



Carnegie Mellon University
Language Technologies Institute

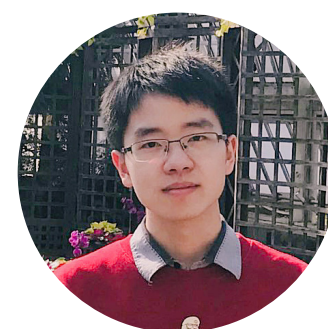
Neuro-Symbolic Language Modeling with Retrieval Automaton

Uri Alon

Language Technologies Institute
Carnegie Mellon University



Frank F. Xu
CMU



Junxian He
CMU



Sudipta Sengupta
Amazon AWS



Dan Roth
AWS AI Labs

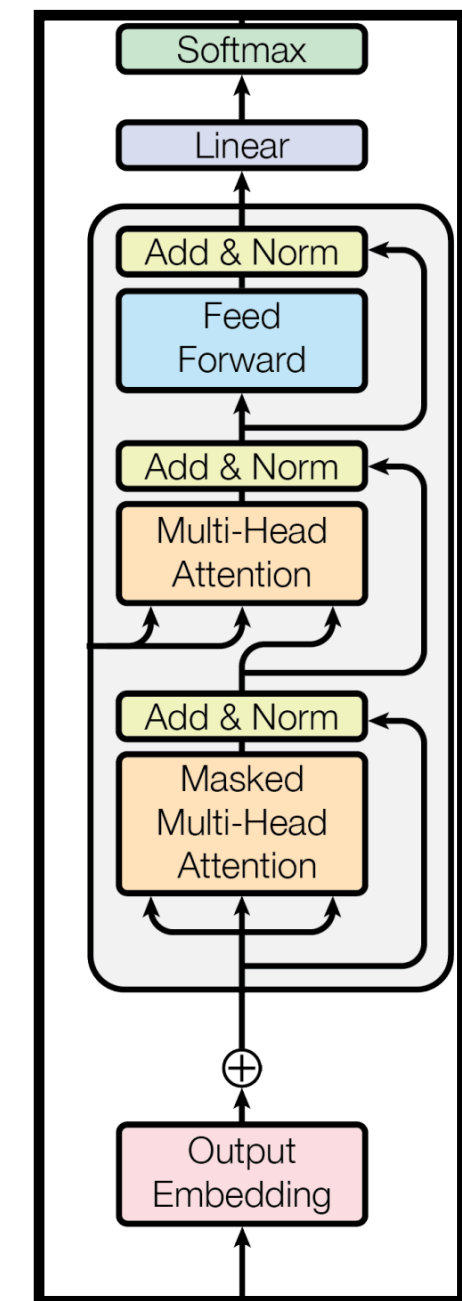


Graham Neubig
CMU

RetoMaton - TL;DR:

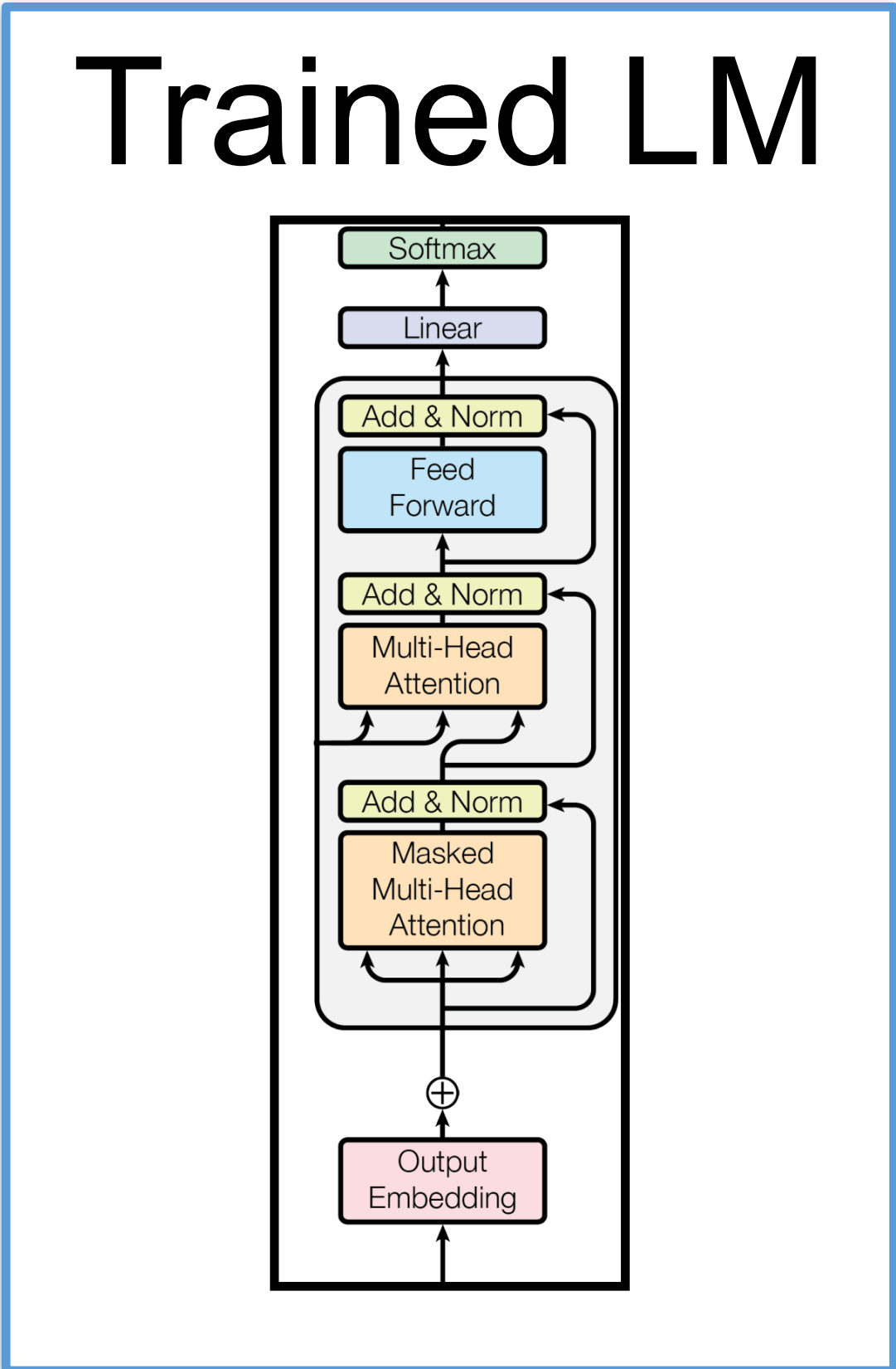
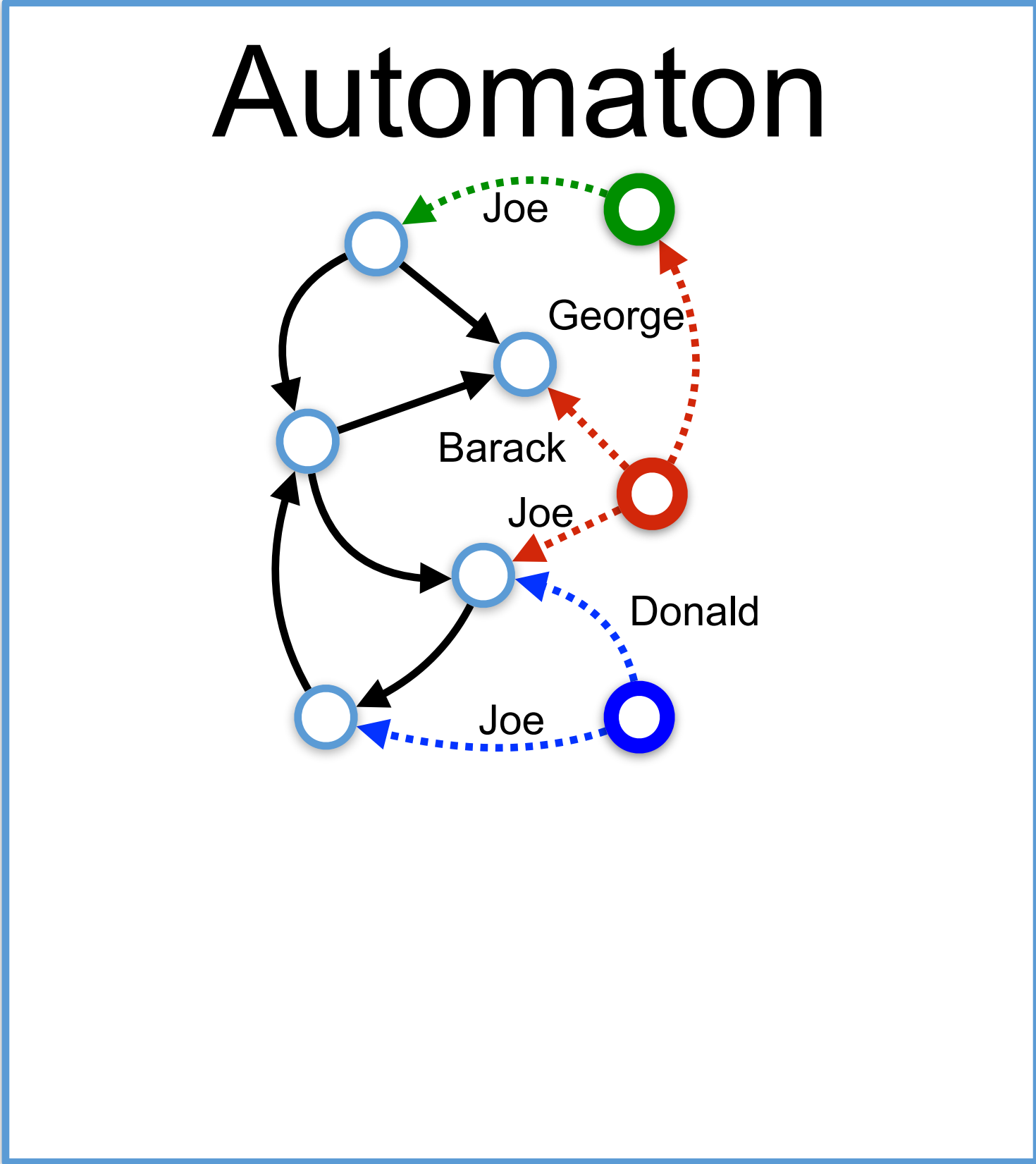
Given a trained LM and its training corpus, we construct a **weighted finite-state automaton**.

