parametric data stores:**
  - *Neural Machine Translation with External Phrase Memory. Tang et al. 2016.*
  - *Incorporating Discrete Translation Lexicons into Neural Machine Translation. Arthur et al. 2016.*
  - *Generalizing and Hybridizing Count-based and Neural Language Models. Neubig and Dyer 2017.*
- **Bias probabilities based on count-based aggregation of retrieved sentences:**
  - *Guiding Neural Machine Translation with Retrieved Translation Pieces. Zhang et al. 2018.*
  - *Deep Weighted Averaging Classifiers. Card et al. 2019.*
  - *Nearest Neighbor Machine Translation. Khandelwal et al. 2020.*
- **Attend to retrieved sentences:**
  - *Search Engine Guided Non-Parametric Neural Machine Translation. Gu et al. 2018.*
  - *Generating Sentences by Editing Prototypes. Guu et al. 2018.*
  - *REALM: Retrieval-Augmented Language Model Pre-Training. Guu et al. 2020.*

Language Technologies Institute

A (Very Incomplete) Survey of Neural Non-parametric Models for NLP

- **Count-based non-parametric data stores:**
  - *Neural Machine Translation with External Phrase Memory. Tang et al. 2016.*
  - *Incorporating Discrete Translation Lexicons into Neural Machine Translation. Arthur et al. 2016.*
  - *Generalizing and Hybridizing Count-based and Neural Language Models. Neubig and Dyer 2017.*
- **Bias probabilities based on count-based aggregation of retrieved sentences:**
  - *Guiding Neural Machine Translation with Retrieved Translation Pieces. Zhang et al. 2018.*
  - *Deep Weighted Averaging Classifiers. Card et al. 2019.*
  - *Nearest Neighbor Machine Translation. Khandelwal et al. 2020.*
- **Attend to retrieved sentences:**
  - *Search Engine Guided Non-Parametric Neural Machine Translation. Gu et al. 2018.*
  - *Generating Sentences by Editing Prototypes. Guu et al. 2018.*
  - *REALM: Retrieval-Augmented Language Model Pre-Training. Guu et al. 2020.*
- **Feed retrieved states into model:**
  - *Learning to Remember Rare Events. Kaiser et al. 2017.*

Language Technologies Institute

A (Very Incomplete) Survey of Neural Non-parametric Models for NLP

- **Count-based non-parametric data stores:**
  - *Neural Machine Translation with External Phrase Memory. Tang et al. 2016.*
  - *Incorporating Discrete Translation Lexicons into Neural Machine Translation. Arthur et al. 2016.*
  - *Generalizing and Hybridizing Count-based and Neural Language Models. Neubig and Dyer 2017.*
- **Bias probabilities based on count-based aggregation of retrieved sentences:**
  - *Guiding Neural Machine Translation with Retrieved Translation Pieces. Zhang et al. 2018.*
  - *Deep Weighted Averaging Classifiers. Card et al. 2019.*
  - *Nearest Neighbor Machine Translation. Khandelwal et al. 2020.*
- **Attend to retrieved sentences:**
  - *Search Engine Guided Non-Parametric Neural Machine Translation. Gu et al. 2018.*
  - *Generating Sentences by Editing Prototypes. Guu et al. 2018.*
  - *REALM: Retrieval-Augmented Language Model Pre-Training. Guu et al. 2020.*
- **Feed retrieved states into model:**
  - *Learning to Remember Rare Events. Kaiser et al. 2017.*
- **Fine-tune models on retrieved sentences:**
  - *One Sentence One Model for Neural Machine Translation. Li et al. 2016.*

Language Technologies Institute

Non-Parametric Language Models: Pros/Cons

Pros

- Mitigate pressure on the parametric models
- Improved interpretability in the modeling process

Language Technologies Institute

Non-Parametric Language Models: Pros/Cons

Pros

- Mitigate pressure on the parametric models
- Improved interpretability in the modeling process

Cons

- In almost all cases, large parametric datastore leads to significant issues with memory and speed efficiency at test time

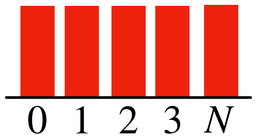
Language
Technologies
Institute

A Concrete Example: Prototype-based Language Models (Guu et al. TACL 2018)

