Generalization through Memorization: Nearest Neighbor Language Models

Authors: Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, Mike Lewis

Stanford University, Facebook AI Research

ICLR 2020

Presented by: Itay Levy

* Some figures from https://urvashik.github.io/pdfs/knnlm_iclr_final.pdf

Talk outline

kNN-LM

- Motivation
- Constructing the datastore
- Inference

Experiment setting

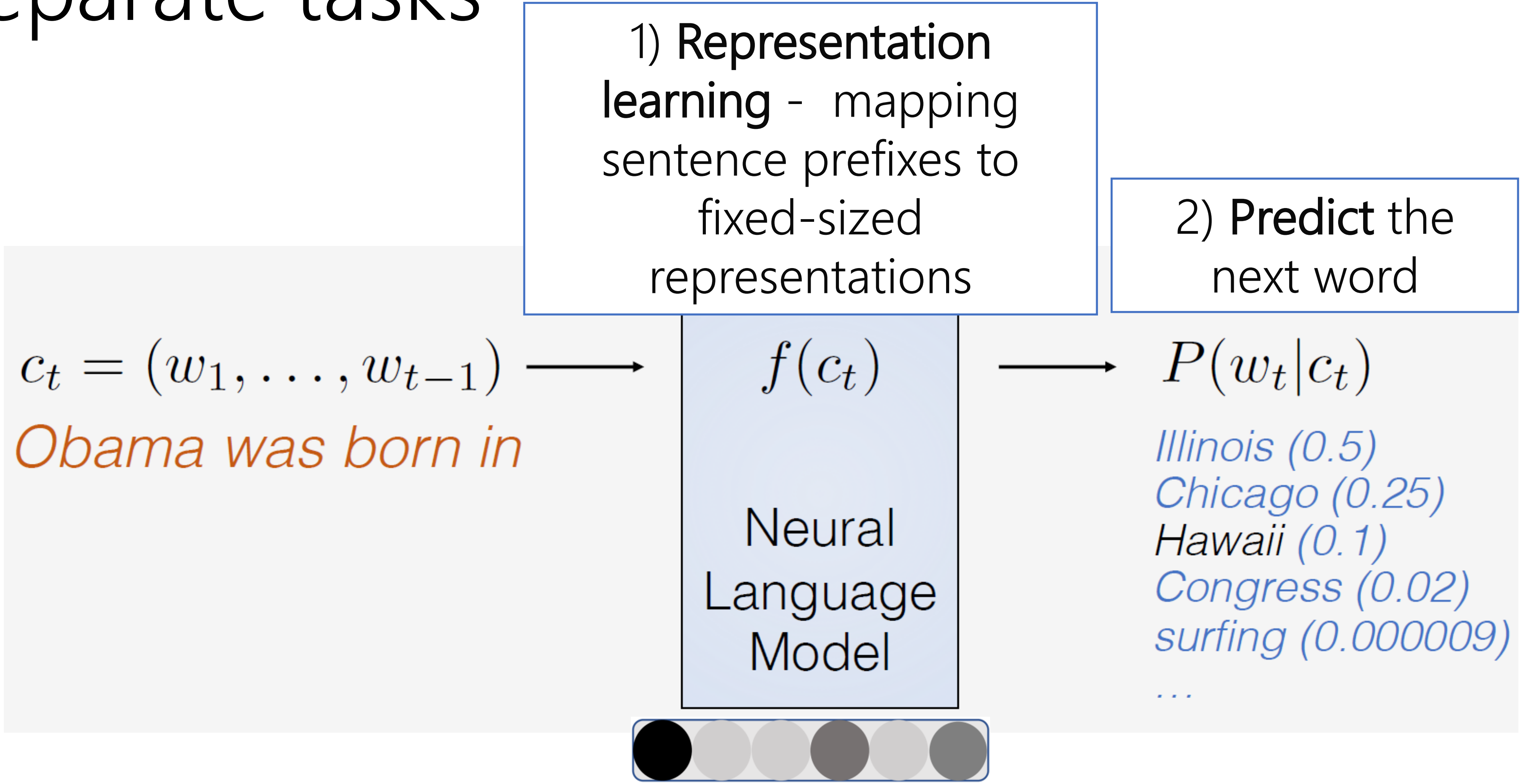
Key results

- Improved generalization w/o adding parameters
- Scaling up to larger training sets using the datastore
- Domain adaptation