Trained LM



RetoMaton - TL;DR:

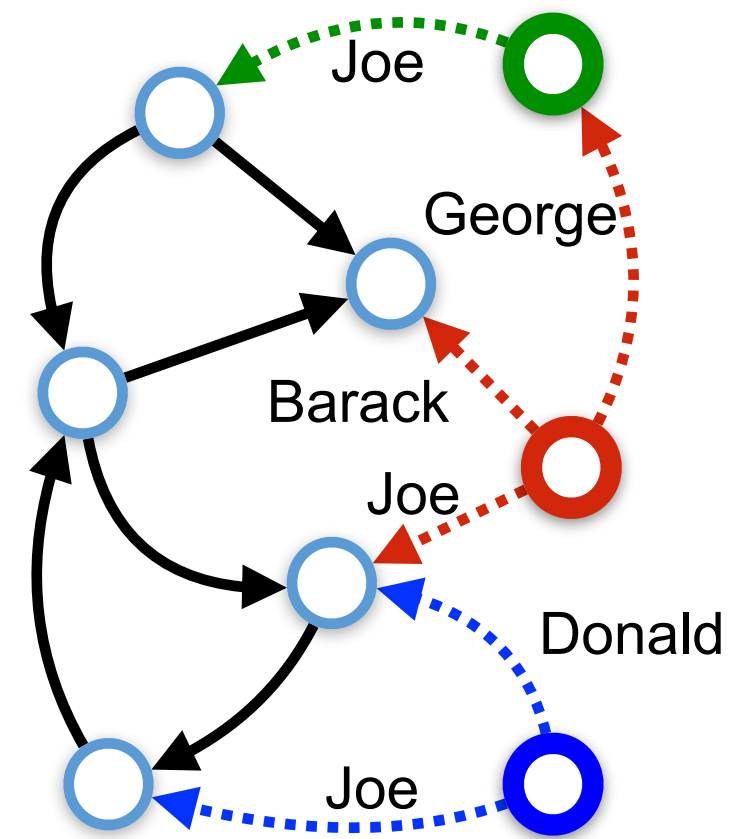
Given a trained LM and its training corpus, we construct a **weighted finite-state automaton**.



RetoMaton - TL;DR:

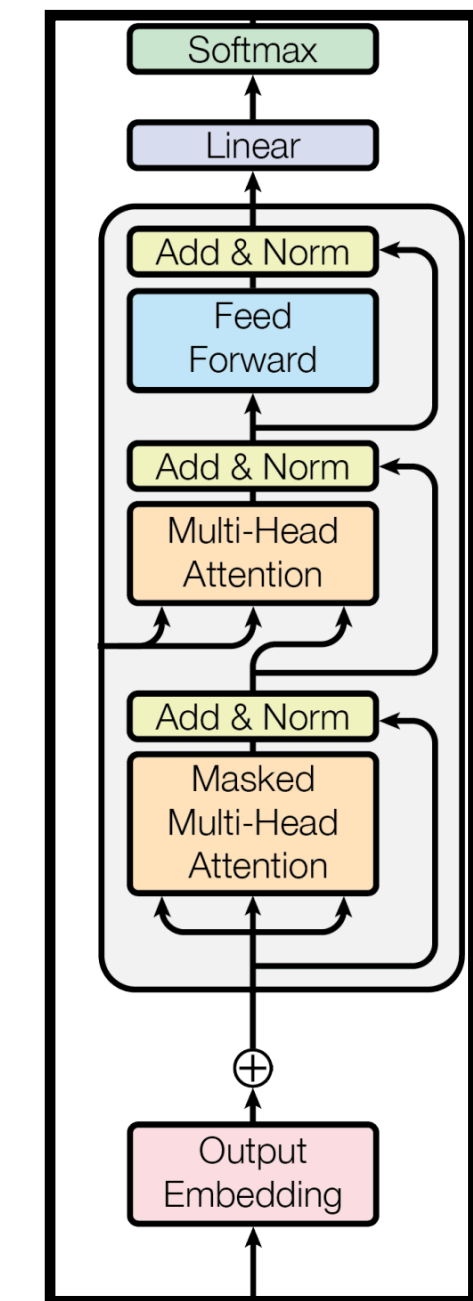
Given a trained LM and its training corpus, we construct a **weighted finite-state automaton**.