Language
Technologies
Institute

Previous Approach

Prototype distribution

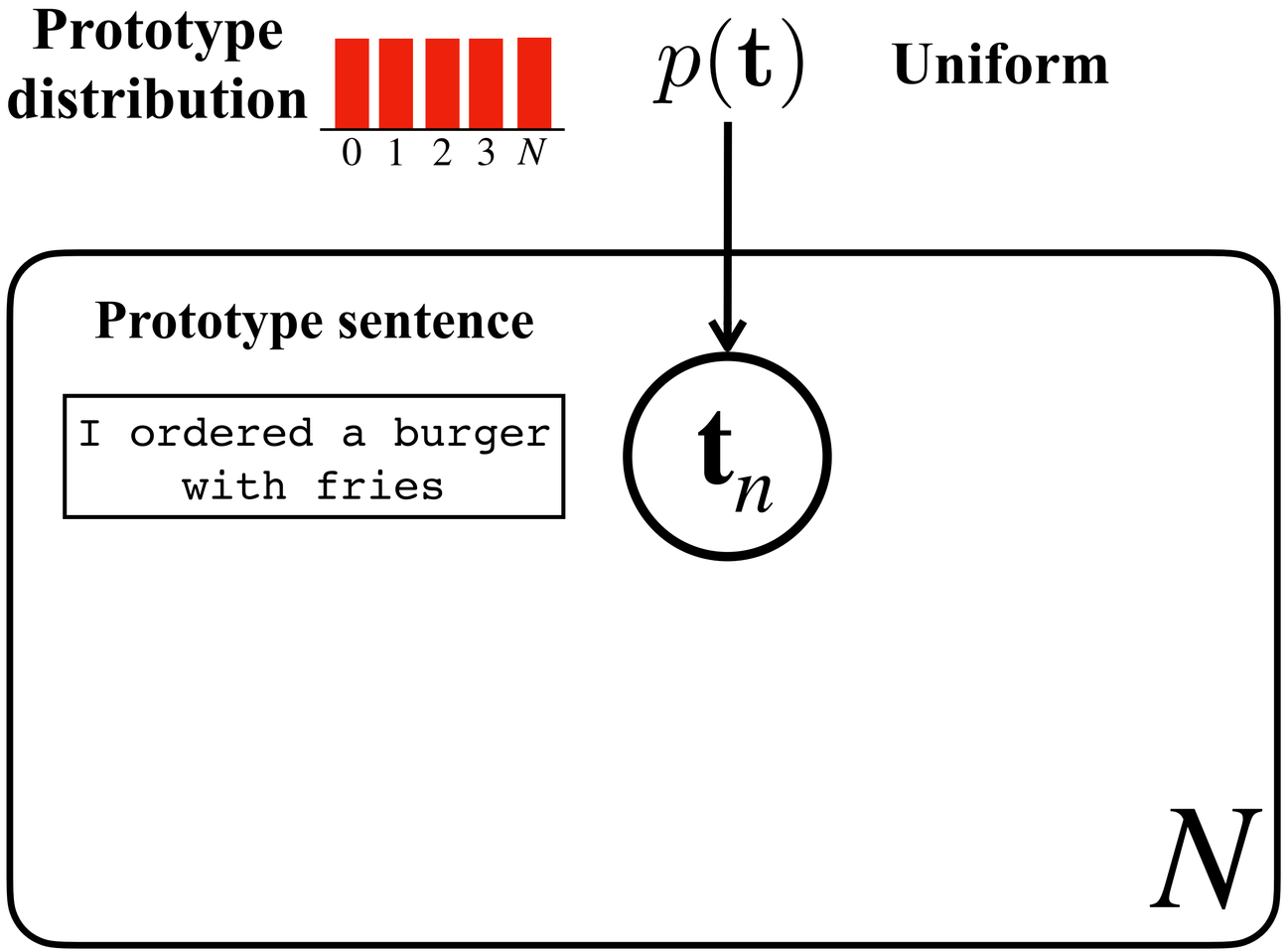


$p(\mathbf{t})$ **Uniform**

Guu et al. 2018

Language
Technologies
Institute

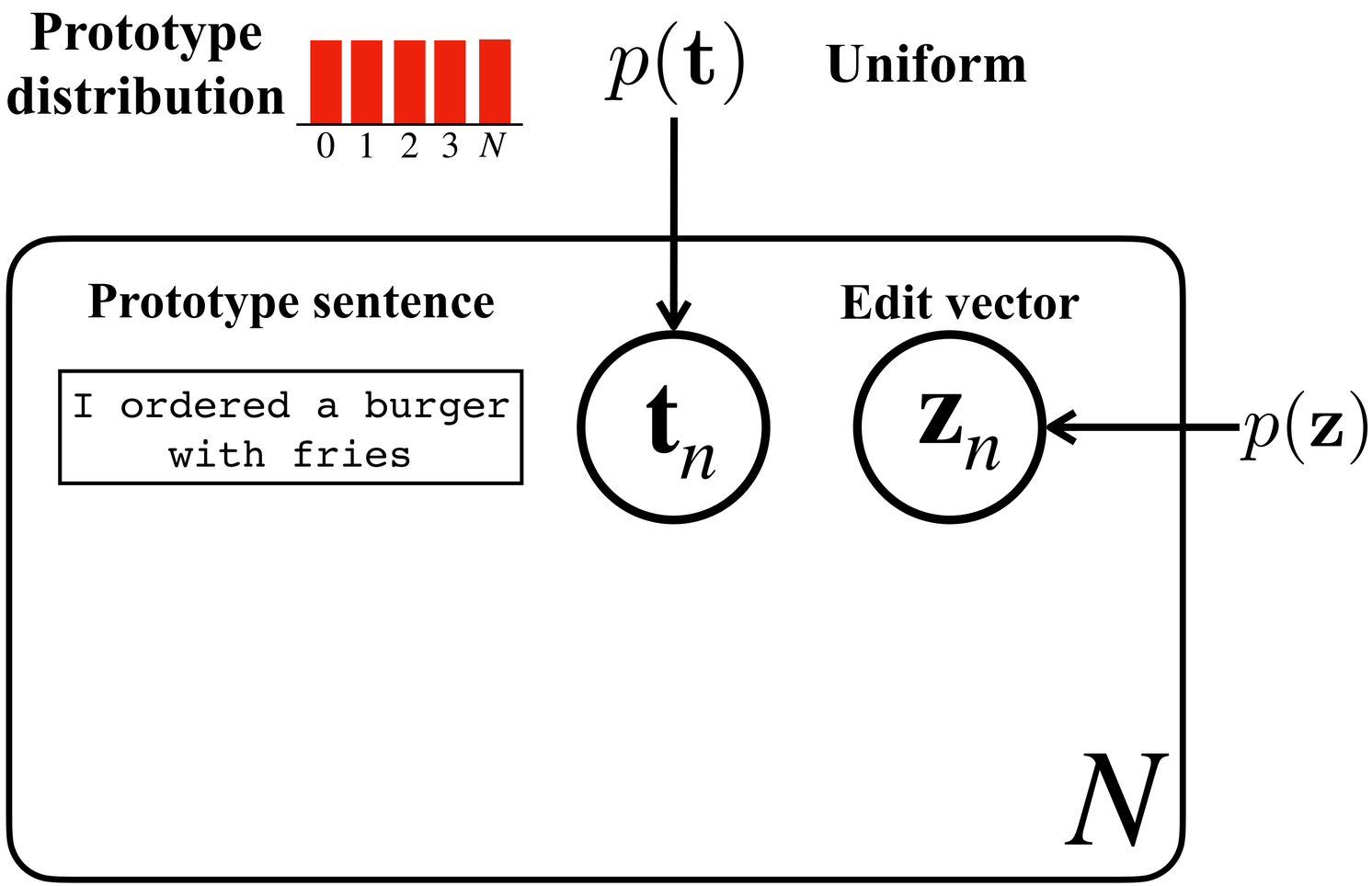
Previous Approach



Guu et al. 2018

Language Technologies Institute

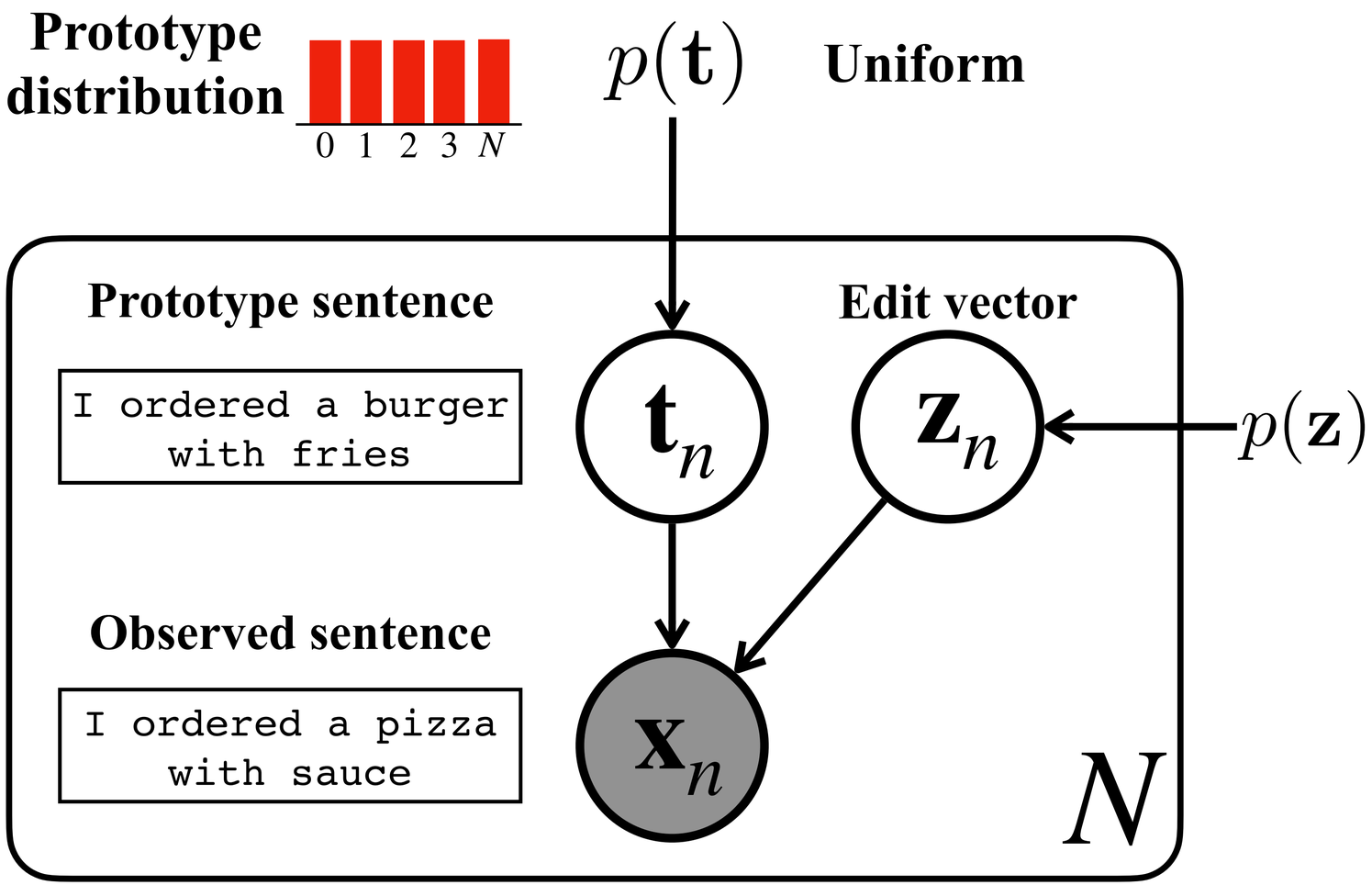
Previous Approach



Guu et al. 2018

Language Technologies Institute

Previous Approach



Guu et al. 2018

Language
Technologies
Institute

Proposed Model:
Sparse Prototypes for Text Generation
(He et al. NeurIPS 2020)

Language Technologies Institute

Prototypes can be Sparse

Prototype Sentence

I ordered a burger with fries

Data Examples

I ordered a pizza with sauce

I ordered a turkey with sauce

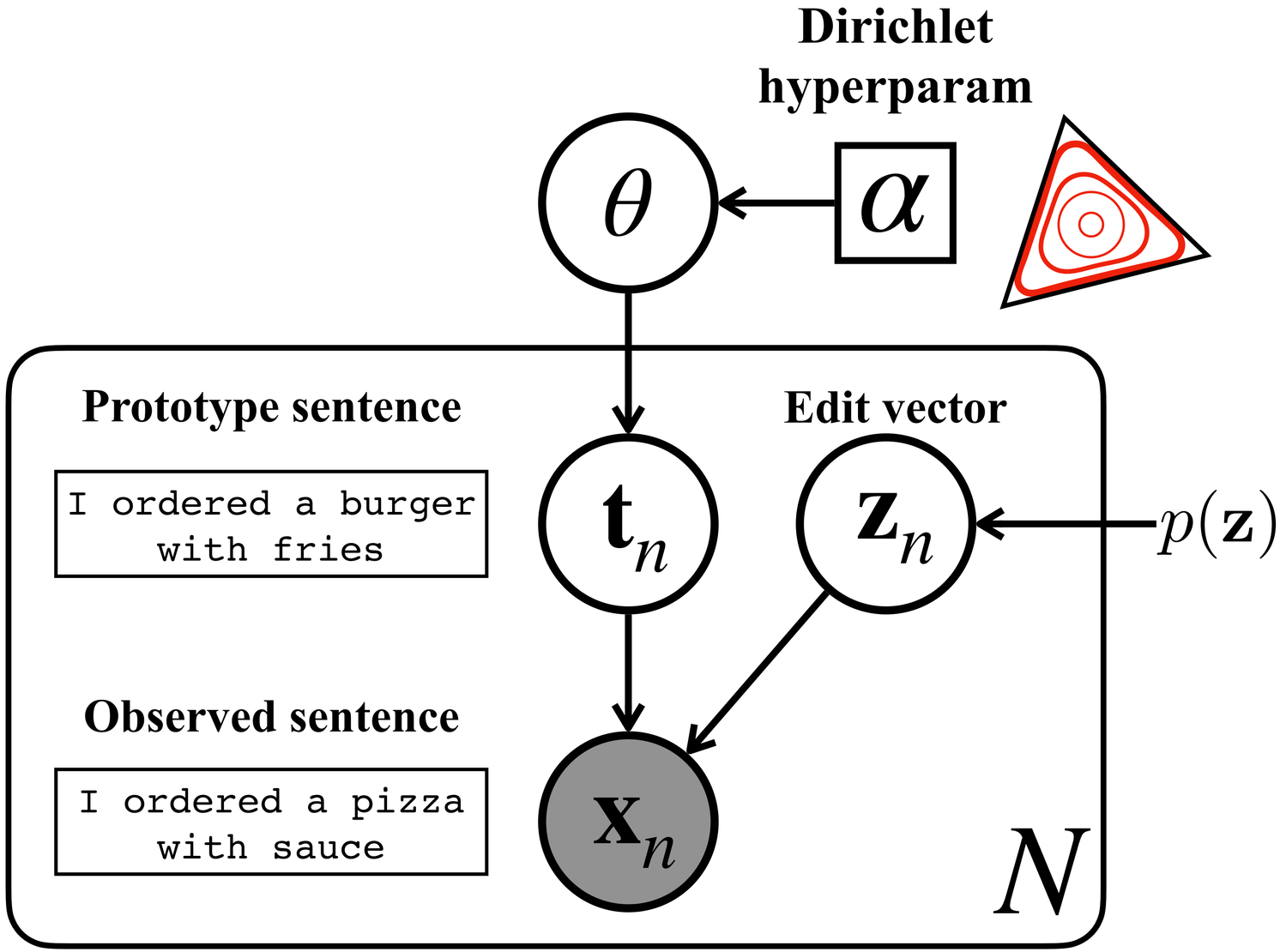
I ordered a sandwich with fries and coke