Summary

- Maybe also some additional stuff

Motivation: Autoregressive LM as two separate tasks



Motivation: Hypotheses

1. **Representation learning may be easier for LM than the prediction problem**

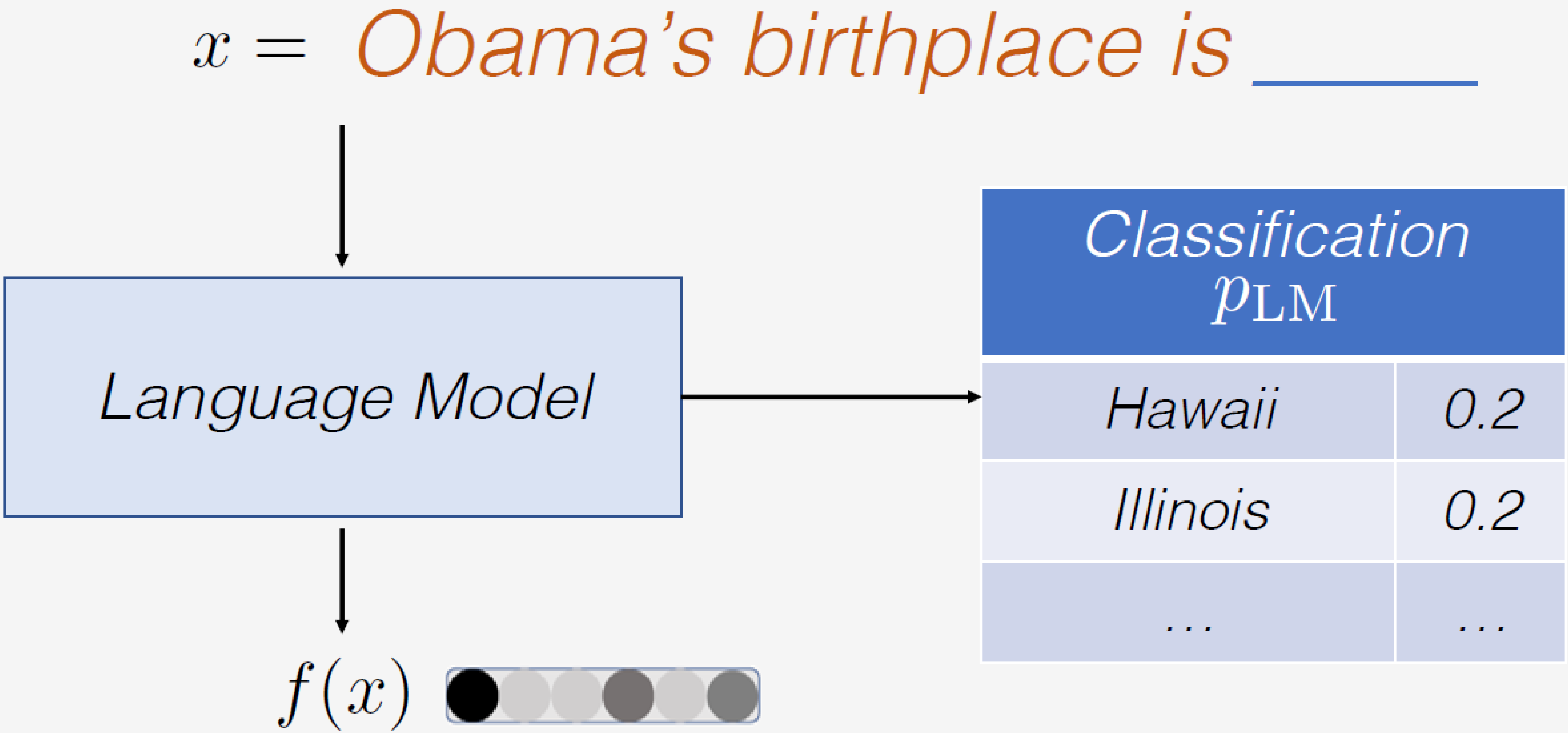
2. Context representation function is a similarity function.
   - Contexts which are close in representation space are more likely to be followed by the same target word.
   - E.g. *"Dickens is the author of"* ~ *"Dickens wrote"*
   - *Seen before in the seminar* - in openQA we compare the query representation to the passage representation

Motivation: finding close contexts in representation space

Test Context: Obama's birthplace is _???_

Previously Seen Contexts	*Targets*
Obama was senator for	*Illinois*
Barack is married to	*Michelle*
Obama was born in	*Hawaii*
...	...
Obama is a native of	*Hawaii*

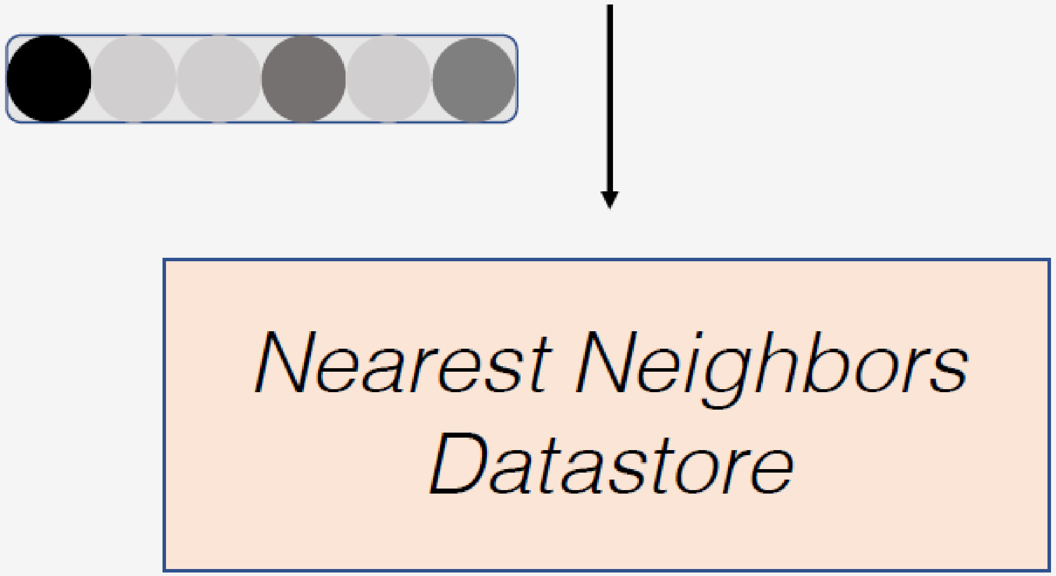
kNN-LM motivation



kNN-LM motivation

$x =$ *Obama's birthplace is* ______

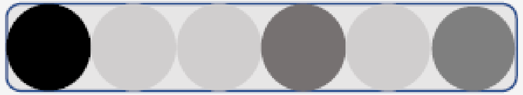
$q = f(x) =$



Focus on neighbors for inference efficiency (like in openQA)

kNN-LM motivation

$x =$ *Obama's birthplace is* ______

$q = f(x) =$ 

Keys	Values
f(Obama was senator for)	*Illinois*
f(Obama was born in)	*Hawaii*
$\ldots$	$\ldots$