Automaton



States: clusters of training examples, encoded by the LM
Edges: pointers between consecutive examples, shared in cluster

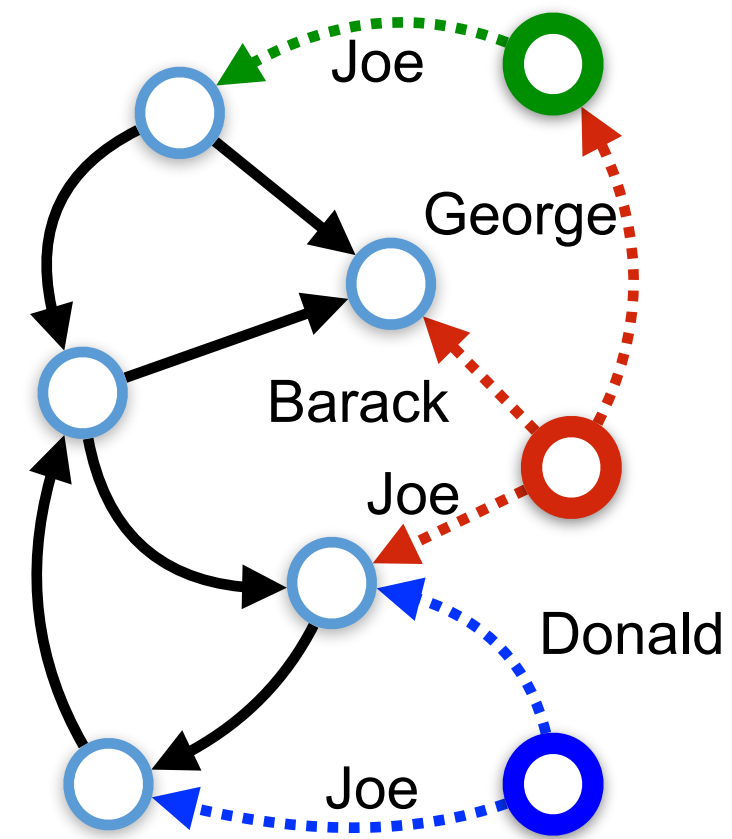
Trained LM



RetoMaton - TL;DR:

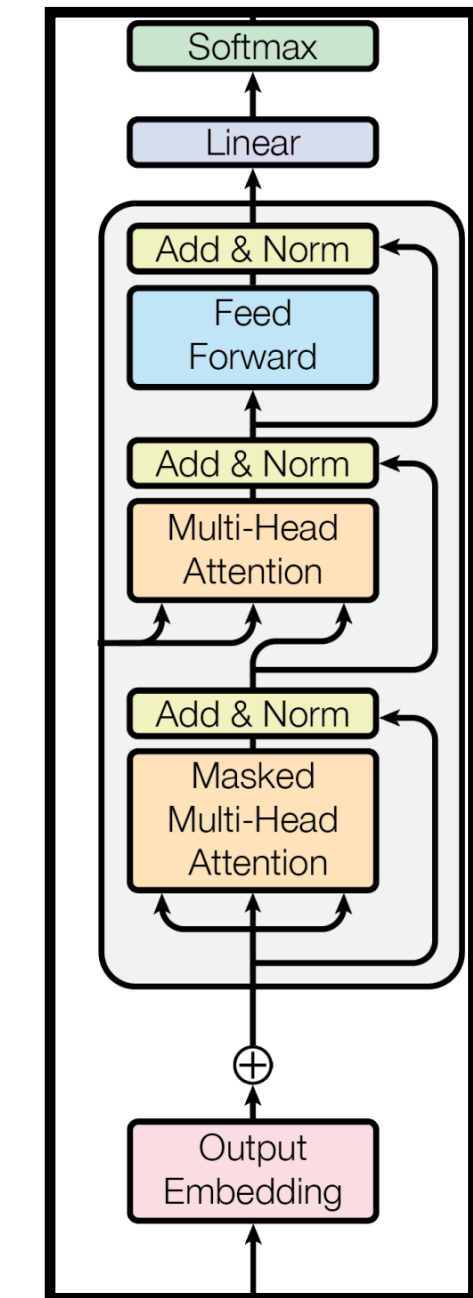
Given a trained LM and its training corpus, we construct a **weighted finite-state automaton**.

Automaton



- States: clusters of training examples, encoded by the LM
- Edges: pointers between consecutive examples, shared in cluster
- Weights: $-\|h^{(t)}, h_i\|_2$

Trained LM

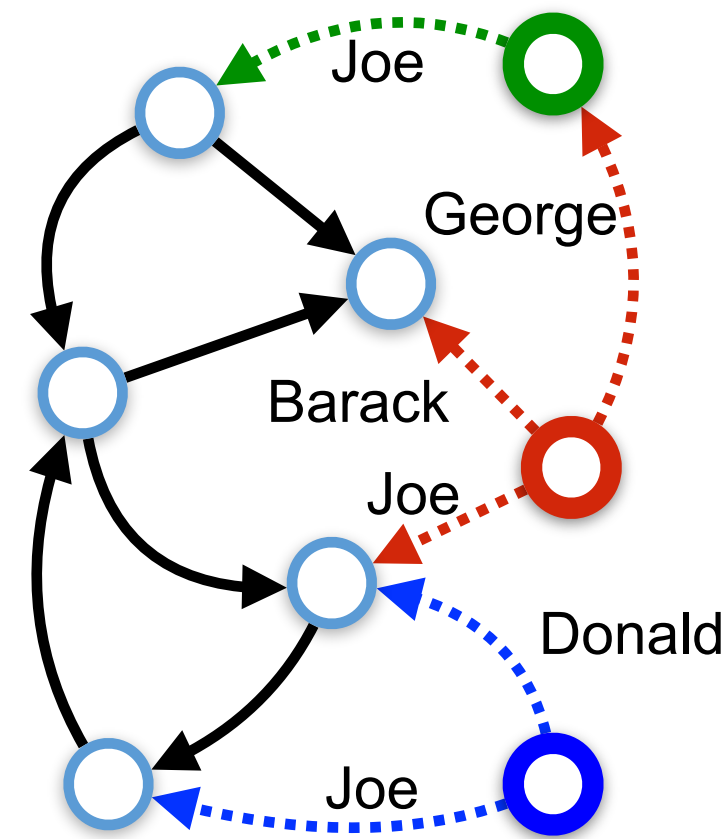


RetoMaton - TL;DR:

Given a trained LM and its training corpus, we construct a **weighted finite-state automaton**.