I ordered a lobster roll with oysters and spicy sauce

Guu et al. 2018

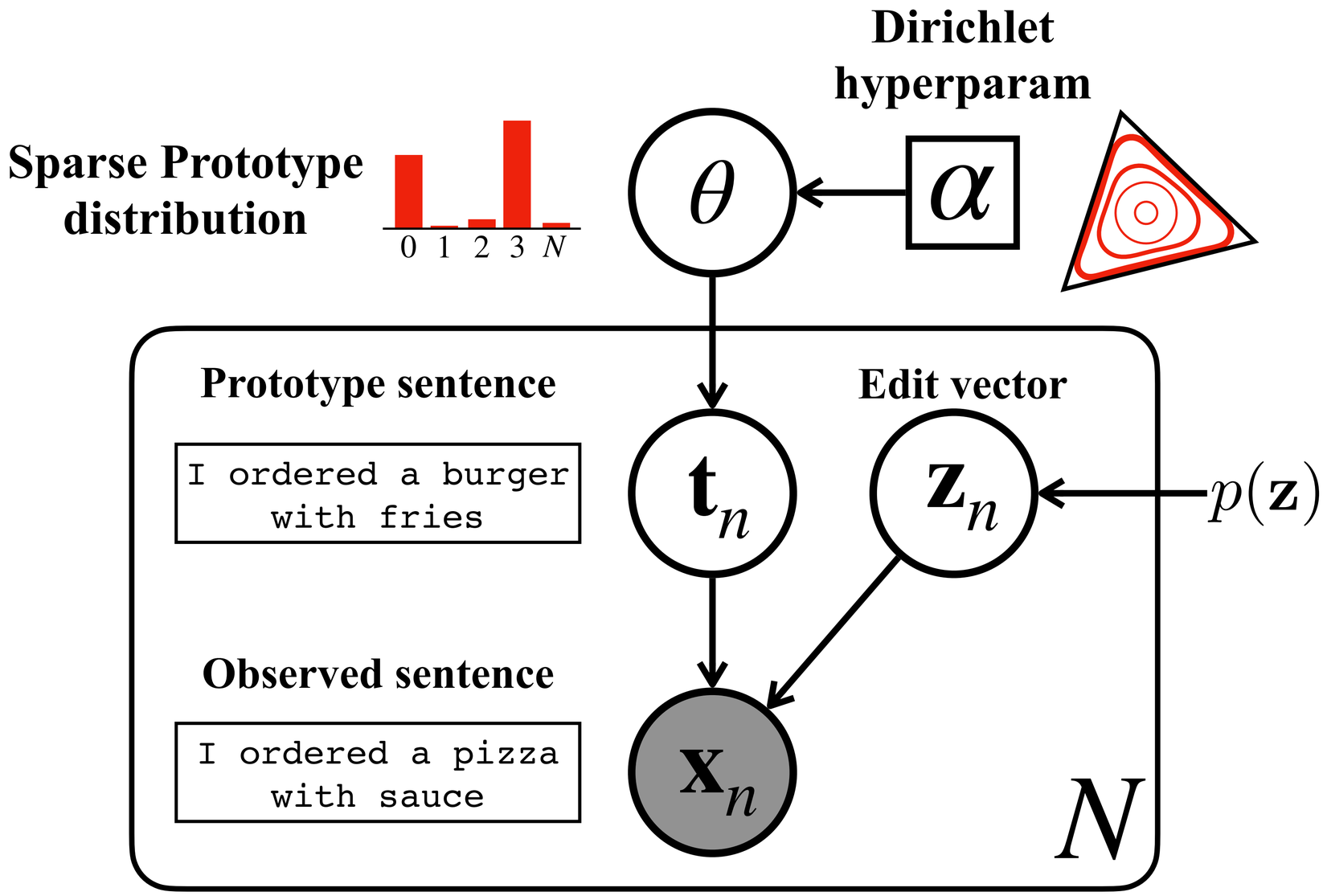
Language Technologies Institute

Sparse Neural Editor



Language
Technologies
Institute

Sparse Neural Editor



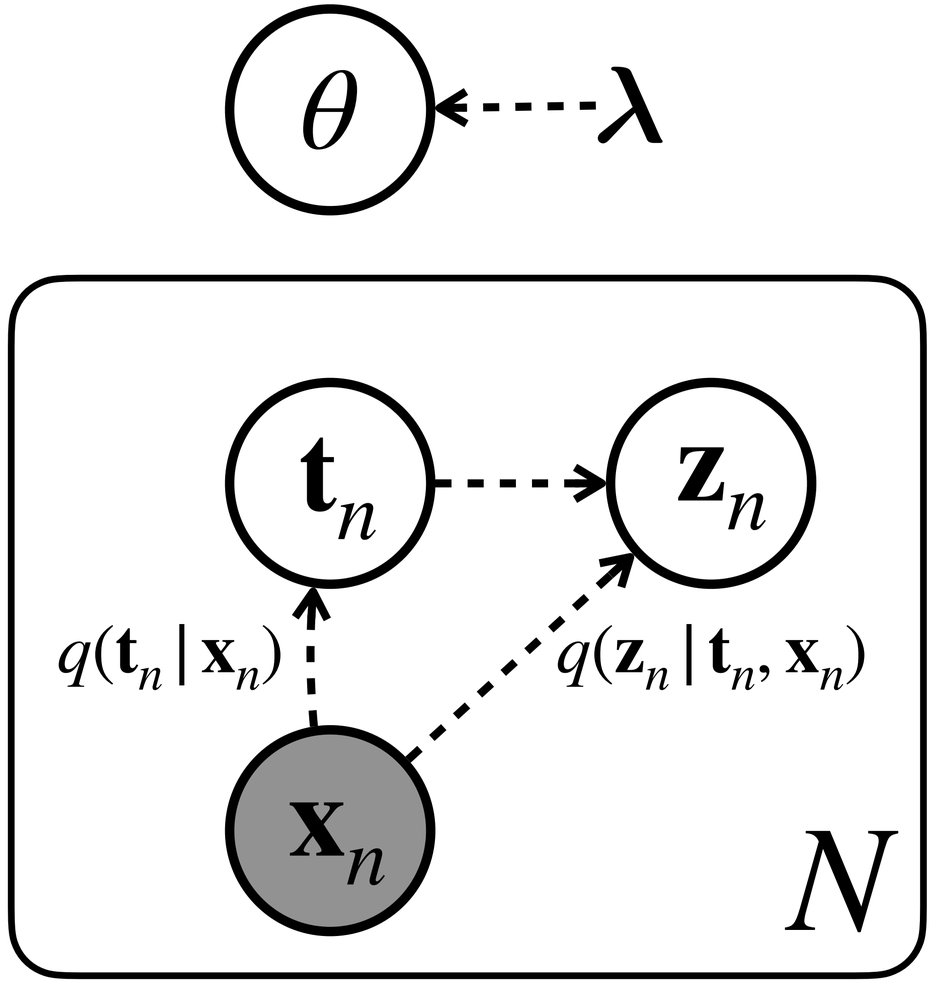
Language Technologies Institute

Learning and Inference

$$\log p(\mathbf{x}) \geq \mathcal{L}_{\text{ELBO}}$$

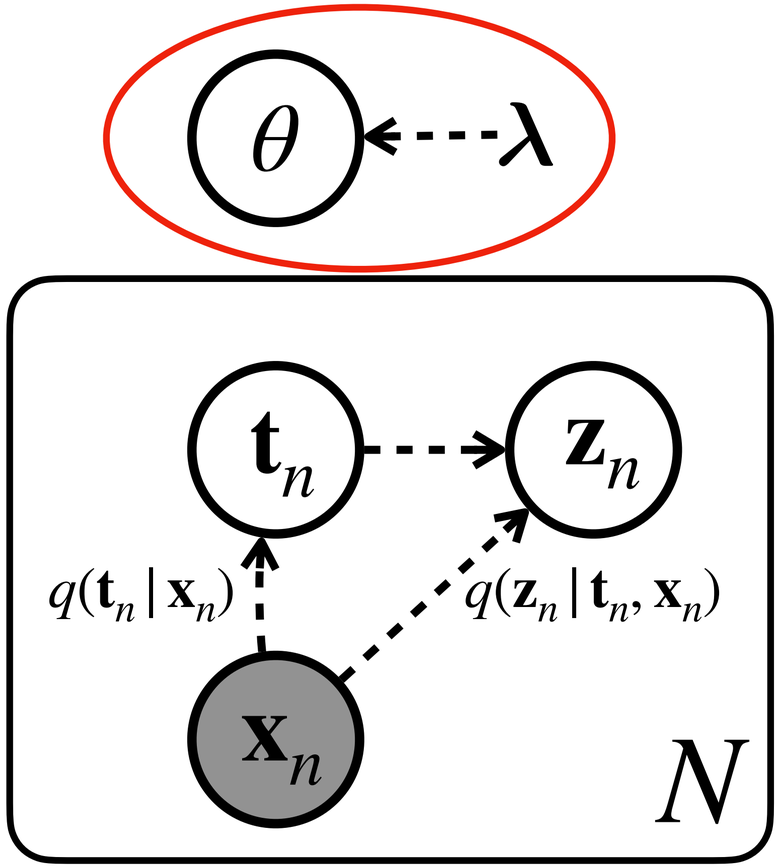
$$= \mathbb{E}_{q(\mathbf{t},\mathbf{z},\theta|\mathbf{x})} \log p(\mathbf{x}|\mathbf{t},\mathbf{z},\theta) - D_{\text{KL}}(q(\mathbf{t},\mathbf{z},\theta)||p(\mathbf{t},\mathbf{z},\theta))$$

Approximate posterior



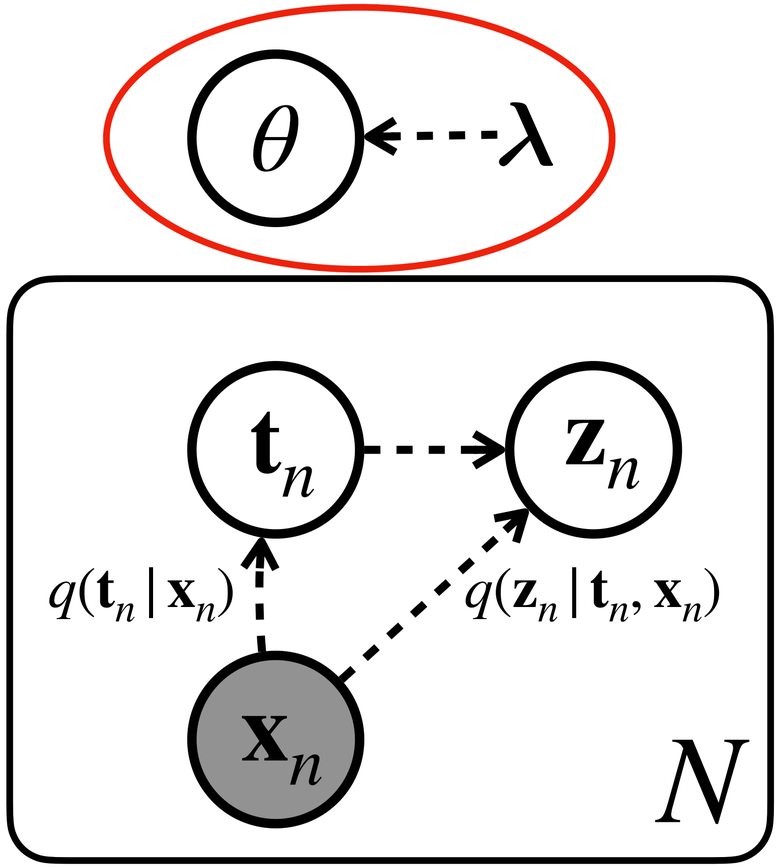
Language
Technologies
Institute

The Approximate Posterior



Language
Technologies
Institute

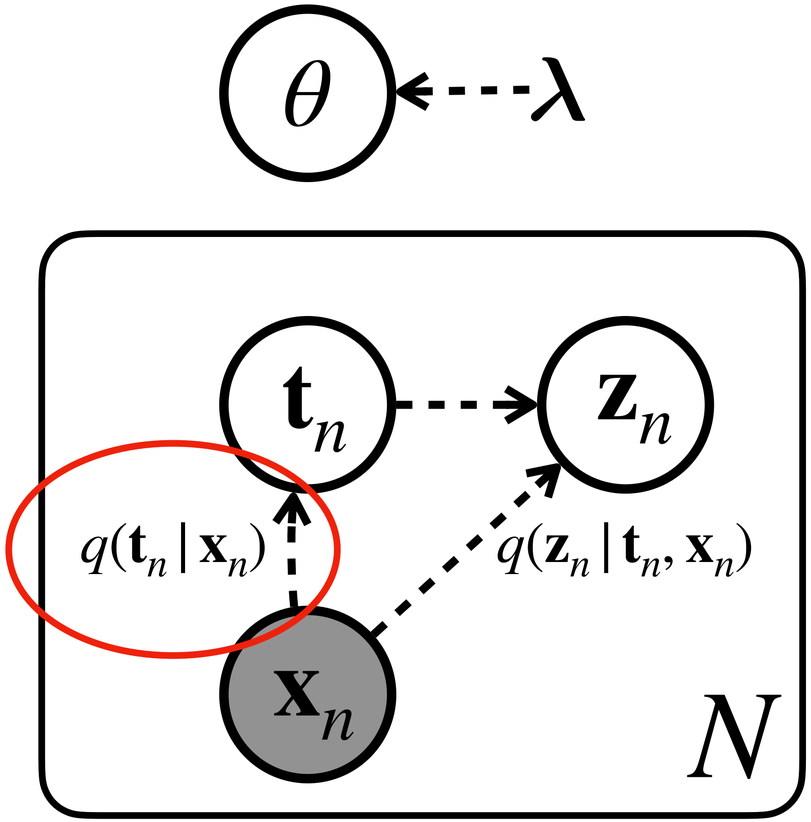
The Approximate Posterior



$$q(\theta) = \mathrm{Dir}(\theta; \boldsymbol{\lambda})$$

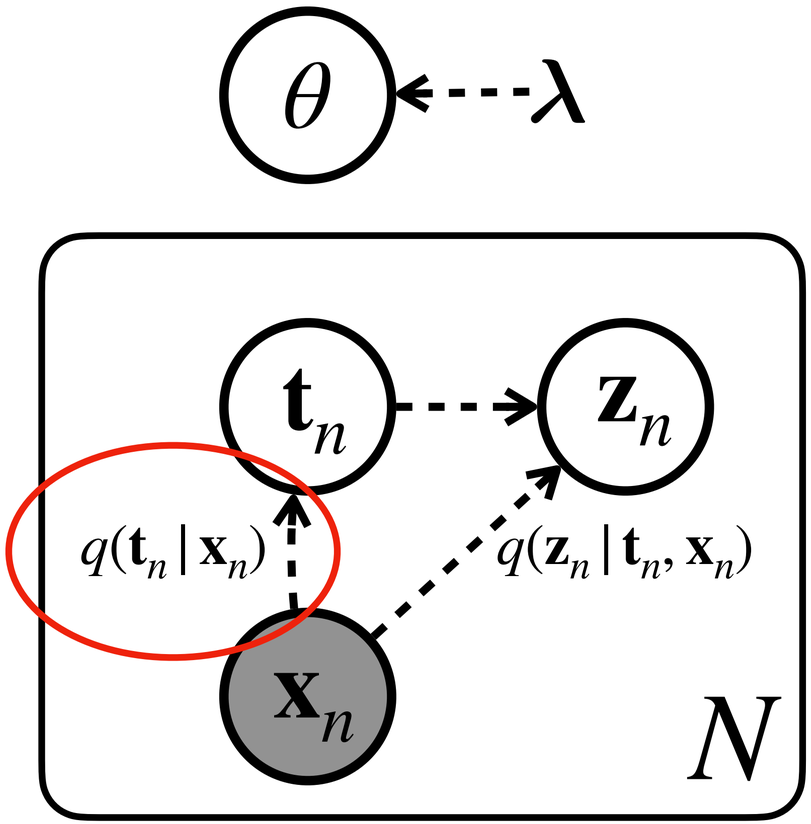
Language
Technologies
Institute

The Approximate Posterior



Language
Technologies
Institute

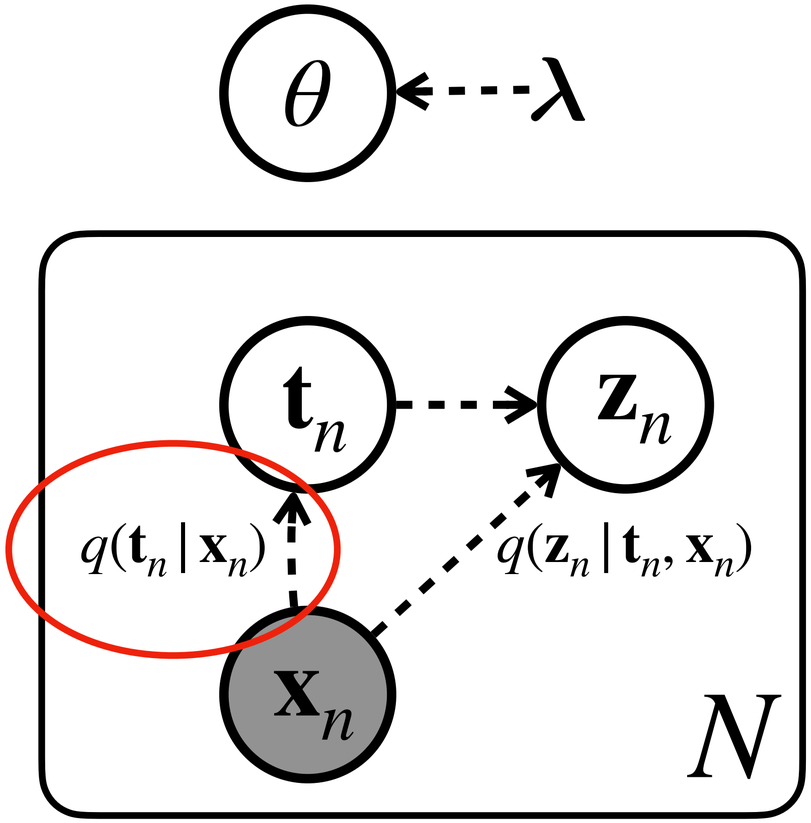
The Approximate Posterior



$$q(\mathbf{t} = \mathbf{x}_k|\mathbf{x}) \propto \begin{cases} \exp\left(h(\mathbf{x}_k, \mathbf{x})/\mu\right) & \text{if } \mathbf{x}_k \neq \mathbf{x} \\ 0 & \text{if } \mathbf{x}_k = \mathbf{x}, \end{cases}$$