Talk outline

kNN-LM

- Motivation
- Constructing the datastore
- Inference

Experiment setting

Key results

- Improved generalization w/o adding parameters
- Scaling up to larger training sets using the datastore
- Domain adaptation

Summary

- Maybe also some additional stuff

Constructing the datastore

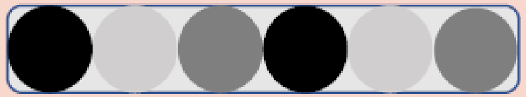
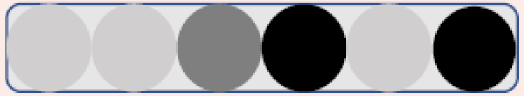
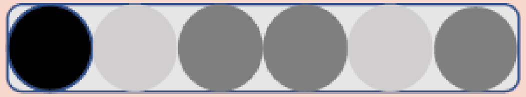
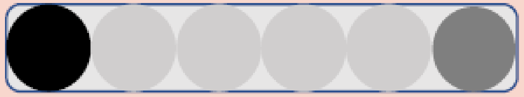
The text collection for the NN datastore can be the original LM training data or a different dataset

Training Contexts c_i	*Targets* v_i
Obama was senator for	*Illinois*
Barack is married to	*Michelle*
Obama was born in	*Hawaii*
...	...
Obama is a native of	*Hawaii*

Constructing the datastore

Create entry for each training set token

Datastore

	Keys	Values
Training Contexts c_i	*Representations* $k_i = f(c_i)$	*Targets* v_i
Obama was senator for		*Illinois*
Barack is married to		*Michelle*
Obama was born in		*Hawaii*
...	...	...
Obama is a native of		*Hawaii*

Constructing the datastore - efficiency

Task	Time for Wikitext-103 [hours]	Hardware
[reference] A single epoch of LM training forward + backward pass	5	1 GPU
A single forward pass over the dataset to save keys/values	4	1 GPU(+SSD storage)
Creating the datastore using FAISS	2	1 CPU(+SSD storage)

Time complexity - roughly like 1 epoch of training

Highly parallelizable method

Datastore implementation details

How large is it?

- Key for every token
- 4000 bytes for each key
- **~400GB of storage**

FAISS -open source library for fast nearest neighbor retrieval in high dimensional spaces

- Reduces I/O
    - Clusters the keys
    - Search based on cluster centroids
- Reduces storage usage
    - Stores compressed versions of the vectors (64 bytes)

Talk outline

kNN-LM

- Motivation
- Constructing the datastore
- Inference

Experiment setting

Key results

- Improved generalization w/o adding parameters
- Scaling up to larger training sets using the datastore
- Domain adaptation

Summary

- Maybe also some additional stuff

Inference – neighbors retrieval

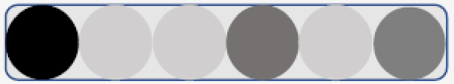
The k-nearest neighbors for $q = f(x)$

Representations $k_i = f(c_i)$	Targets v_i	Distances $d_i = d(q, k_i)$
	Illinois	*4*
	Michelle	*100*
	Hawaii	*3*
$\dots$	$\dots$	$\dots$
	Hawaii	*5*

$K = 1024$

Inference – choosing nearest neighbors

The k-nearest neighbors for $q = f(x)$



Targets v_i
Illinois
Michelle
Hawaii
...
Hawaii

Distances $d_i = d(q, k_i)$
4
100
3
...
5

Nearest k	
Hawaii	3
Illinois	4
Hawaii	5

Inference – kNN distribution

Nearest k	
Hawaii	*3*
Illinois	*4*
Hawaii	*5*

Normalization $p(k_i) \propto exp(-d_i)$	
Hawaii	*0.7*
Illinois	*0.2*
Hawaii	*0.1*

Aggregation $p_{\mathrm{kNN}} = \sum_i \mathbb{I}_{y=v_i} p(k_i)$	
Hawaii	*0.8*
Illinois	*0.2*

Softmax over negative distances

Inference – interpolation with the original model

$x =$ *Obama's birthplace is* ______

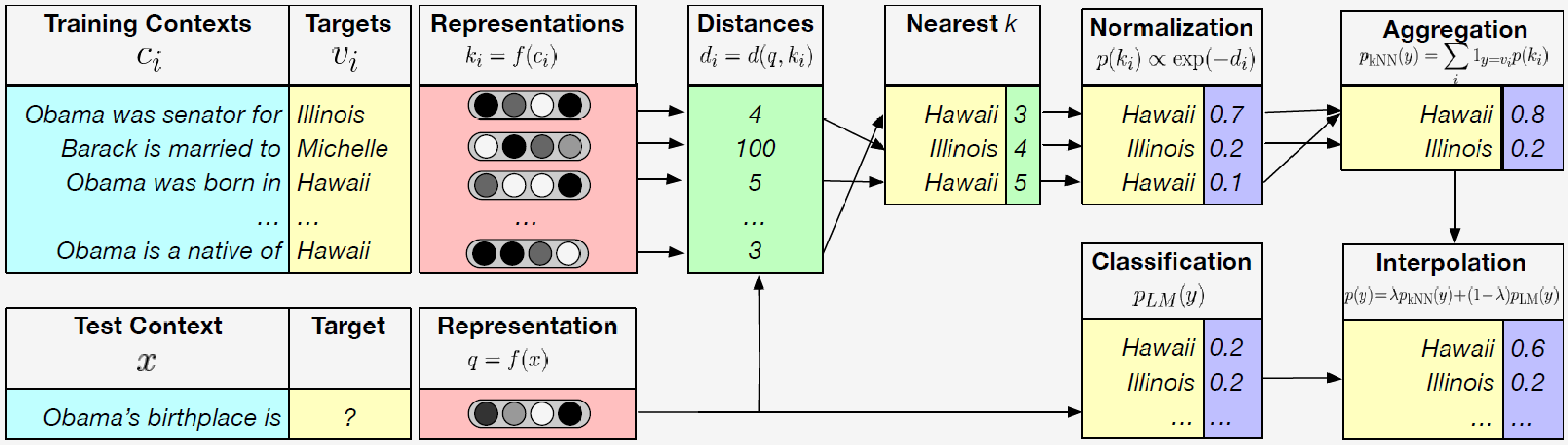
Language Model	
Hawaii	*0.2*
Illinois	*0.2*
…	…