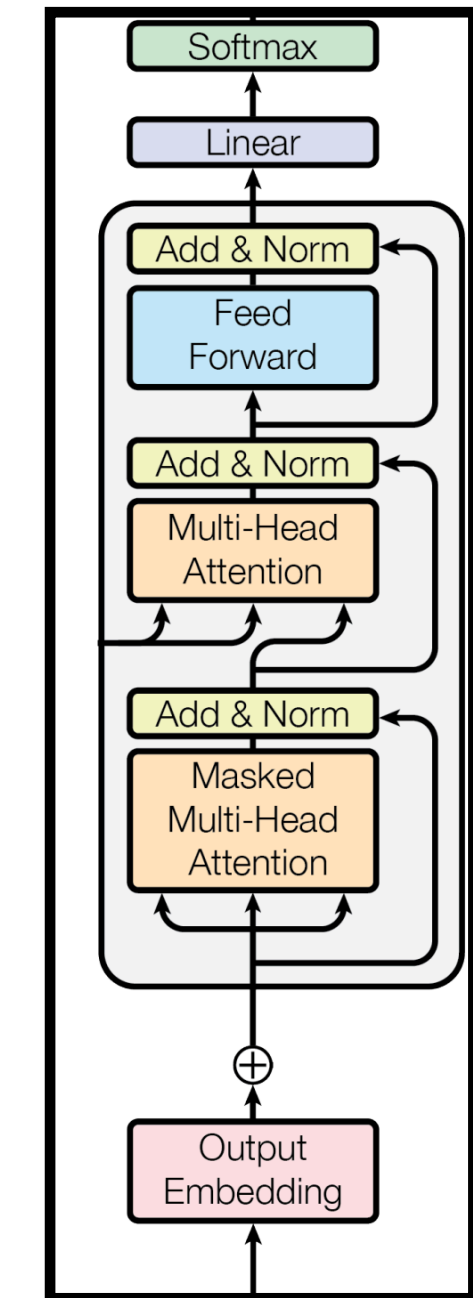
At **inference** time, we traverse the automaton in parallel with the LM.

Automaton



- States: clusters of training examples, encoded by the LM
- Edges: pointers between consecutive examples, shared in cluster
- Weights: $-\|h^{(t)}, h_i\|_2$

Trained LM



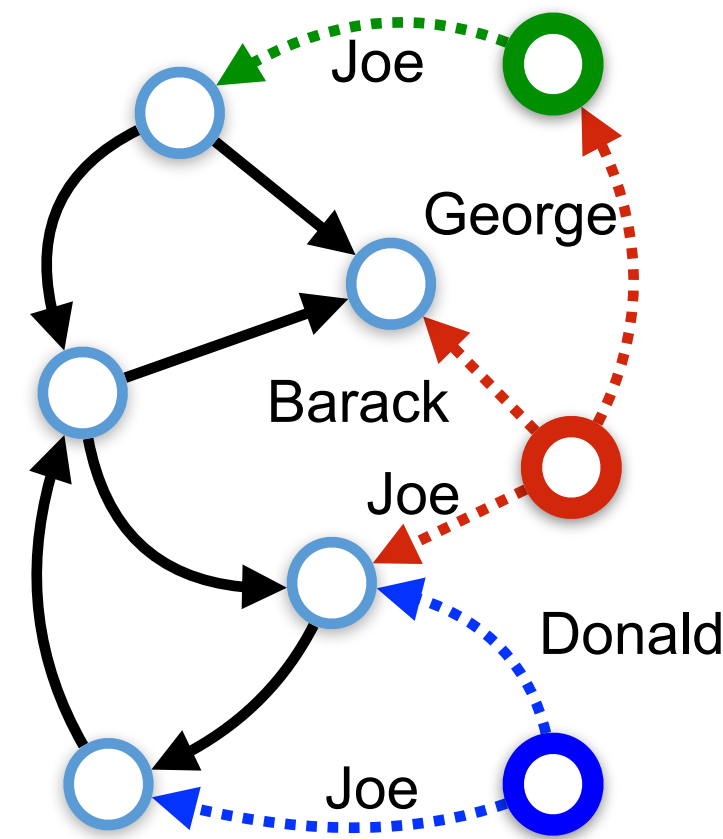
RetoMaton - TL;DR:

Given a trained LM and its training corpus, we construct a **weighted finite-state automaton**.

At **inference** time, we traverse the automaton in parallel with the LM.

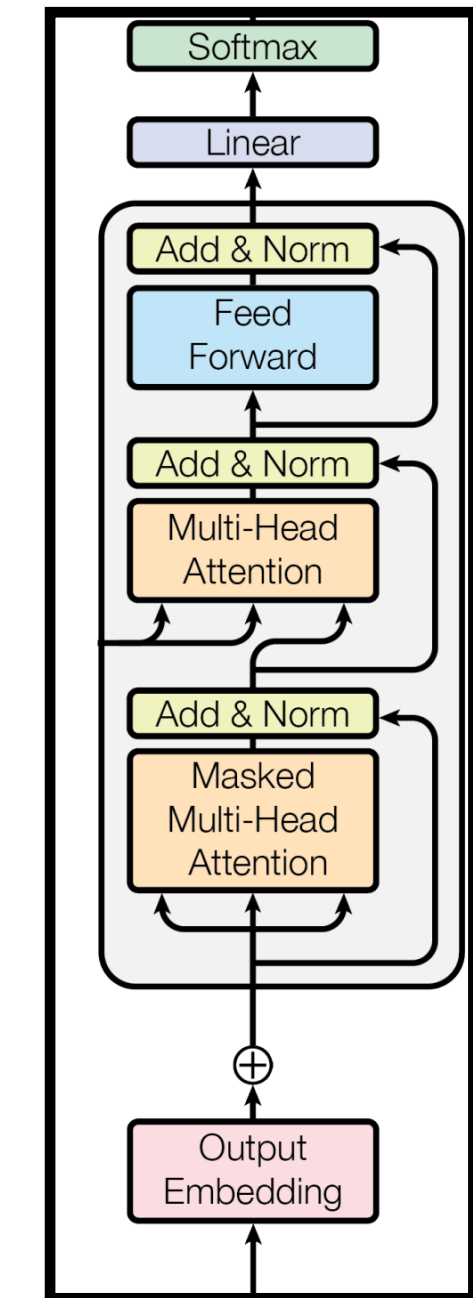
We **interpolate** this automaton's probability with the base LM's probability.

