@ukhndlwl

Generalization through Memorization: Nearest Neighbor Language Models

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

Stanford University, Facebook AI Research

Stanford | NLP

facebook Artificial Intelligence

ICLR 2020

Neural Autoregressive Language Models

Given prior context, estimate the probability for the target token

$c_t = (w_1, \ldots, w_{t-1}) \longrightarrow f(c_t) \longrightarrow P(w_t | c_t)$

Obama was born in

Neural Language Model

Illinois (0.5)
Chicago (0.25)
Hawaii (0.1)
Congress (0.02)
surfing (0.000009)
...

Language Models

Lots of text is very easily available, so we train models on large amounts of data.

But improving LM performance or scaling to larger datasets, by training bigger and bigger models with billions of parameters, requires massive amounts of GPU compute..☹

Instead, can explicitly memorizing data make LMs generalize better without the added cost of training?

Nearest Neighbor Language Models

Key Results

Explicitly memorizing the training data helps generalization.

LMs can scale to larger text collections without the added cost of training.

A single LM can adapt to multiple domains without any in-domain training.

Nearest Neighbor Language Models (kNN-LM)

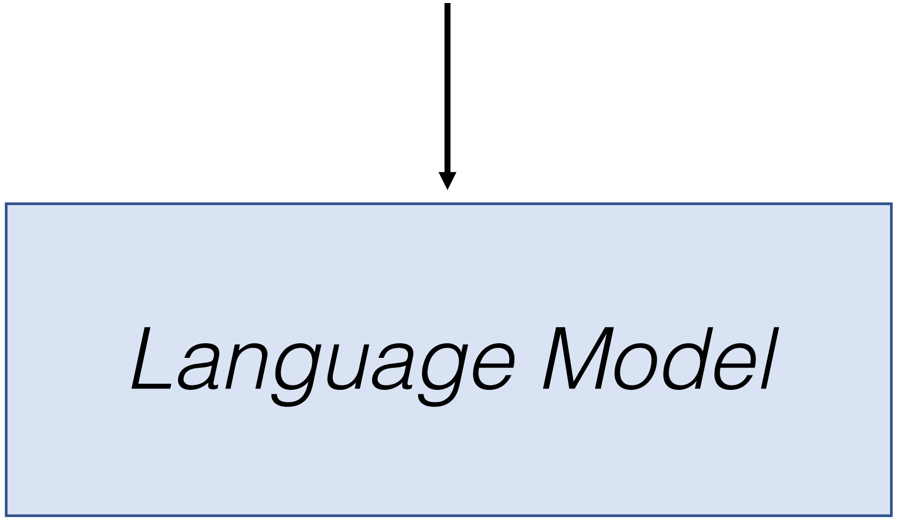
kNN-LM: Intuition

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*

Given a new test context…

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



Given a new test context…

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

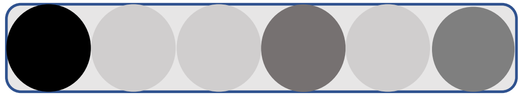
Language Model

Classification p_{LM}	
Hawaii	*0.2*
Illinois	*0.2*
…	…

$f(x)$

Given a new test context…

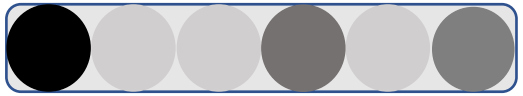
$x =$ *Obama's birthplace is* ______

$q = f(x) =$ 

Nearest Neighbors Datastore

Given a new test context…

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

$q = f(x) =$ 

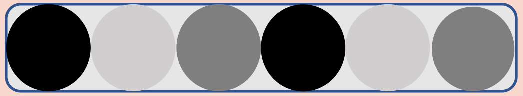
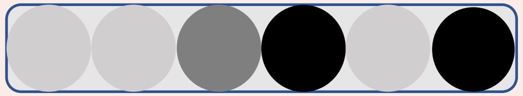
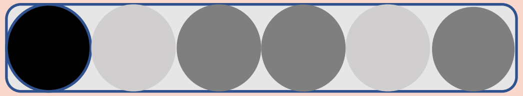
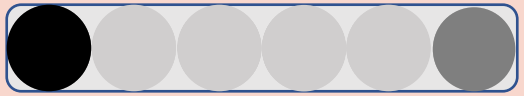
Keys	*Values*
f(Obama was senator for)	*Illinois*
f(Obama was born in)	*Hawaii*
…	…