k-Nearest Neighbors	
Hawaii	*0.8*
Illinois	*0.2*

→

kNN-LM $p \; (1-\lambda)\, p_{\mathrm{LM}} + \lambda\, p_{\mathrm{kNN}}$	
Hawaii	*0.6*
Illinois	*0.2*
…	…

Inference – complete illustration



Highly interpretable

Inference - efficiency

Language model	Decoding speed on a single GPU [tokens/sec]
Vanilla LM	500
kNN-LM	60

Talk outline

kNN-LM

- Motivation
- Constructing the datastore
- Inference

Experiment setting

Key results

- Improved generalization w/o adding parameters
- Scaling up to larger training sets using the datastore
- Domain adaptation

Summary

- Maybe also some additional stuff

Datasets

Dataset	Description	#Tokens
WIKITEXT-103	**Standard benchmark for autoregressive language modelling** Wikipedia articles	100M
BOOKS (Toronto Books Corpus)	10K books in 16 different genres	1B
WIKI-3B	English Wikipedia articles	3B
WIKI-100M	Random articles subset of WIKI-3B	100M

Model details

- Decoder-only Transformers
  - Architecture and optimization from Baevski & Auli (2019)
  - 16 layers
    - The keys in the datastore are the input to the final layer’s feedforward network
  - 1024 dimensional hidden states – our keys

Evaluation

- Loss function during training
  - negative log-likelihood of the training corpus
- Evaluation metric
  - Perplexity (standard practice)

Metric	Definition	Formula
Likelihood	$P(D)$	$\prod_{y_i \in D} P(y_i \mid Y_{<i})$
Log-likelihood	$\log P(D)$	$\sum_{y_i \in D} \log P(y_i \mid Y_{<i})$
Cross Entropy	$-\frac{1}{\lvert D \rvert} \log P(D)$	$-\frac{1}{\lvert D \rvert} \sum_{y_i \in D} \log P(y_i \mid Y_{<i})$
Perplexity	$\left(P(D)\right)^{-\frac{1}{\lvert D \rvert}}$	$\exp\left(-\frac{1}{\lvert D \rvert} \sum_{y_i \in D} \log P(y_i \mid Y_{<i})\right)$

Credit: Advanced Methods in NLP course by Omer Levy

Qualitative Analysis
Memorizing rare information

The model is specifically helpful in predicting rare patterns

- factual knowledge
- Names
- near-duplicate sentences from the training set

Test Context ($p_{\text{kNN}} = 0.998$, $p_{\text{LM}} = 0.124$)	**Test Target**	
it was organised by New Zealand international player Joseph Warbrick, promoted by civil servant Thomas Eyton, and managed by James Scott, a publican. The Natives were the first New Zealand team to perform a haka, and also the first to wear all black. They played 107 rugby matches during the tour, as well as a small number of Victorian Rules football and association football matches in Australia. Having made a significant impact on the...	development	
Training Set Context	**Training Set Target**	**Context Probability**
As the captain and instigator of the 1888-89 Natives – the first New Zealand team to tour the British Isles – Warbrick had a lasting impact on the...	development	0.998
promoted to a new first grade competition which started in 1900. Glebe immediately made a big impact on the...	district	0.00012
centuries, few were as large as other players managed. However, others contend that his impact on the...	game	0.000034
Nearly every game in the main series has either an anime or manga adaptation, or both. The series has had a significant impact on the...	development	0.00000092

Figure 6: Example where the kNN model has much higher confidence in the correct target than the LM. Although there are other training set examples with similar local n-gram matches, the nearest neighbour search is highly confident of specific and very relevant context.

Talk outline

kNN-LM

Experiment setting

Key results

- Improved generalization w/o adding parameters
- Scaling up to larger training sets using the datastore
- Domain adaptation

Summary

- Maybe also some additional stuff

Using the training data as the datastore

Model	*Perplexity*
Previous Best (Luo et al., 2019)	*17.40*
Base LM	*18.65*
kNN-LM	*16.12*

Very significant result!