Automaton



- States: clusters of training examples, encoded by the LM
- Edges: pointers between consecutive examples, shared in cluster
- Weights: $-\|h^{(t)}, h_i\|_2$

Trained LM



$$\lambda P_{auto} + (1 - \lambda) P_{LM}$$

Background: K-Nearest Neighbor Language Model (k NN-LM)

(Khandelwal et al., ICLR'2020)

Background: K-Nearest Neighbor Language Model (k NN-LM) (Khandelwal et al., ICLR'2020)

Test

The	president	is	—
-----	-----------	----	---

Background: K-Nearest Neighbor Language Model (k NN-LM) (Khandelwal et al., ICLR'2020)

Training

k NN search



Test

The	president	is	—
-----	-----------	----	---

Background: K-Nearest Neighbor Language Model (k NN-LM)

(Khandelwal et al., ICLR'2020)

Training

k NN search



Test

... by the president Joe Biden ...

The	president	is	—
-----	-----------	----	---

Background: K-Nearest Neighbor Language Model (k NN-LM)

(Khandelwal et al., ICLR'2020)

Training

Context Next word



... by the president Joe Biden ...

k NN search

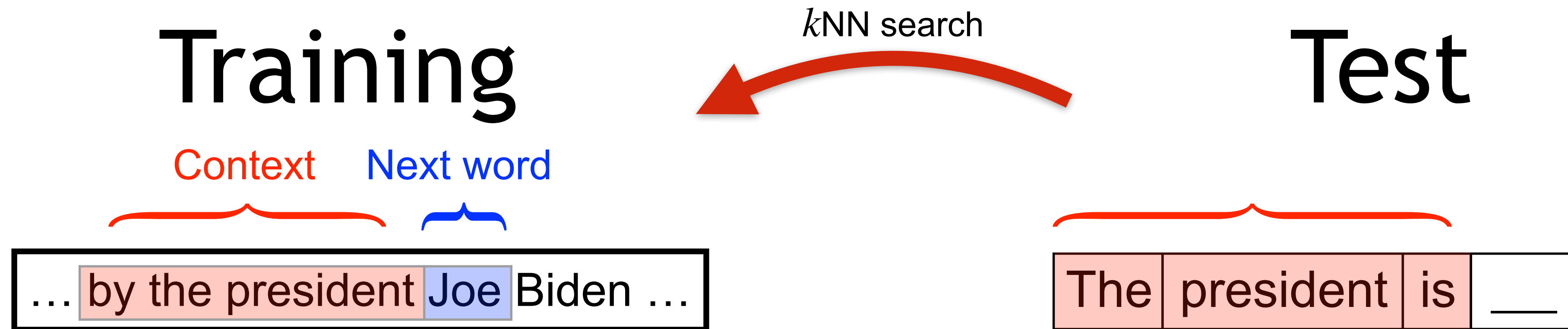


Test

The president is _____

Background: K-Nearest Neighbor Language Model (k NN-LM)

(Khandelwal et al., ICLR'2020)



Background: K-Nearest Neighbor Language Model (k NN-LM)

(Khandelwal et al., ICLR'2020)

Training

Context Next word

... by the president Joe Biden ...

k NN search

Test

The president is _____
Joe

Background: K-Nearest Neighbor Language Model (k NN-LM)

(Khandelwal et al., ICLR'2020)

Training

Context Next word

... by the president Joe Biden ...

k NN search

Test

time t

The president is Joe

Joe

Background: K-Nearest Neighbor Language Model (k NN-LM)

(Khandelwal et al., ICLR'2020)

Training

Context Next word

... by the president Joe Biden ...

k NN search



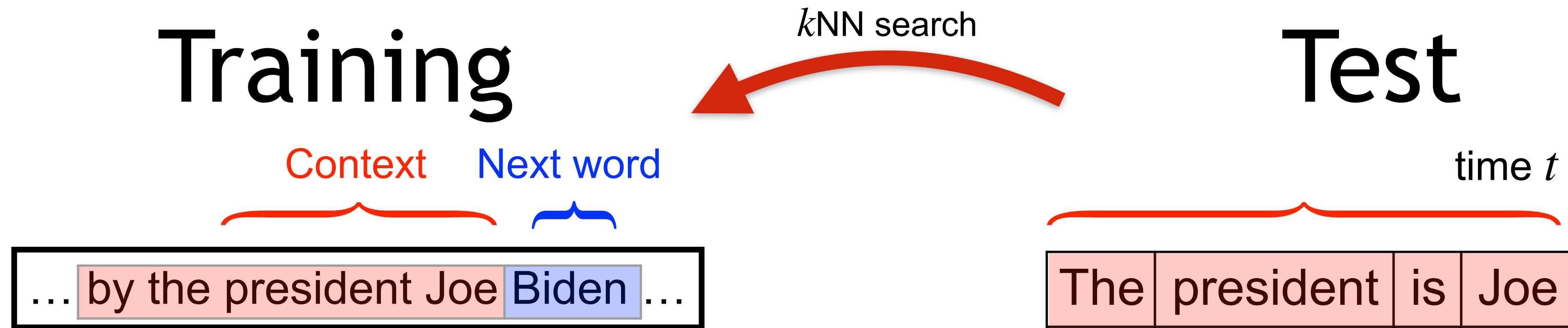
Test

time t

The	president	is	Joe
-----	-----------	----	-----

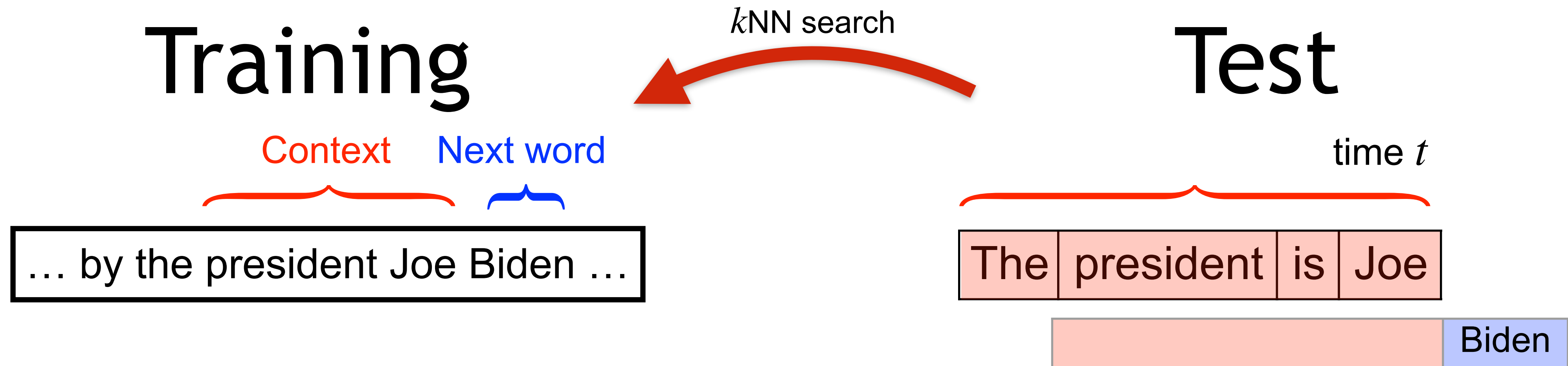
Background: K-Nearest Neighbor Language Model (k NN-LM)