Language Technologies Institute

The Approximate Posterior

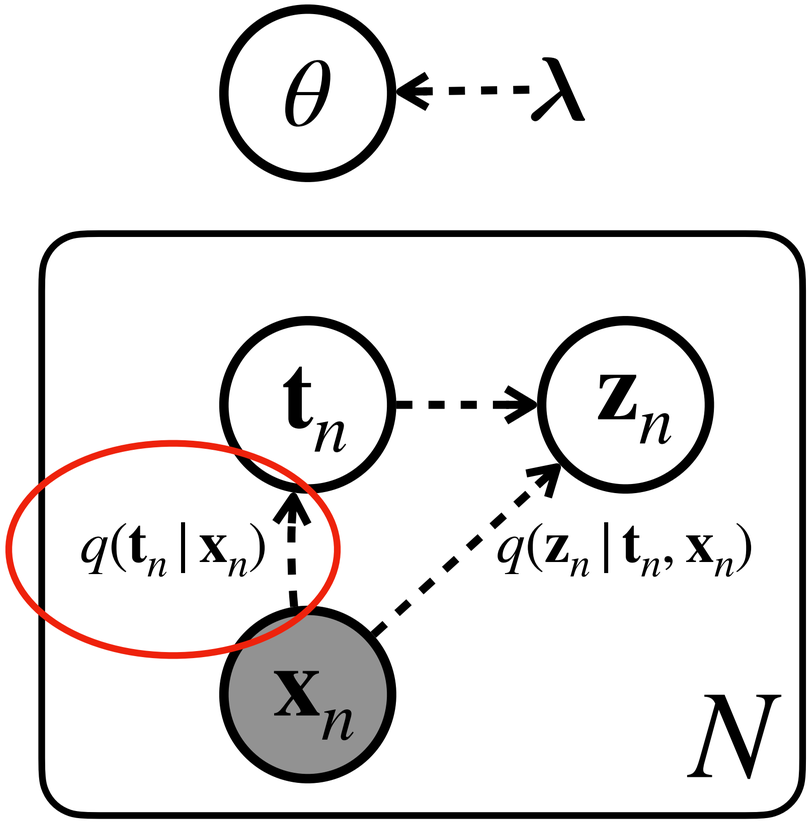


$$q(\mathbf{t} = \mathbf{x}_k | \mathbf{x}) \propto \begin{cases} \exp\left(h(\mathbf{x}_k, \mathbf{x})/\mu\right) & \text{if } \mathbf{x}_k \neq \mathbf{x} \\ 0 & \text{if } \mathbf{x}_k = \mathbf{x}, \end{cases}$$

$$h(\mathbf{x}_k, \mathbf{x}) = \texttt{Embed}(\mathbf{x}_k)^\top \boldsymbol{W} \texttt{Embed}(\mathbf{x})$$

Language Technologies Institute

The Approximate Posterior



$$q(\mathbf{t} = \mathbf{x}_k|\mathbf{x}) \propto \begin{cases} \exp\left(h(\mathbf{x}_k, \mathbf{x})/\mu\right) & \text{if } \mathbf{x}_k \neq \mathbf{x} \\ 0 & \text{if } \mathbf{x}_k = \mathbf{x}, \end{cases}$$

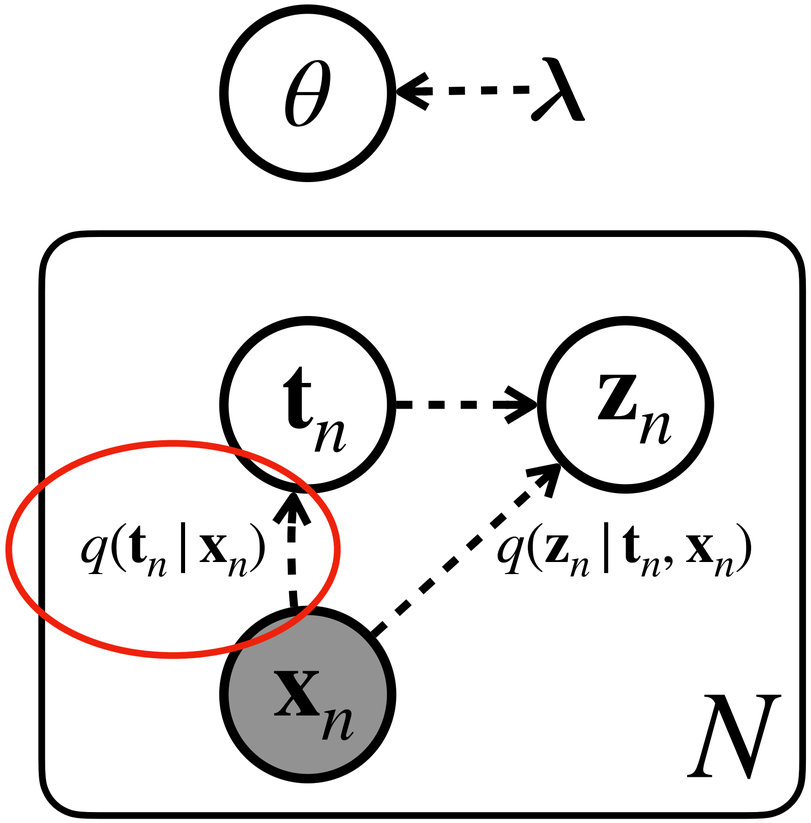
$$h(\mathbf{x}_k, \mathbf{x}) = \texttt{Embed}(\mathbf{x}_k)^\top \boldsymbol{W} \texttt{Embed}(\mathbf{x})$$

BERT

Data example x

Language Technologies Institute

The Approximate Posterior



$$q(\mathbf{t} = \mathbf{x}_k|\mathbf{x}) \propto \begin{cases} \exp\left(h(\mathbf{x}_k, \mathbf{x})/\mu\right) & \text{if } \mathbf{x}_k \neq \mathbf{x} \\ 0 & \text{if } \mathbf{x}_k = \mathbf{x}, \end{cases}$$

$$h(\mathbf{x}_k, \mathbf{x}) = \texttt{Embed}(\mathbf{x}_k)^\top \boldsymbol{W} \texttt{Embed}(\mathbf{x})$$

BERT

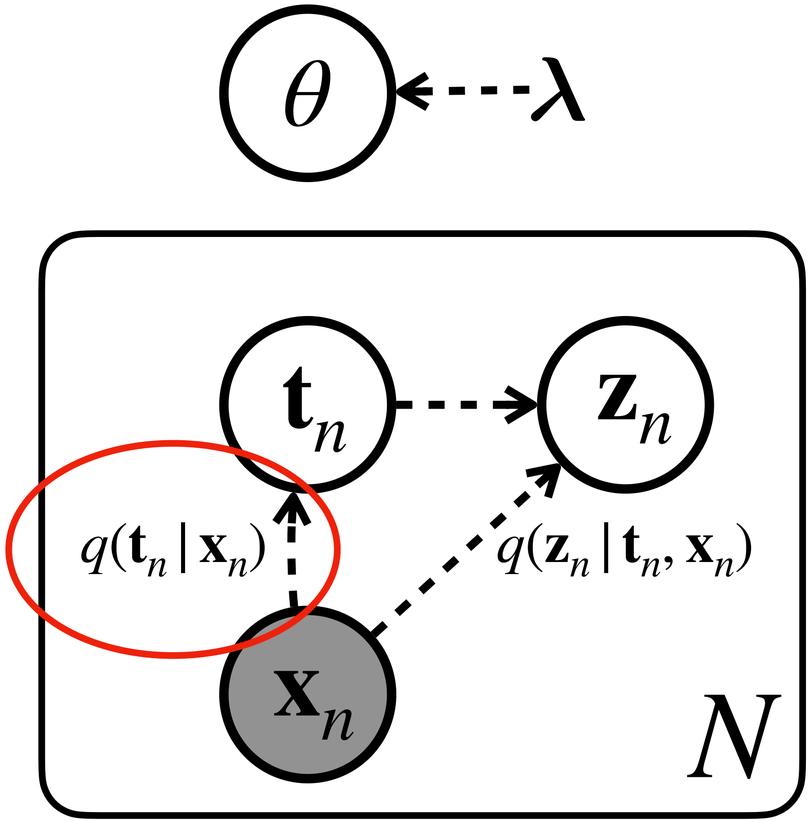
Data example x

BERT

Prototype x_k

Language Technologies Institute

The Approximate Posterior



$$q(\mathbf{t} = \mathbf{x}_k|\mathbf{x}) \propto \begin{cases} \exp\left(h(\mathbf{x}_k, \mathbf{x})/\mu\right) & \text{if } \mathbf{x}_k \neq \mathbf{x} \\ 0 & \text{if } \mathbf{x}_k = \mathbf{x}, \end{cases}$$

$$h(\mathbf{x}_k, \mathbf{x}) = \texttt{Embed}(\mathbf{x}_k)^\top \boldsymbol{W} \texttt{Embed}(\mathbf{x})$$

BERT

Data example x

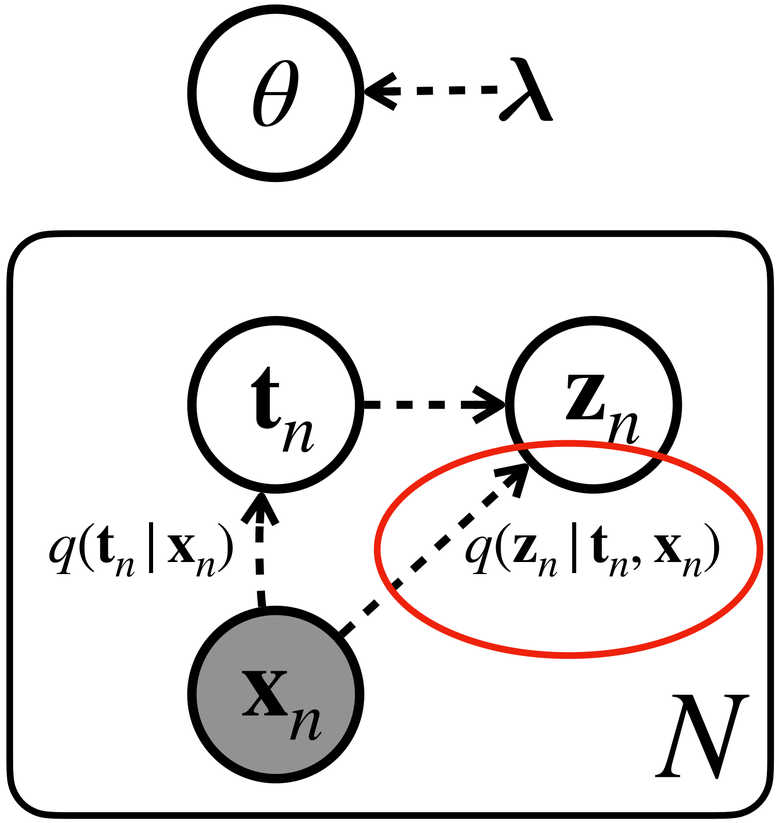
BERT

Prototype x_k

Only the linear transformation is learned while BERT is fixed. We use REINFORCE algorithm to propagate gradients over discrete prototype variables

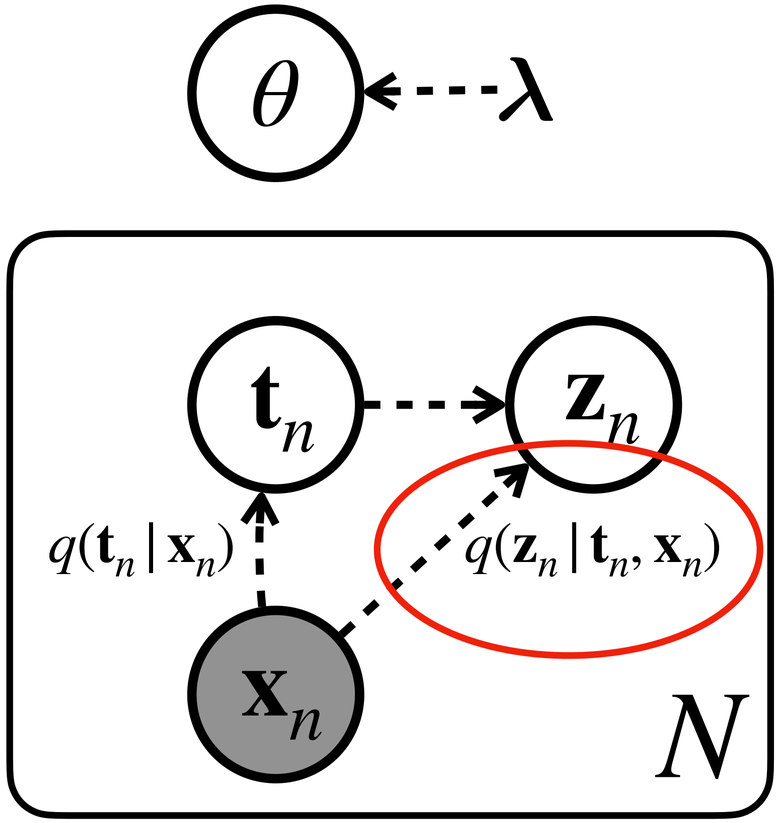
Language
Technologies
Institute

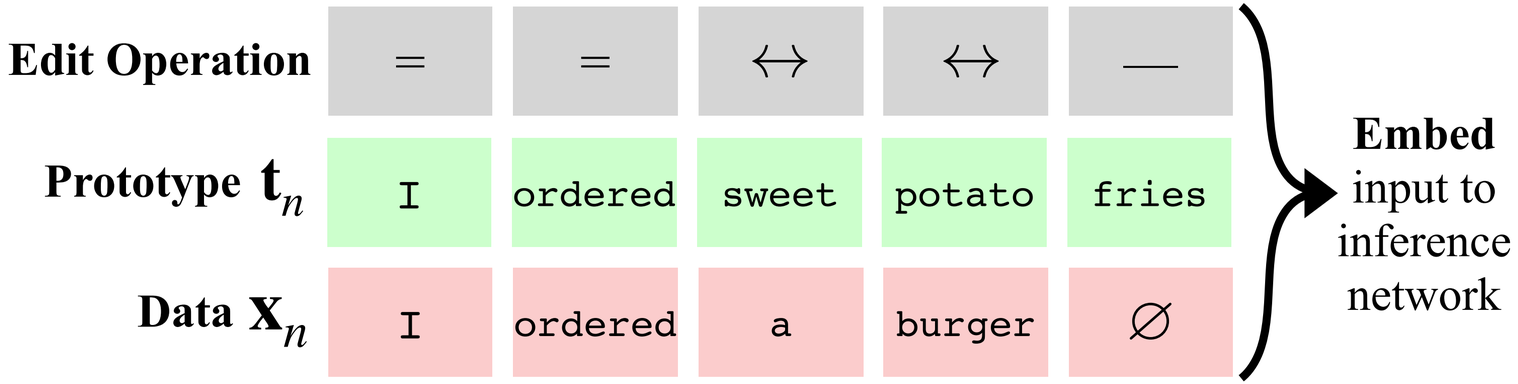
The Approximate Posterior



Language Technologies Institute

The Approximate Posterior





Language
Technologies
Institute

Experiments

Language
Technologies
Institute

Experiments

- Datasets
  - MSCOCO image caption (40K training)
  - Yelp restaurant reviews (17M training)