Constructing the datastore

Constructing the datastore

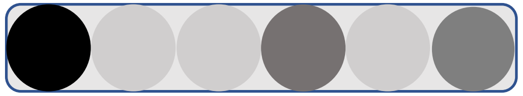
Training Contexts c_i	*Targets* v_i
Obama was senator for	*Illinois*
Barack is married to	*Michelle*
Obama was born in	*Hawaii*
$\ldots$	$\ldots$
Obama is a native of	*Hawaii*

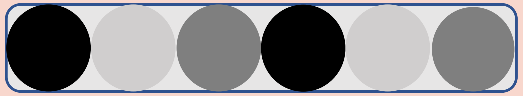
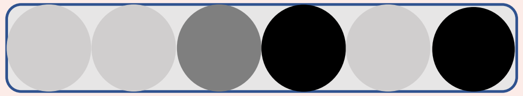
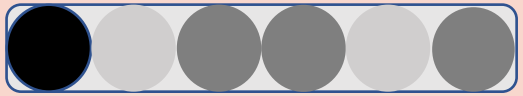
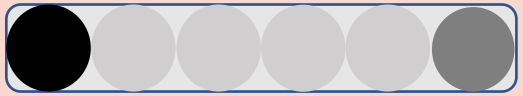
Constructing the datastore

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*
$\dots$	$\dots$	$\dots$
Obama is a native of		*Hawaii*

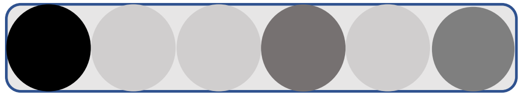
Back to inference!

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*
$\ldots$	$\ldots$	$\ldots$
	*Hawaii*	*5*

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*

The 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*

The 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*

Given a new test context…

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

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

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

→

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

Experiments

Our Base LM is the Transformer LM from Baevski and Auli (2019).

Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. In ICLR, 2019.

Key Results

Explicitly memorizing the training data helps generalization.

LMs can scale to larger text collections without the added cost of training, by simply adding the data to the datastore.

A single LM can adapt to multiple domains without the in-domain training, by adding domain-specific data to the datastore.

Memorizing with Wikitext-103

Standard LM benchmark, 103 million tokens

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

Memorizing with Wikitext-103

Datastore contains 103M examples, $\lambda = 0.25$

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

Memorizing with Wikitext-103

Datastore contains 103M examples, $\lambda = 0.25$

Model	*Perplexity*
Previous Best (Luo et al., 2019)	*17.40*
Base LM	*18.65*
kNN-LM	*16.12*
*kNN-LM + Cont. Cache**	*15.79*

*Edouard Grave, Armand Joulin, and Nicolas Usunier. Improving neural language models with a continuous cache. In ICLR, 2017

Key Results

Explicitly memorizing the training data helps generalization.

LMs can scale to larger text collections without the added cost of training, by simply adding the data to the datastore.

A single LM can adapt to multiple domains without the in-domain training, by adding domain-specific data to the datastore.

Scaling up from Wiki-100M to Wiki-3B

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

Scaling up from Wiki-100M to Wiki-3B

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

Scaling up from Wiki-100M to Wiki-3B

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

Retrieving nearest neighbors from the corpus outperforms training on it!

Key Results

Explicitly memorizing the training data helps generalization.

LMs can scale to larger text collections without the added cost of training, by simply adding the data to the datastore.

A single LM can adapt to multiple domains without the in-domain training, by adding domain-specific data to the datastore.

Domain Adaptation from Wiki to Books

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

Domain Adaptation from Wiki to Books

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

Domain Adaptation from Wiki to Books

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

A single LM can be useful in multiple domains by simply adding a domain-specific datastore!

kNN-LM

Explicitly memorizing the training data helps generalization.

LMs can scale to larger text collections without the added cost of training, by simply adding the data to the datastore.

A single LM can adapt to multiple domains without the in-domain training, by adding domain-specific data to the datastore.

Thanks!

"To make a long story short, what it all boils down to in the final analysis is that what you should take away from this is..."

Explicitly memorizing the training data helps generalization.

LMs can scale to larger text collections without the added cost of training, by simply adding the data to the datastore.

A single LM can adapt to multiple domains without the in-domain training, by adding domain-specific data to the datastore.

Paper:
https://arxiv.org/pdf/1911.00172.pdf

Code:
https://github.com/urvashik/knnlm