(Khandelwal et al., ICLR'2020)



Background: K-Nearest Neighbor Language Model (k NN-LM)

(Khandelwal et al., ICLR'2020)



Background: K-Nearest Neighbor Language Model (k NN-LM)

(Khandelwal et al., ICLR'2020)

Training

Context Next word

... by the president Joe Biden ...

k NN search

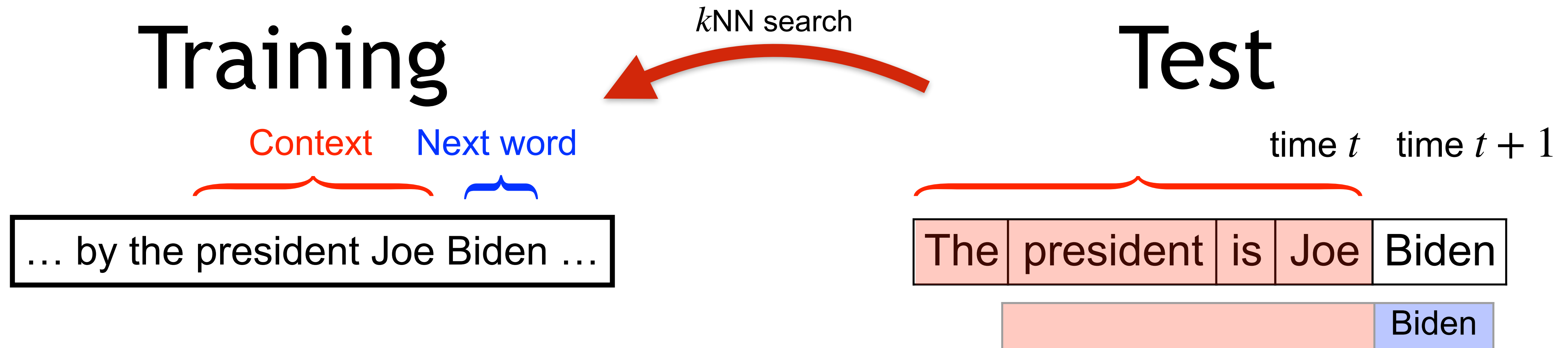
Test

time t time $t + 1$

The president is Joe Biden
Biden

Background: K-Nearest Neighbor Language Model (k NN-LM)

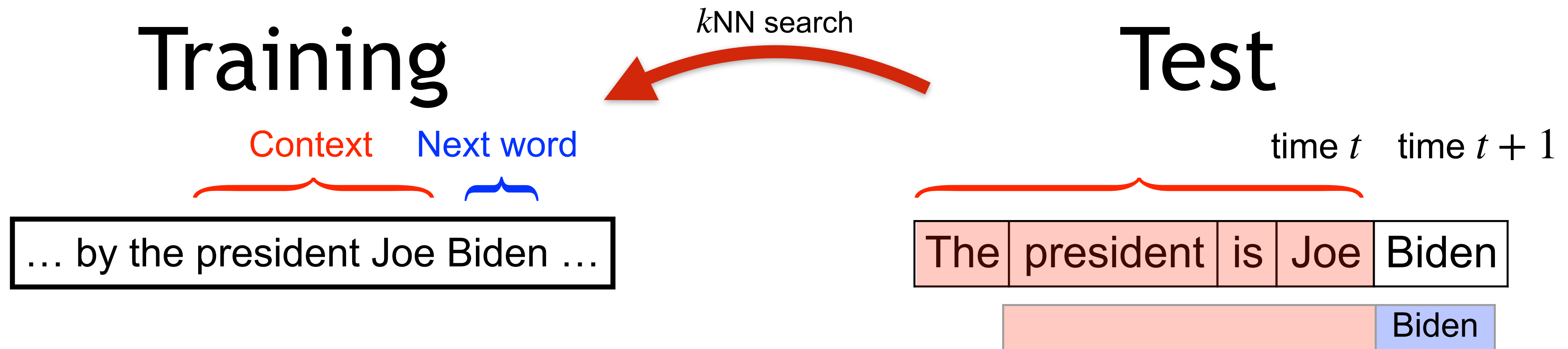
(Khandelwal et al., ICLR'2020)



K-nearest neighbor search: for **every generated token**
time (k NN search) \gg time (forward pass)

Background: K-Nearest Neighbor Language Model (k NN-LM)

(Khandelwal et al., ICLR'2020)



K-nearest neighbor search: for **every generated token**
time (k NN search) \gg time (forward pass)

If we performed **k NN search** to retrieve “Joe”,
can we save the search when predicting “Biden”?

Adding Pointers Between Datastore Entries

Training

... by the president Joe Biden ...

Adding Pointers Between Datastore Entries

Training

... by the president Joe Biden ...

encode(by the president) Joe

Adding Pointers Between Datastore Entries

Training

... by the president Joe Biden ...

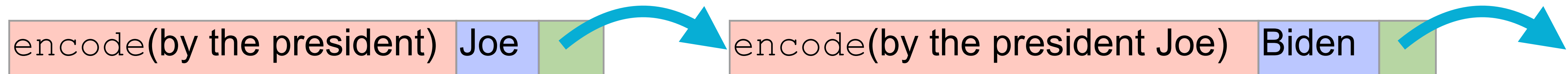
encode(by the president) Joe

encode(by the president Joe) Biden

Adding Pointers Between Datastore Entries

Training

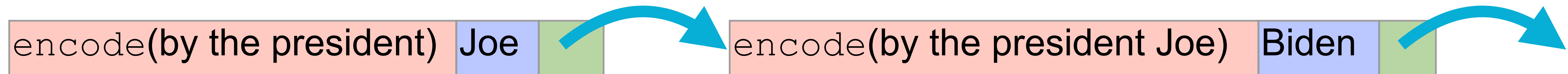
... by the president Joe Biden ...