Language
Technologies
Institute

Perplexity Results

Yelp Large	Yelp Medium	Yelp Large	
prototypes: 779	prototypes: 1.5k	prototypes: 17M	2K

Language Technologies Institute

Perplexity Results

Yelp Large

prototypes: 779

Yelp Medium

prototypes: 1.5k

prototypes: 17M

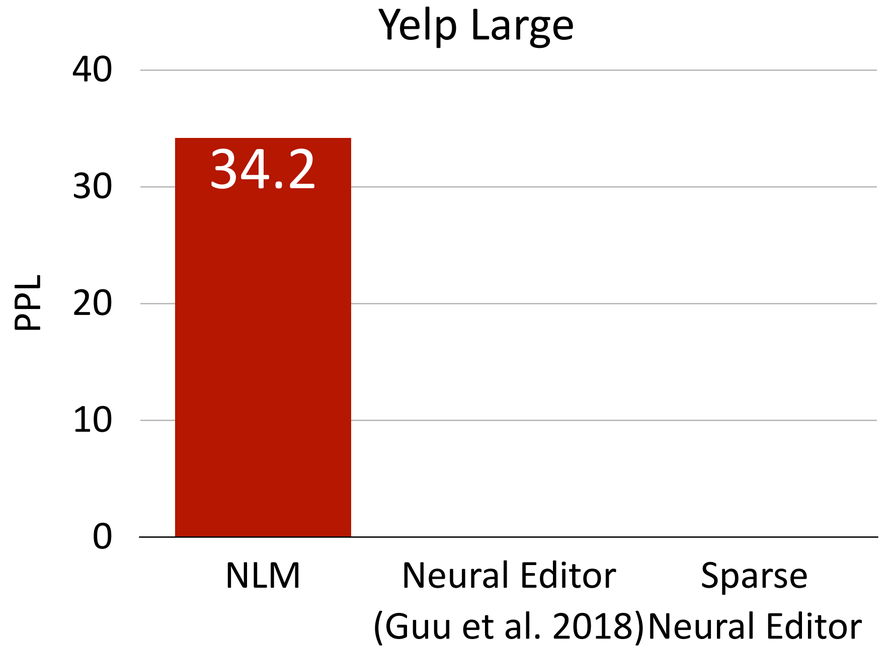
2K

Language Technologies Institute

Perplexity Results

Yelp Large

Yelp Medium



prototypes: 779

prototypes: 1.5k

prototypes: 17M

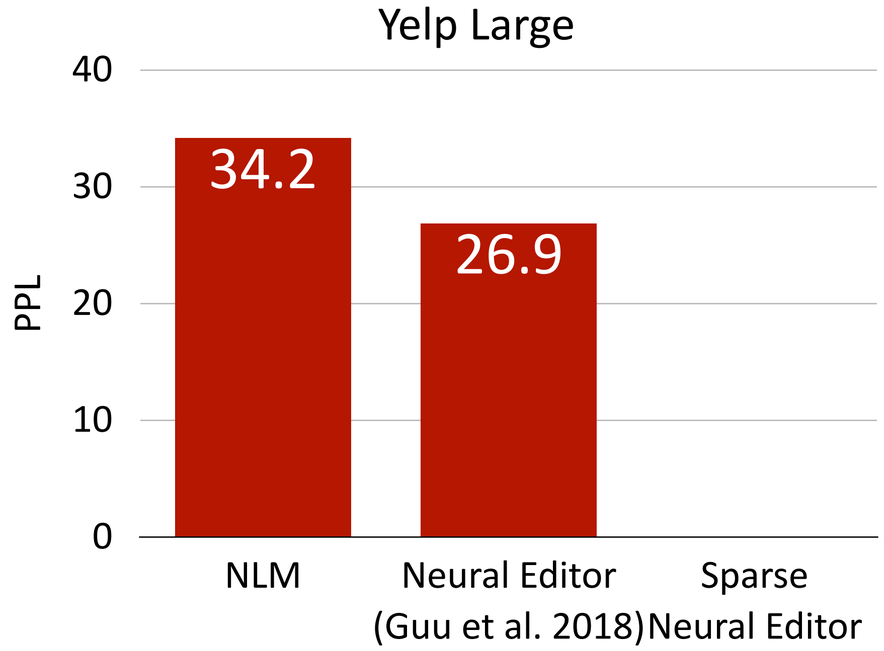
2K

Language Technologies Institute

Perplexity Results

Yelp Large

Yelp Medium



prototypes: 779

prototypes: 1.5k

prototypes: 17M

2K

Language Technologies Institute

Perplexity Results

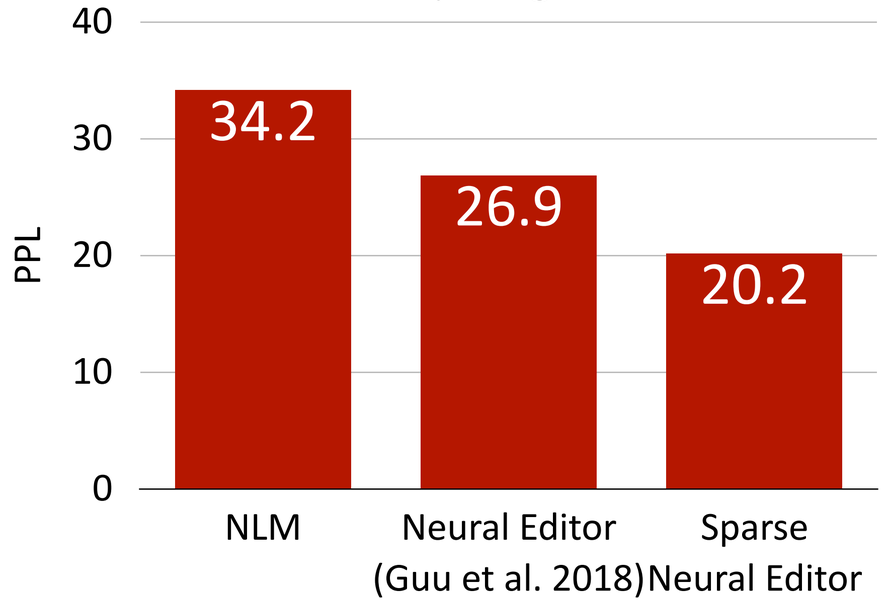
Yelp Large

prototypes: 779

Yelp Medium

prototypes: 1.5k

Yelp Large

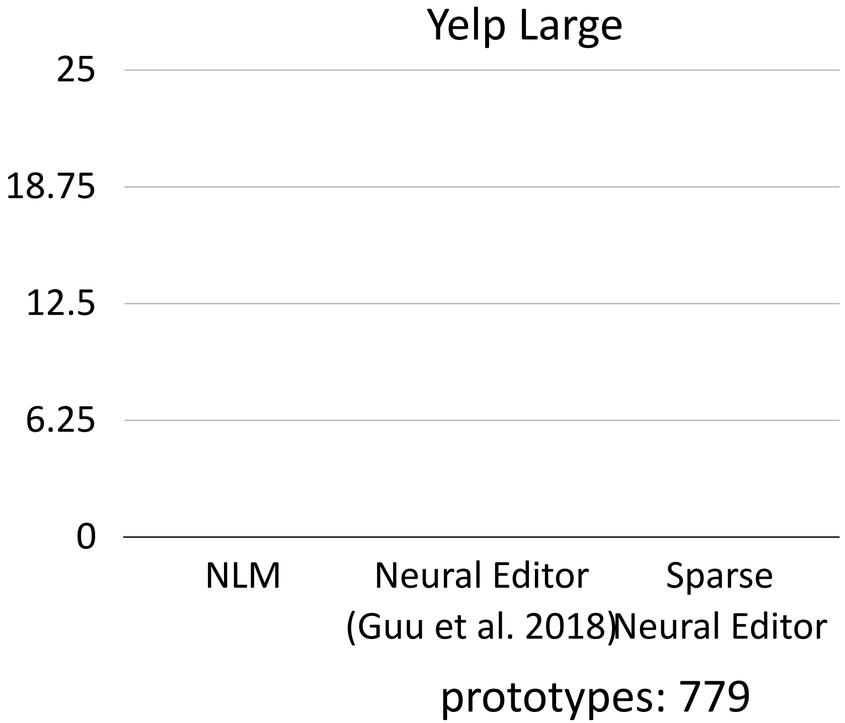


prototypes: 17M

2K

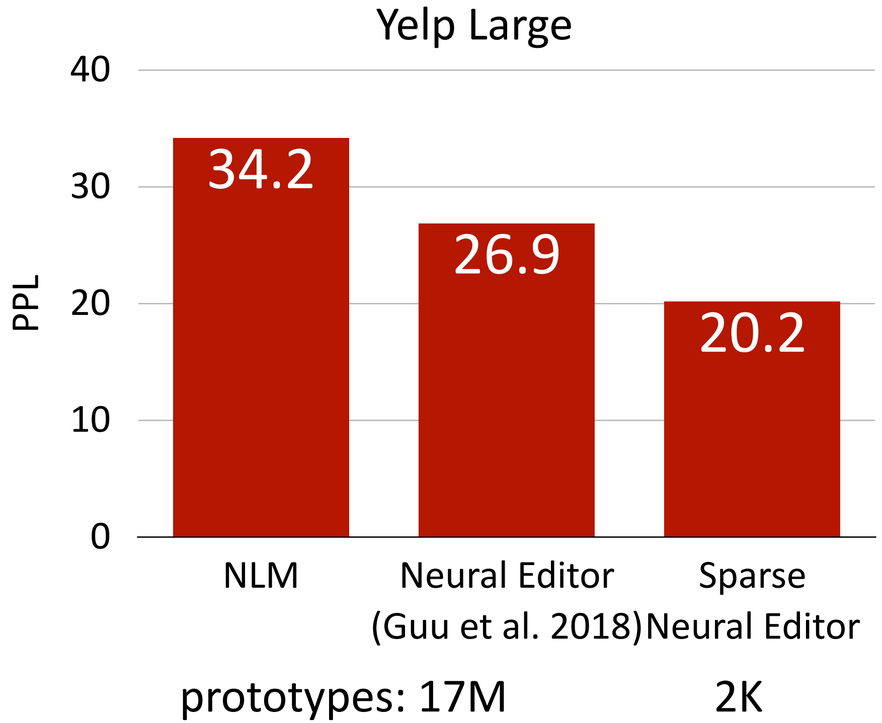
Language Technologies Institute

Perplexity Results



prototypes: 779

prototypes: 1.5k

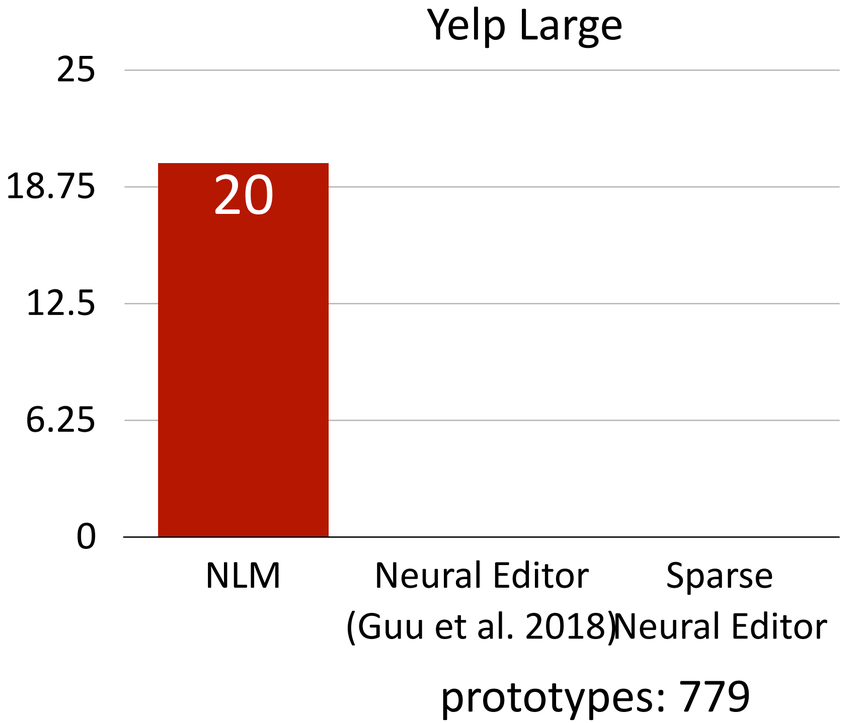


prototypes: 17M

2K

Language Technologies Institute

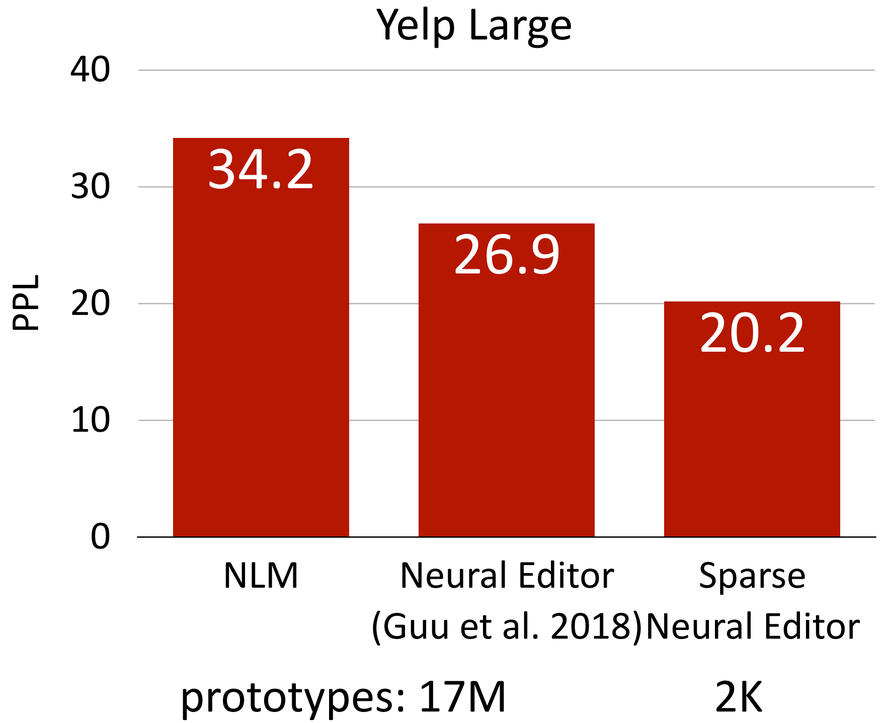
Perplexity Results



prototypes: 779

Yelp Medium

prototypes: 1.5k

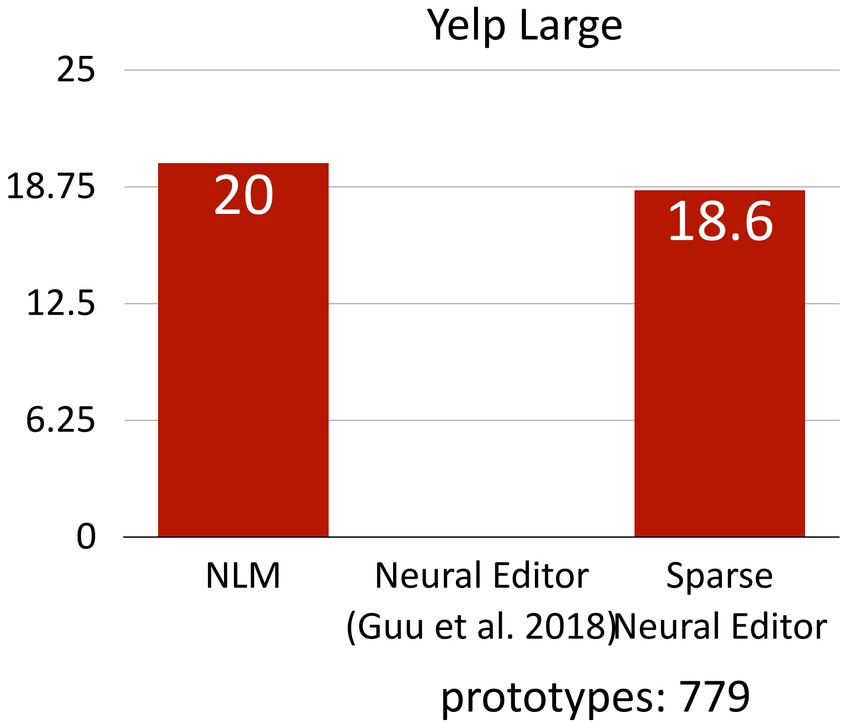


prototypes: 17M

2K

Language Technologies Institute