Key result #1
Improved generalization w/o adding parameters

- Compatible with any autoregressive model
- No added parameters*
- No additional training
  - Takes advantage of effective similarity functions learned by LM

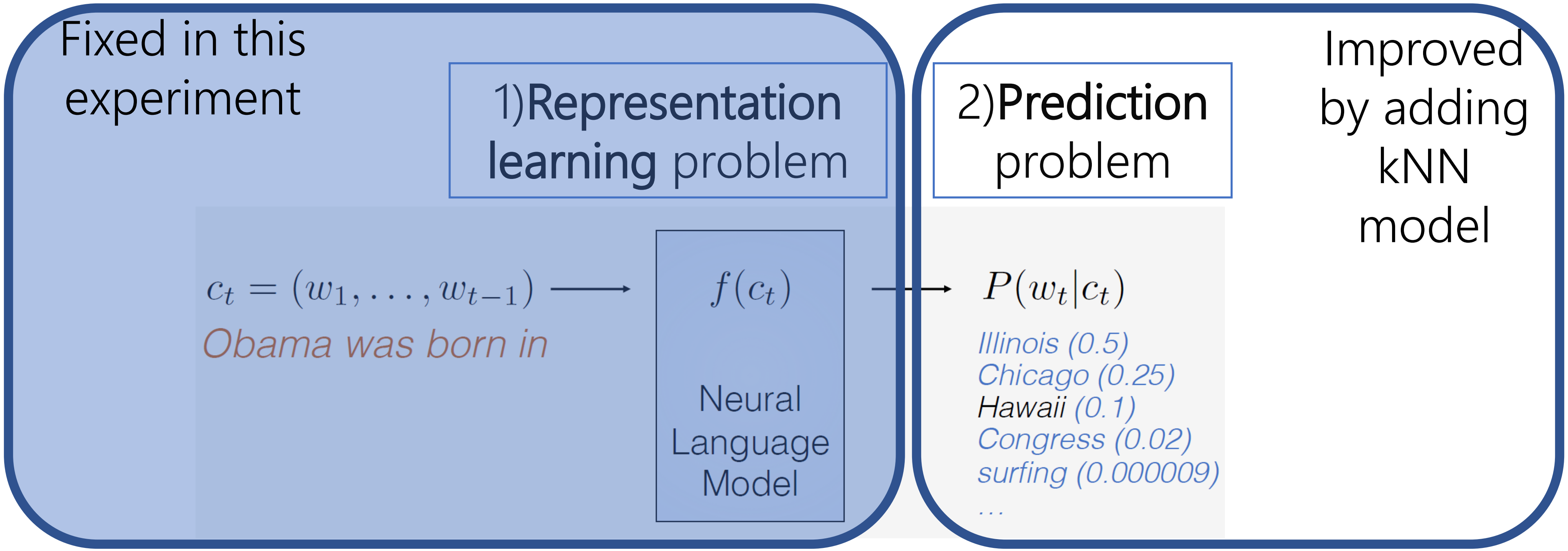
Is it just ensembling with an overfitted model?

Memorizing Transformer – Transformer LM that perfectly memorized the training data

Interpolating original LM with:	Validation perplexity improvement ↑
kNN-LM	1.9
Memorizing Transformer	0.1

Implicit memorization is less effective at generalization than kNN-LM

Back to the hypotheses



1. <u>Prediction problem is harder when done implicitly using LM parameters</u>
2. Contexts which are close in representation space are more likely to be followed by the same target word

Talk outline

kNN-LM

- Motivation
- Constructing the datastore
- Inference

Experiment setting

Key results

- Improved generalization w/o adding parameters
- Scaling up to larger training sets using the datastore
- Domain adaptation

Summary

- Maybe also some additional stuff

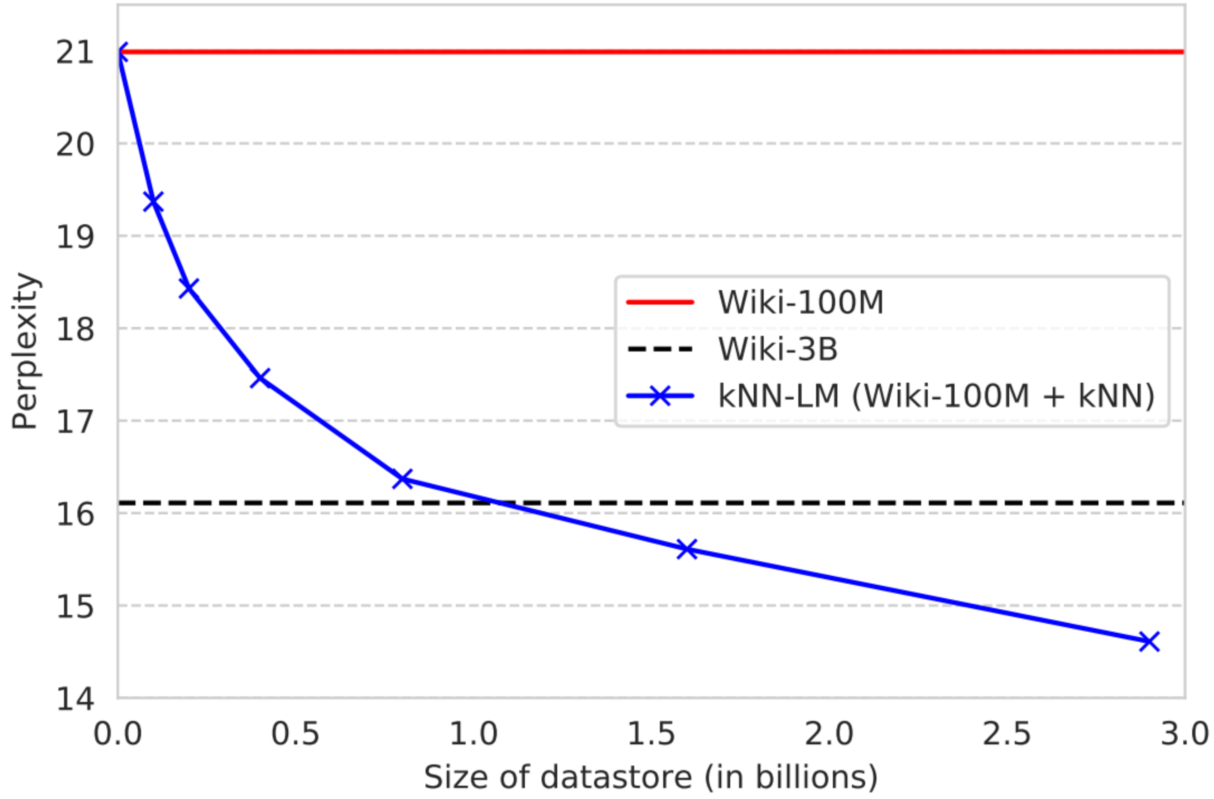
What should I do with extra data?