Adding Pointers Between Datastore Entries

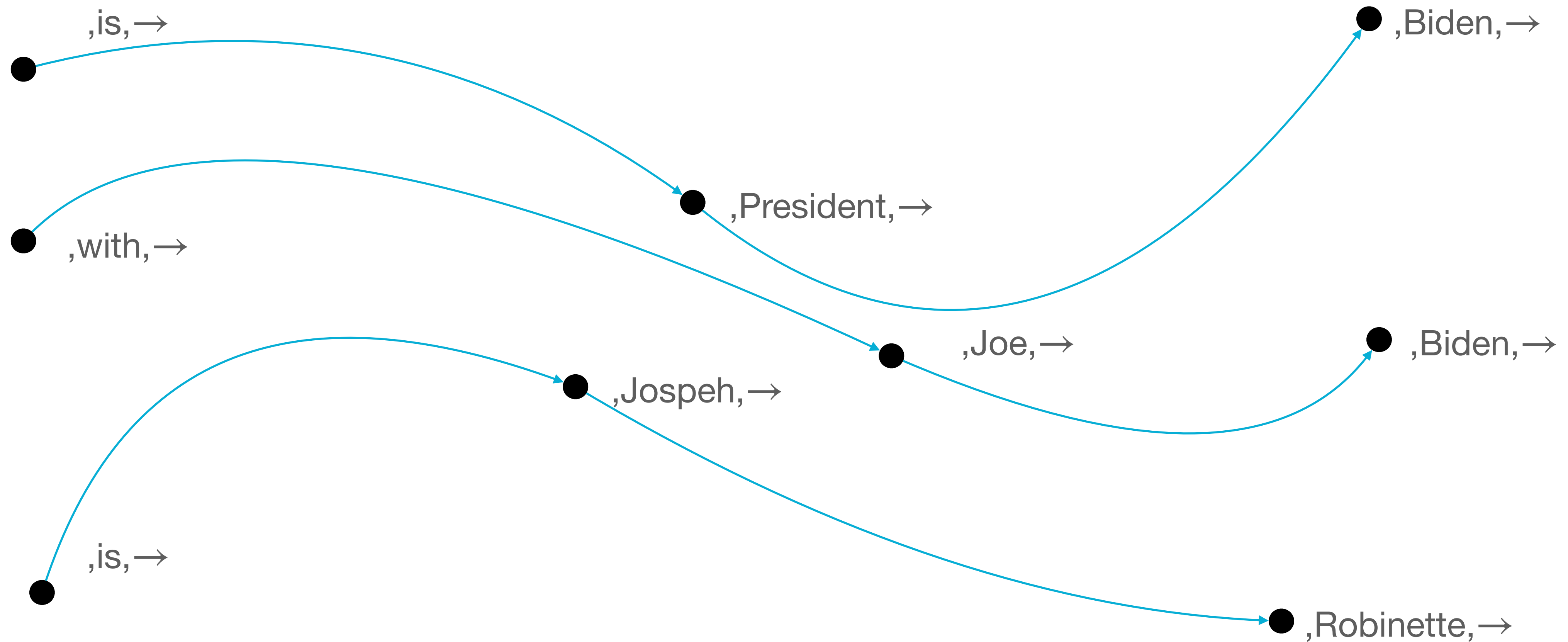
Training

... by the president Joe Biden ...

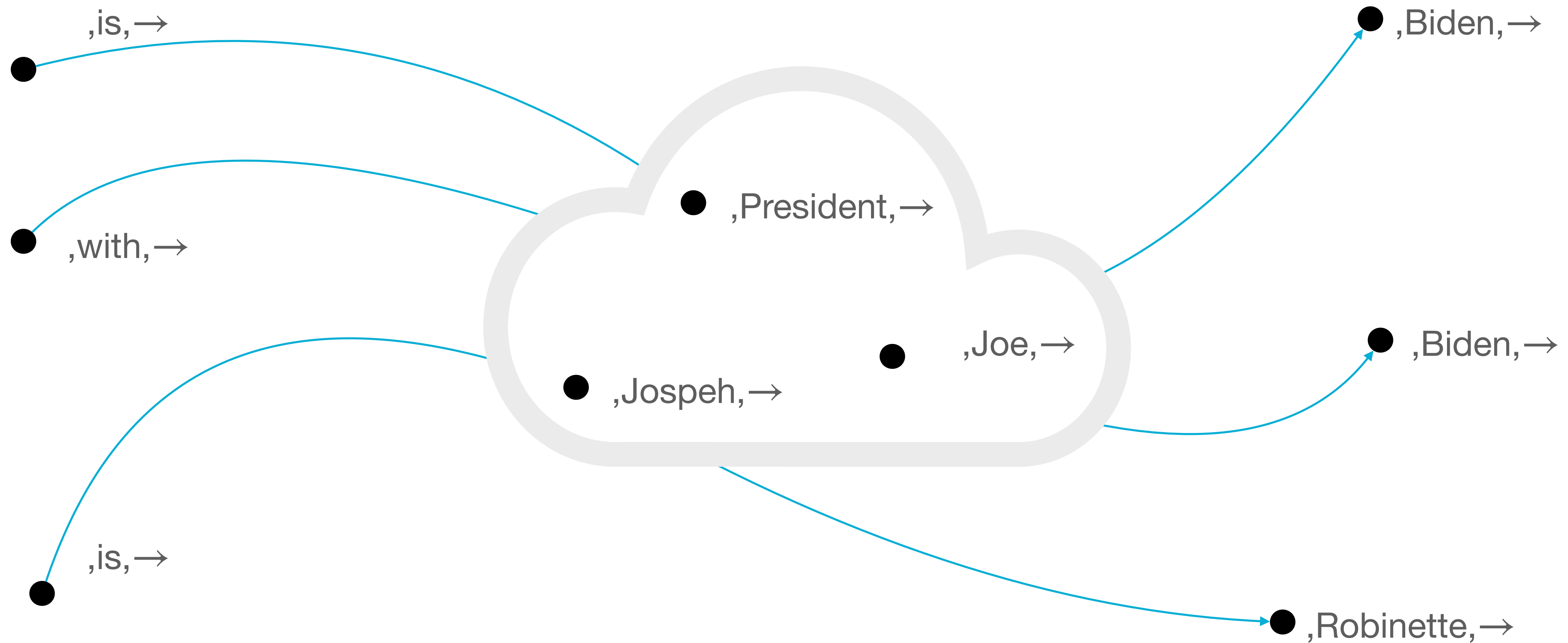


We still need to perform **k NN search** once, but in the following time steps, we can just follow pointers instead!