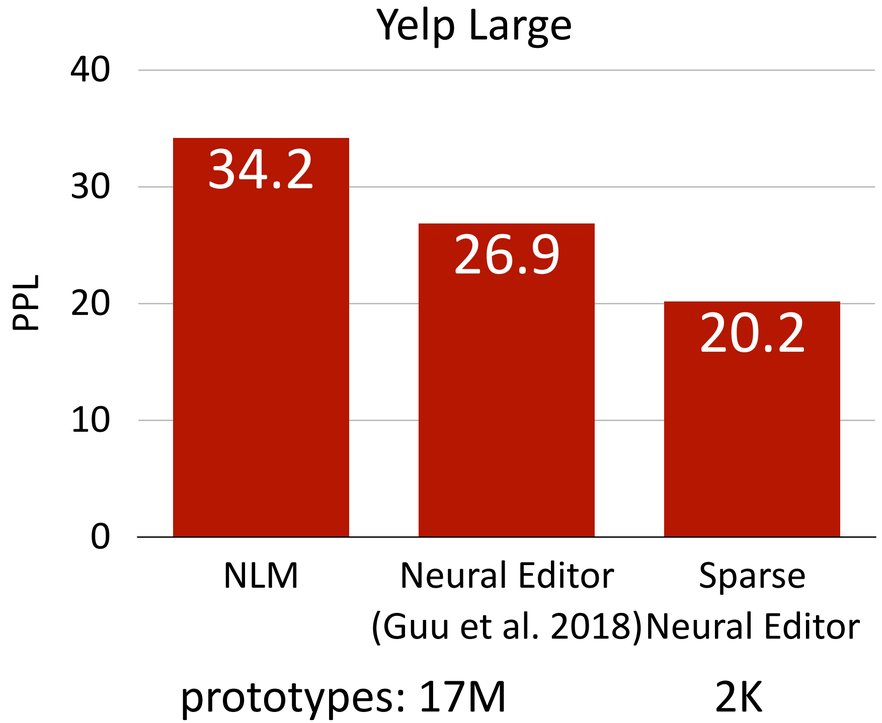
Perplexity Results



prototypes: 779

Yelp Medium

prototypes: 1.5k

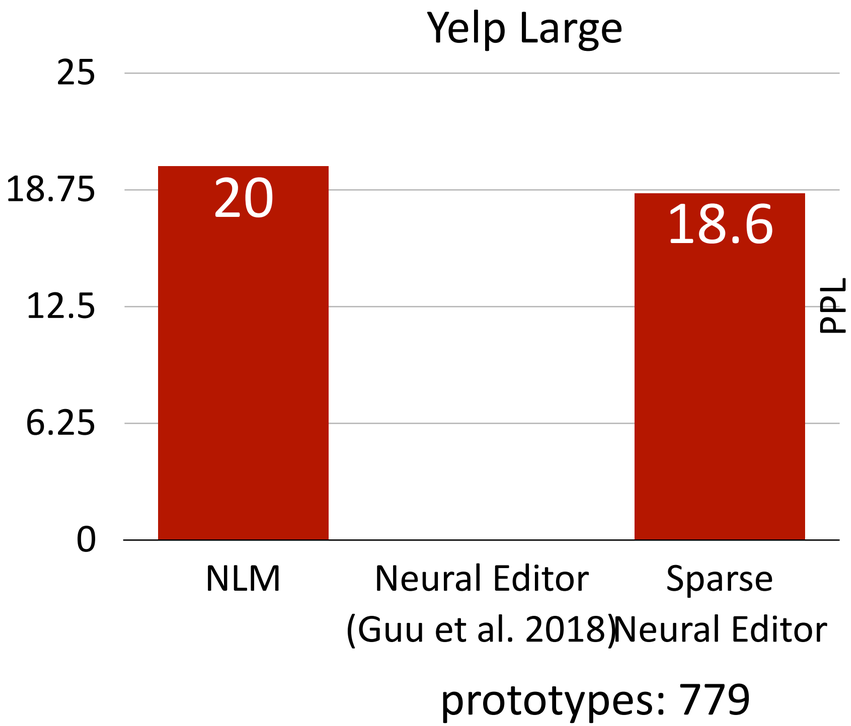


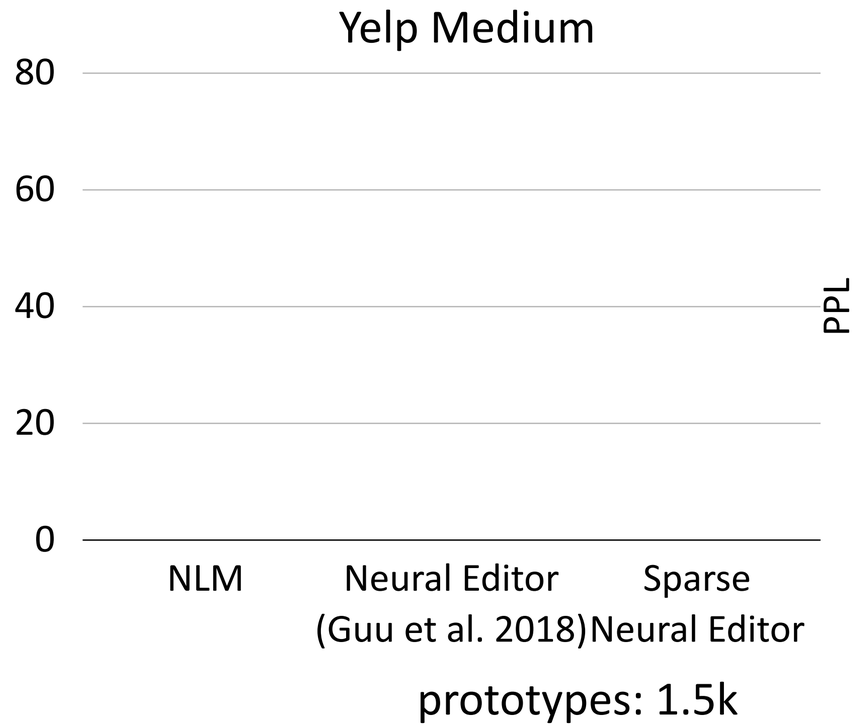
prototypes: 17M

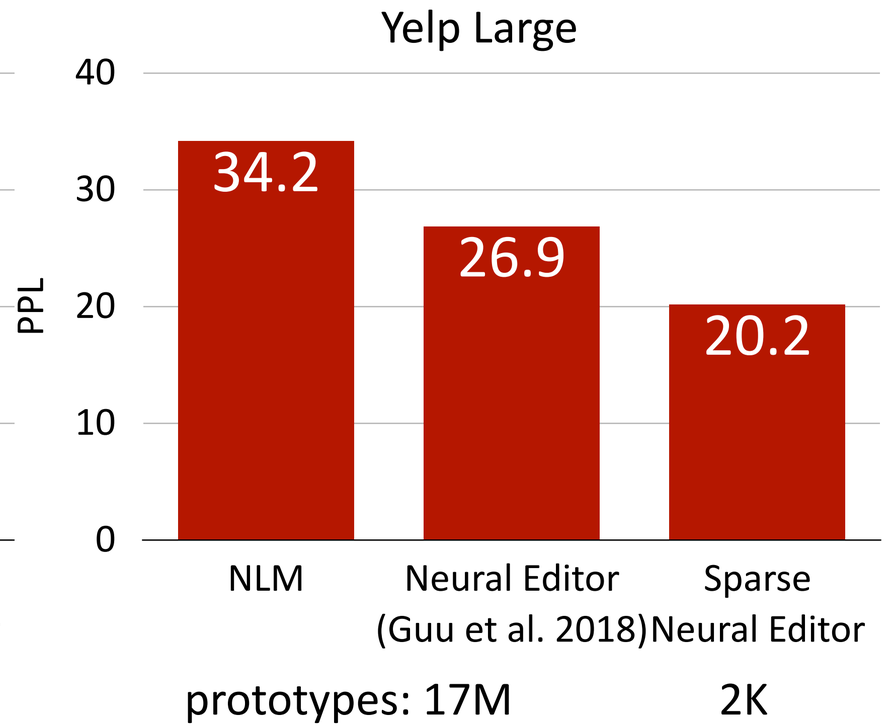
2K

Language Technologies Institute

Perplexity Results

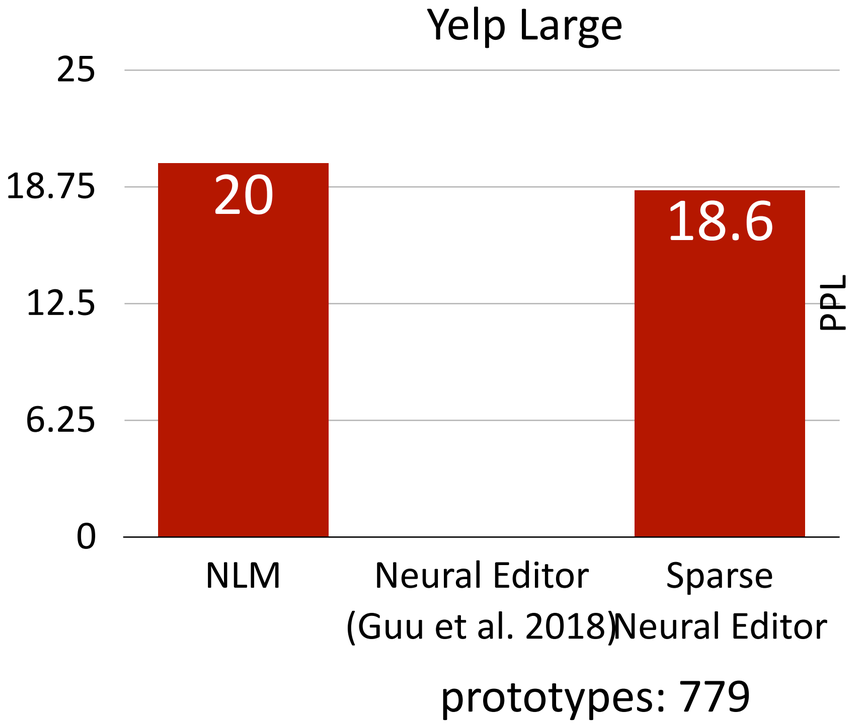


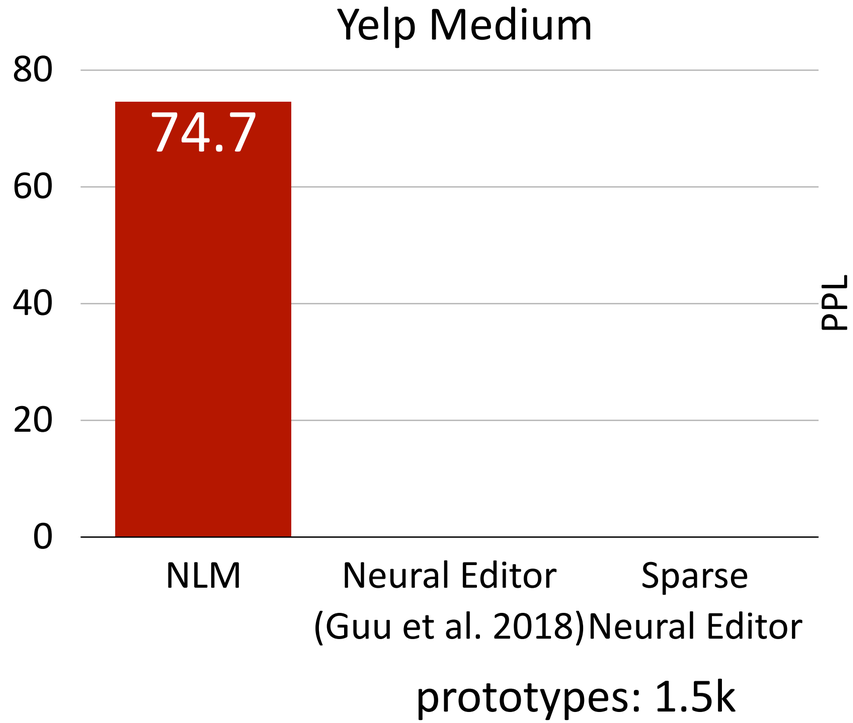


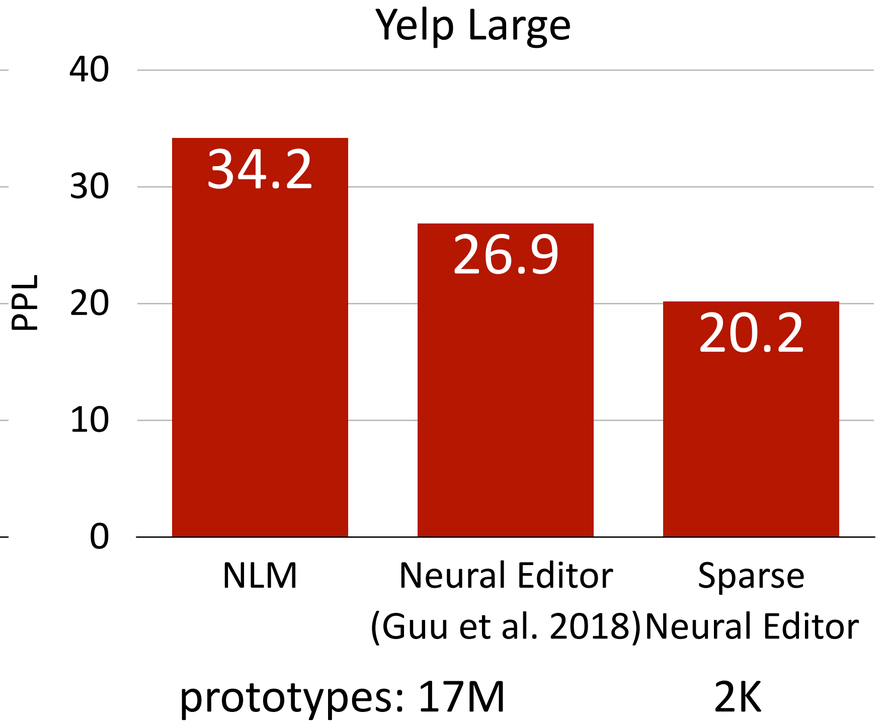


Language Technologies Institute

Perplexity Results

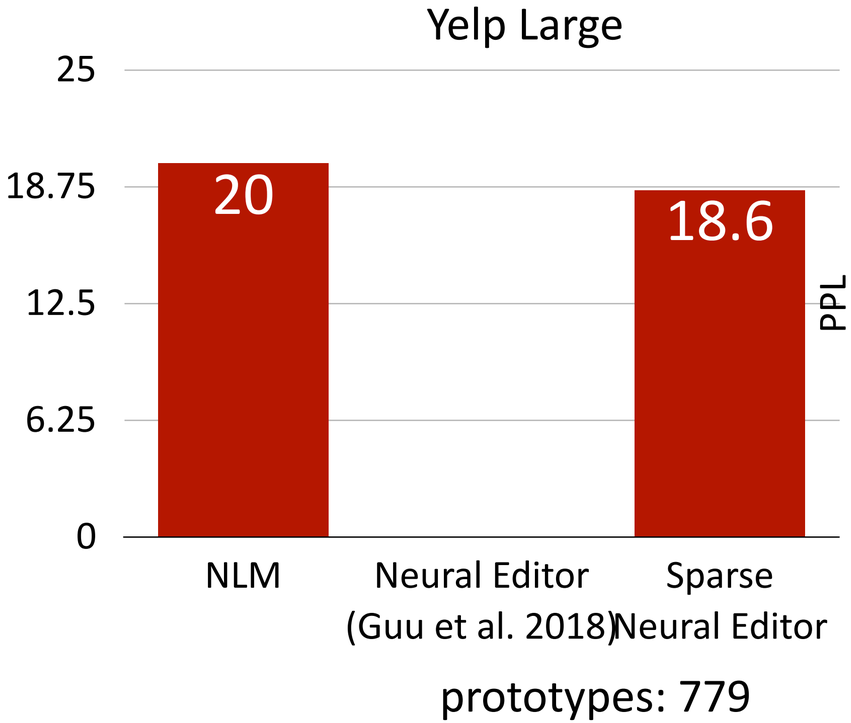




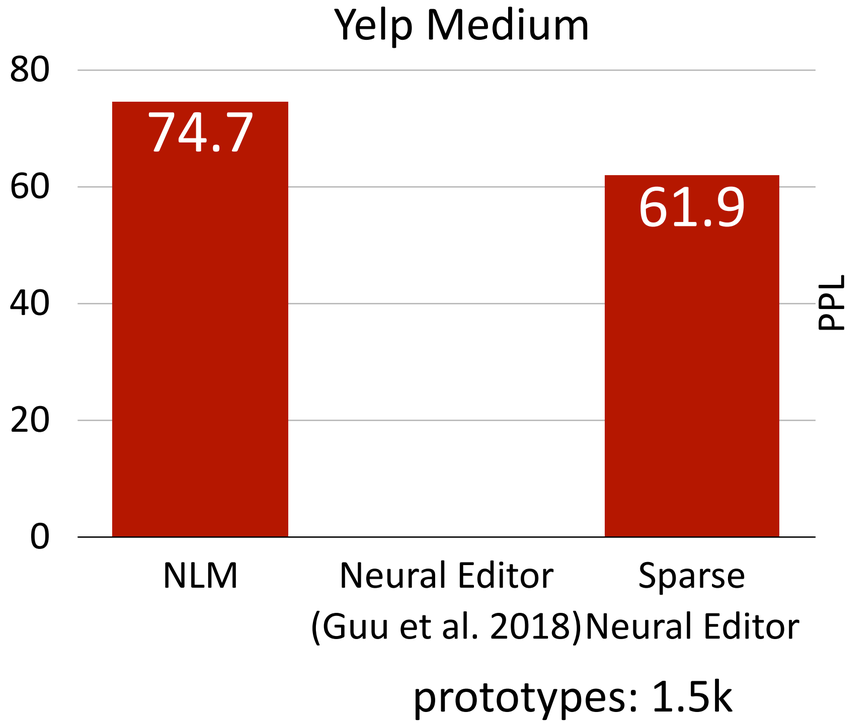


Language Technologies Institute

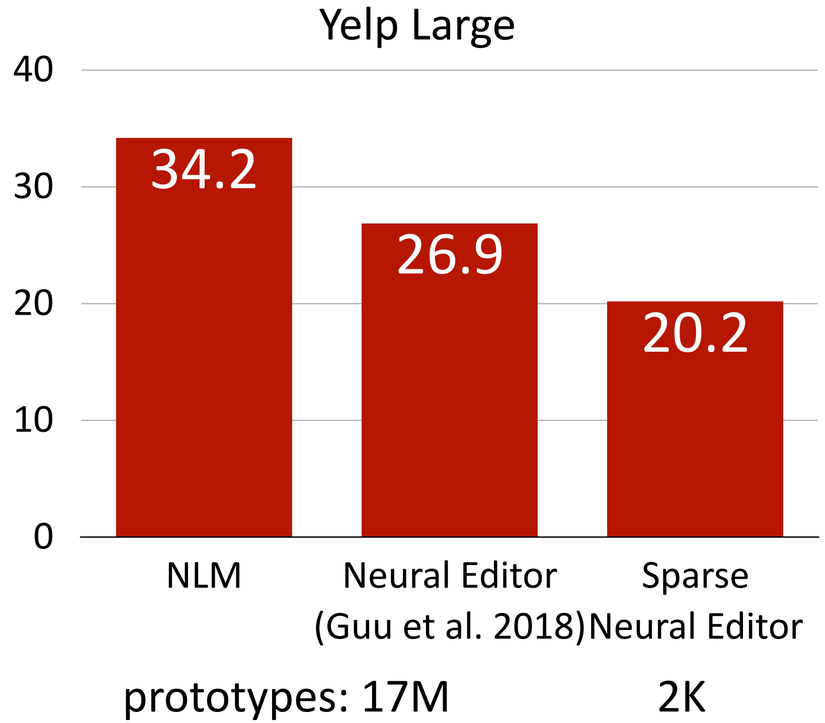
Perplexity Results



prototypes: 779



prototypes: 1.5k



prototypes: 17M

2K

Language
Technologies
Institute

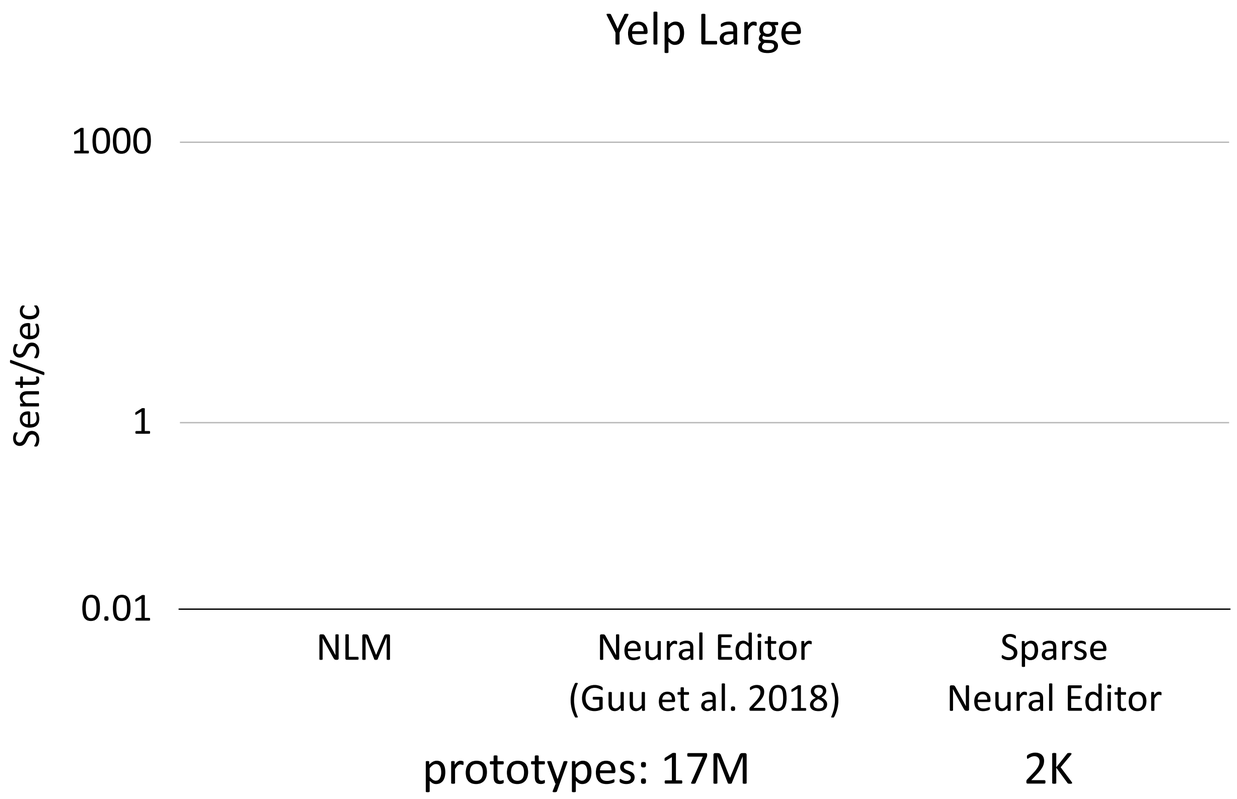
Processing Speed Results

Yelp Large

prototypes: 17M 2K

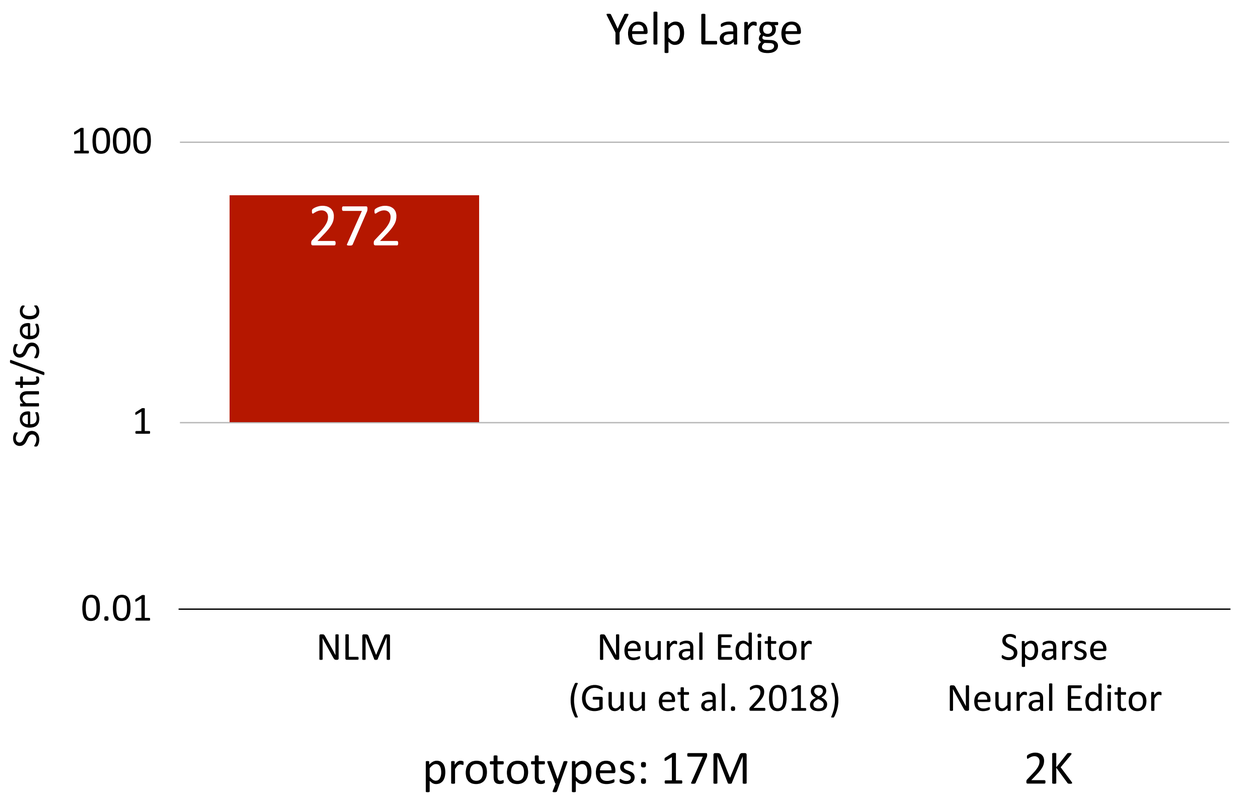
Language Technologies Institute

Processing Speed Results



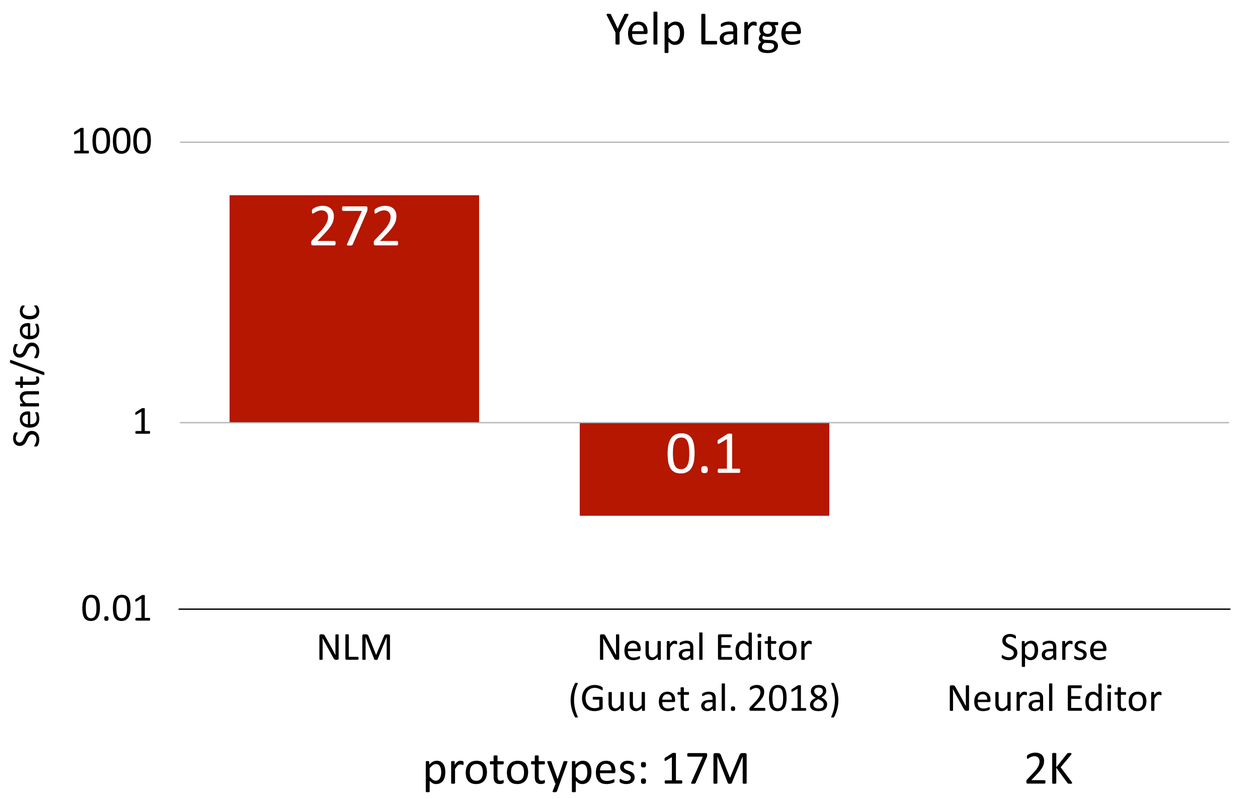
Language Technologies Institute

Processing Speed Results



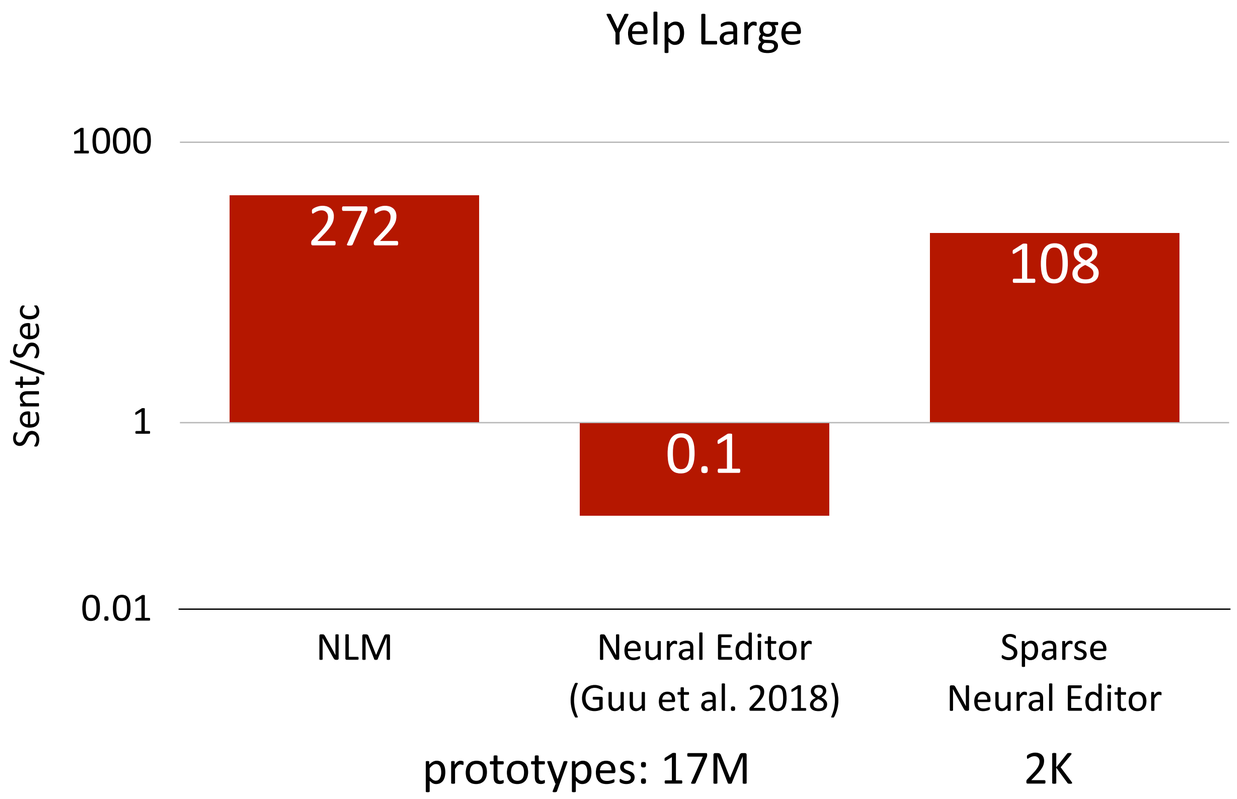
Language Technologies Institute

Processing Speed Results



Language Technologies Institute

Processing Speed Results



Language Technologies Institute

How Does Sparsity Affect Prototypes?