LM Training Data	*Datastore*	*Perplexity*
Wiki-3B	-	*15.17*
Wiki-100M	-	*19.59*
Wiki-100M	*Wiki-3B*	*13.73*

Test on Wiki-3B

Retrieving nearest neighbors from the corpus outperforms training on it!

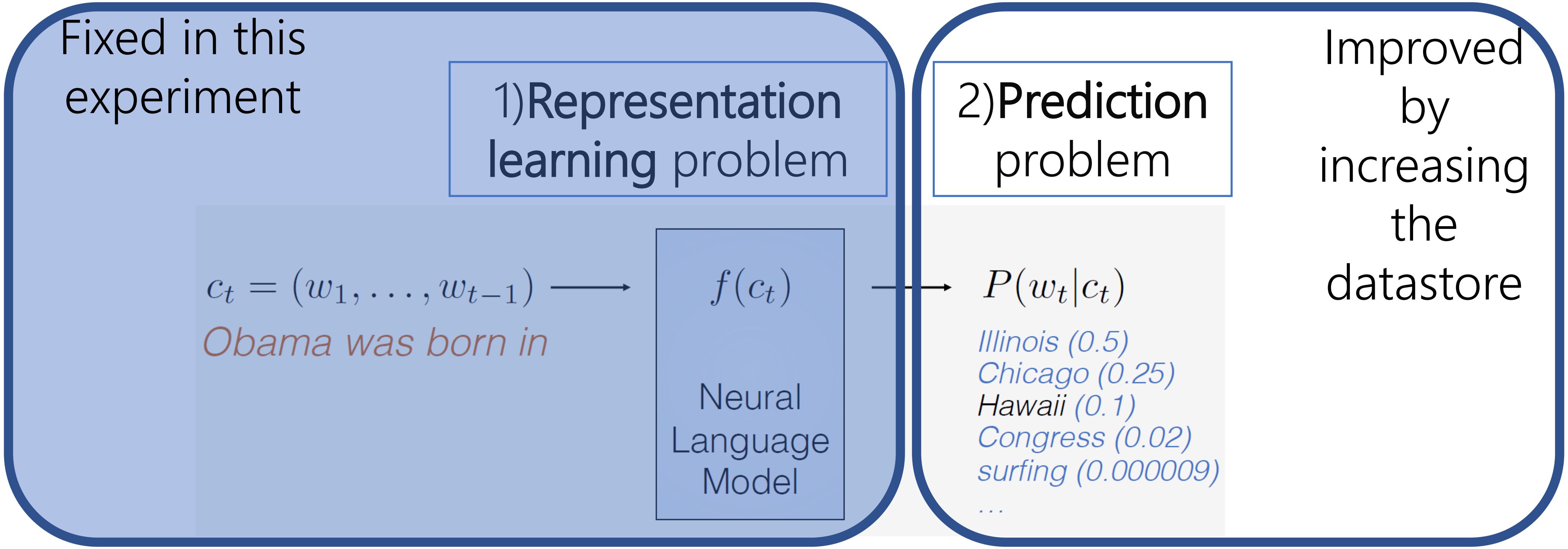
What should I do with extra data?



(a) Effect of datastore size on perplexities.

kNN-LM trained on 100M tokens with a datastore of 1.6B tokens already outperforms the LM trained on all 3B tokens

Back to the hypotheses



<u>Prediction problem is harder when done implicitly using LM parameters</u>

Possible implication?

Massive LM (like GPT-3) are not that better at finding better representations

They are better at prediction by implicitly memorizing more data in parameters

Credit: Advanced Methods in NLP course by Omer Levy

Key result #2
Scaling up to larger training sets using the datastore

- Retrieving nearest neighbors from the corpus outperforms training on it

- New path for efficiently using large datasets in LMs
  - No additional cost of training
  - Just increasing the datastore

- Problem
  - Larger datastore -> slower inference

Talk outline

kNN-LM

- Motivation
- Constructing the datastore
- Inference

Experiment setting

Key results

- Improved generalization w/o adding parameters
- Scaling up to larger training sets using the datastore
- Domain adaptation

Summary

- Maybe also some additional stuff

Domain adaptation

- Varying the NN datastore, again without further training
  - Adding out-of-domain data to the datastore makes a single LM useful across multiple domains

LM Training Data	*Datastore*	*Perplexity on Books*
Books	-	*11.89*
Wiki-3B	-	*34.84*
Wiki-3B	*Books*	*20.47*

Domain adaptation requires more weight on kNN component than in-domain

Key result #3
Domain adaptation

- A single LM can adapt to multiple domains without the in-domain training
- We can domain-specific data to the datastore

Talk outline

kNN-LM

- Motivation
- Constructing the datastore
- Inference

Experiment setting

Key results

- Improved generalization w/o adding parameters
- Scaling up to larger training sets using the datastore
- Domain adaptation

Summary

- Maybe also some additional stuff

Summary

- Explicitly memorizing the training data helps generalization
  - Suggests that prediction problem is harder that representation learning
  - Takes advantage of effective similarity functions learned by LM

- Enables:
  - Compatibility to any autoregressive LM
  - Smaller models trained on smaller datasets
    - Quicker training
  - Adaptability to other domains

Future work

- Explicitly training similarity functions.
- Reducing the size of the datastore.
- Keep training the decoding layers while freezing the encoding part that kNN-LM uses.
- Extend the proposed model to other language tasks.
- Test on various language models to verify the generalization ability across different models.

Result #1 on multiple domains

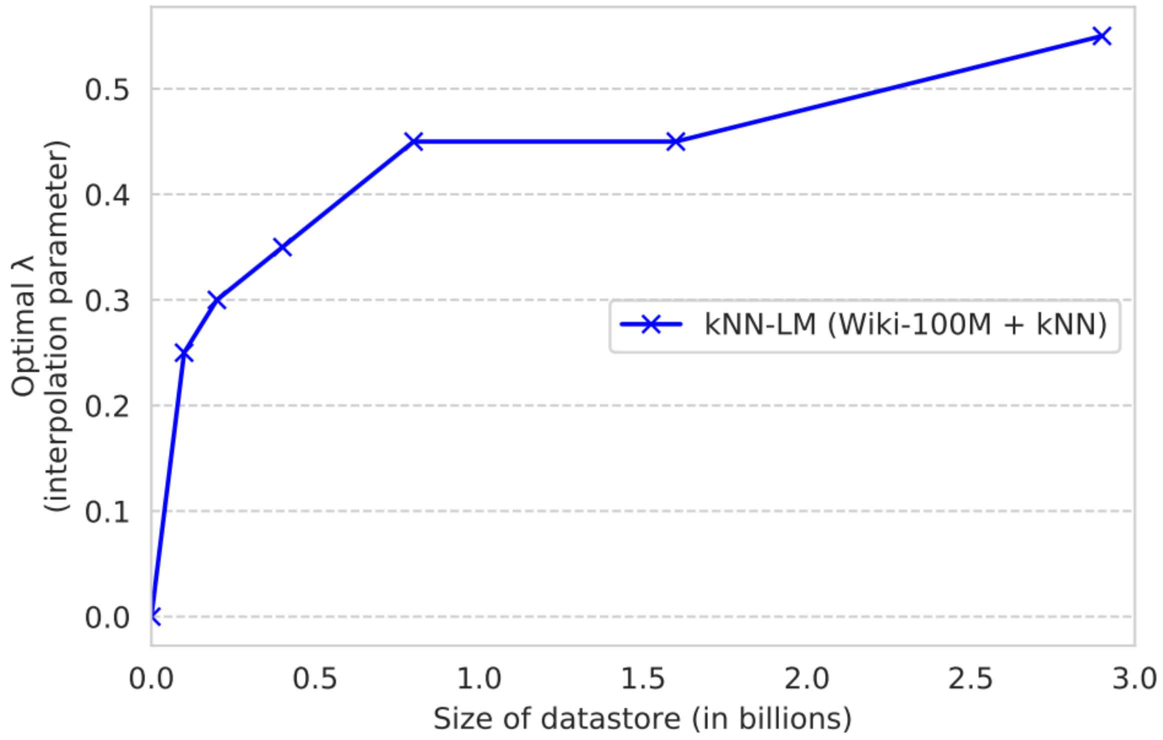
Control for the possibility that encyclopedic Wikipedia text is somehow uniquely good for caching.

Model	**Perplexity (↓)**		**# Trainable Params**
	Dev	Test	
Base LM (Baevski & Auli, 2019)	14.75	11.89	247M
+kNN-LM	**14.20**	**10.89**	247M

Table 2: Performance on BOOKS, showing that kNN-LM works well in multiple domains.