Dense prototypes see more semantic overlap, sparse prototypes are more syntactic.

Data Examples	Prototypes
the best corned beef hash i 've ever had !	(dense) the best real corned beef hash i 've had . (sparse) the chicken satay is the best i 've ever had .
the grilled chicken was flavorful , but too flavorful .	(dense) the chicken was moist but it lacked flavor . (sparse) my sandwich was good but the chicken was a little plain .
i asked her what time they close and she said <cardinal> o'clock .	(dense) i asked what time they closed <date> , and was told <cardinal> . (sparse) we asked how long the wait was and we were informed it would be <time> .

Table 2: Qualitative examples of prototypes when using denser and sparser prototype supports.

Language Technologies Institute

How Does Sparsity Affect Prototypes?

Dense prototypes see more overlapping content words.

Model	Overall	NOUN	DET	AUX	PRON	ADJ	VERB
31K prototypes	91.2K	14.4K	9.6K	9.3K	9.0K	7.2K	6.4K
1.5K prototypes	74.7K	9.9K	8.5K	8.2K	7.3K	5.6K	4.4K
Relative Change	-18.1%	-31.3%	-11.5%	-11.8%	-18.9%	-22.2%	-31.3K

Table 1: Number of matching tokens between examples and their prototypes. Results are reported in cluster of POS tags.

Language
Technologies
Institute

Example Generation Results

Prototype: A man is using a small laptop computer	Prototype: A cat sitting on a sidewalk behind a bush
A man is using his laptop computer with his hands on the keyboard	A cat laying on top of a wooden bench
A man is using a laptop computer with his hands on the keyboard	A cat standing next to a tree in a park
A man is using a laptop computer while sitting on a bench	Two cats sitting on a bench near a park bench
A man is using a laptop computer in the middle of a room	A dog sitting on a bench near a park bench
A young man is using a laptop computer in the middle of a room	A dog sitting on a bench near a park bench

Language
Technologies
Institute

Conclusion

Language
Technologies
Institute

Summary

- Prototype support can be sparse to improve efficiency — superior over previous neural editor model with only 1‰ of datastore and 1‰ of test time
- Prototypes tend to drop semantic but keep syntactic information when they grow sparser

Language
Technologies
Institute

Future Directions?

- **MANY** other promising non-parametric models! How do they play with sparsity?
- What is the interaction between big non-parametric models and small parametric models? Do we need 100 billion parameters if we have Google search?

Code available, try it out!

https://github.com/jxhe/sparse-text-prototype