Finding optimal λ Datastore size

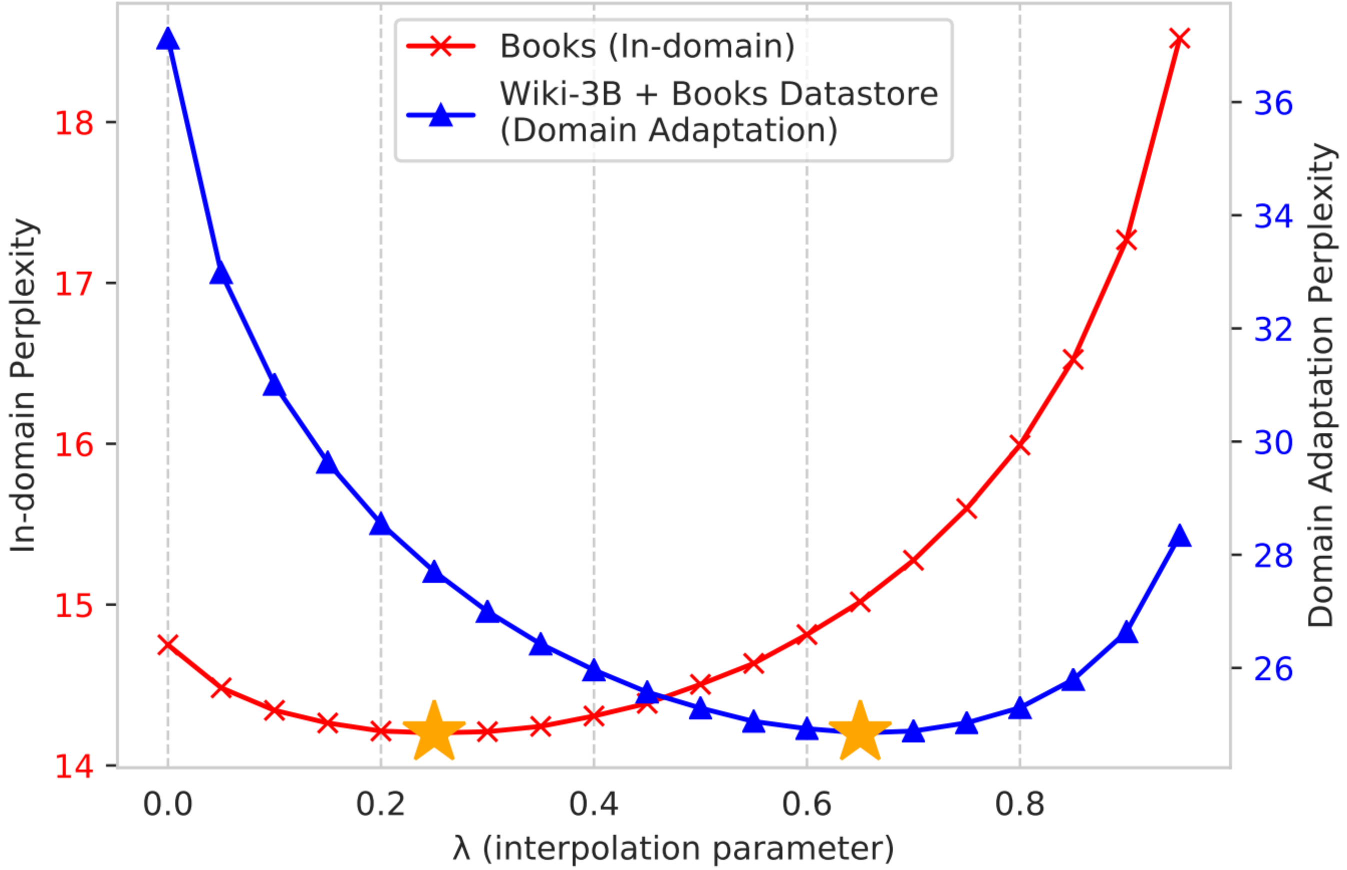
The optimal value of λ increases with the size of the datastore.



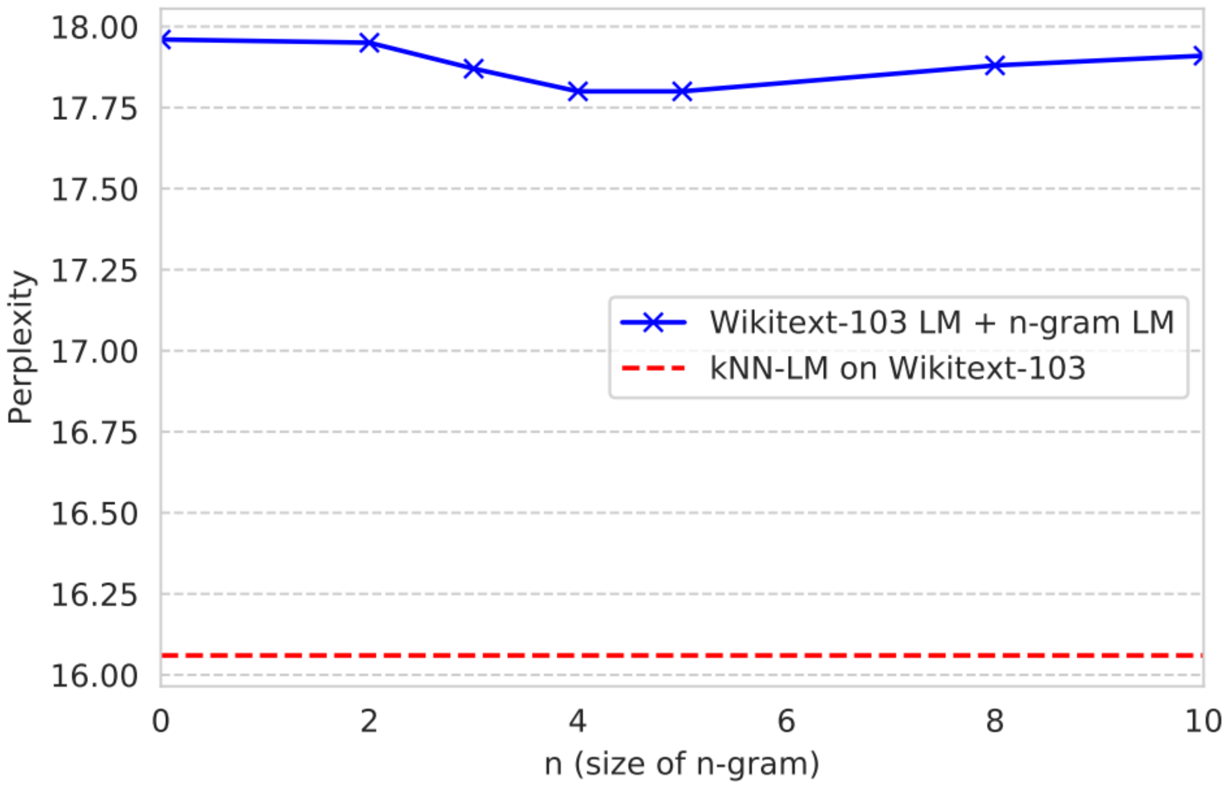
(b) Tuned values of λ for different datastore sizes.

Finding optimal λ
in domain vs. out of domain

More weight on kNN LM improves domain adaptation.



kNN(neural representation) vs n-gram (simple representation)



Highlights the need to use the learned representation function f(·) to measure similarity between more varied contexts