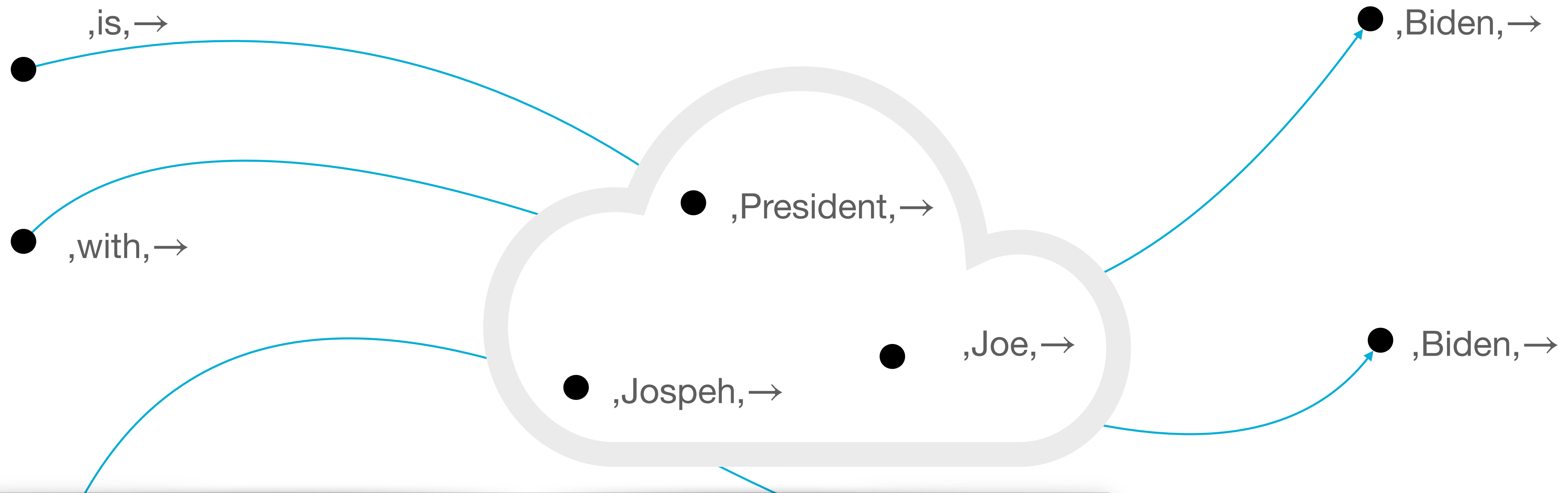
Clustering Entries with Close Keys



Clustering Entries with Close Keys



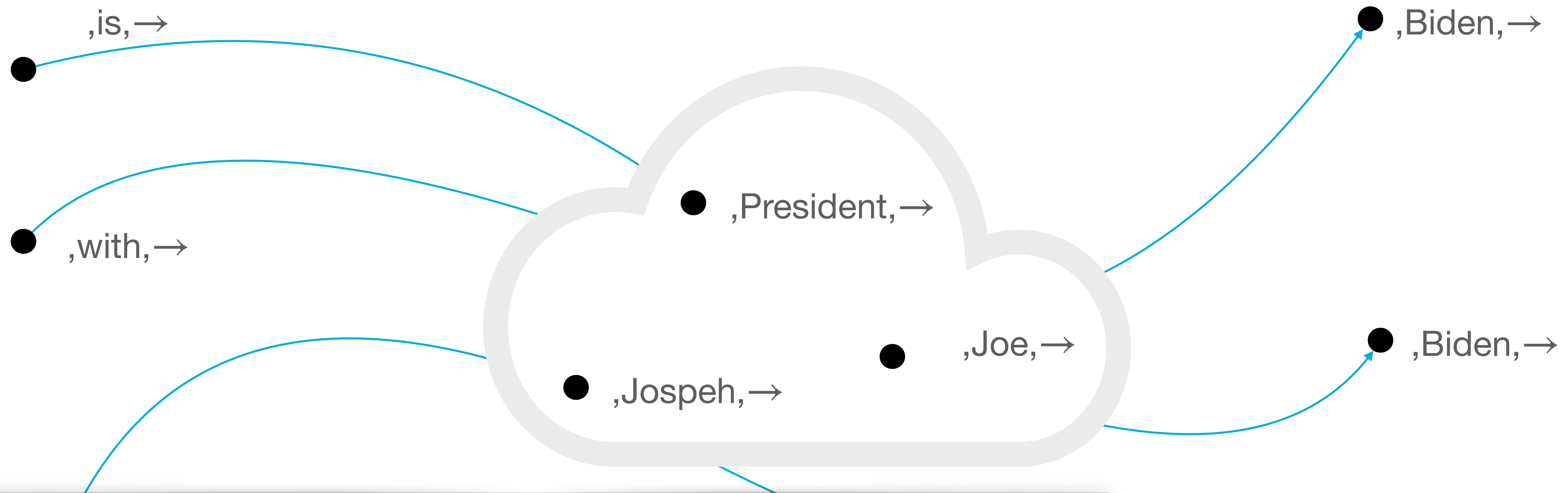
Clustering Entries with Close Keys



Cluster such entries, and share their outgoing pointers

,Robinette, →

Clustering Entries with Close Keys

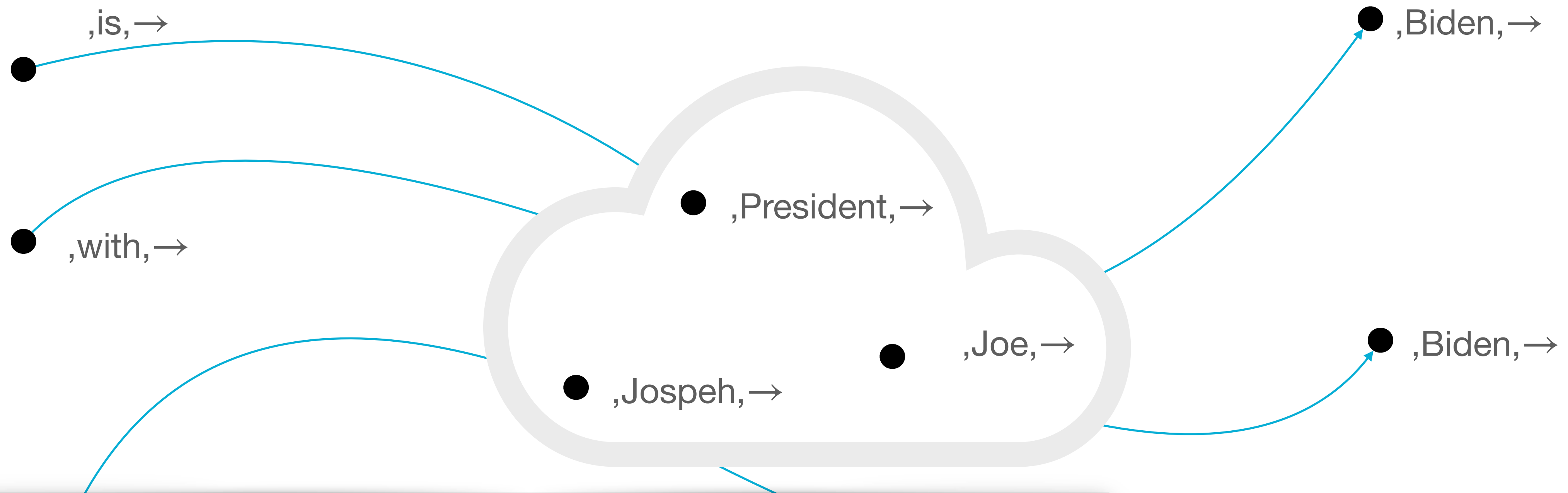


Cluster such entries, and share their outgoing pointers

👍 Capture n-grams that were unseen at training time



Clustering Entries with Close Keys



Cluster such entries, and share their outgoing pointers

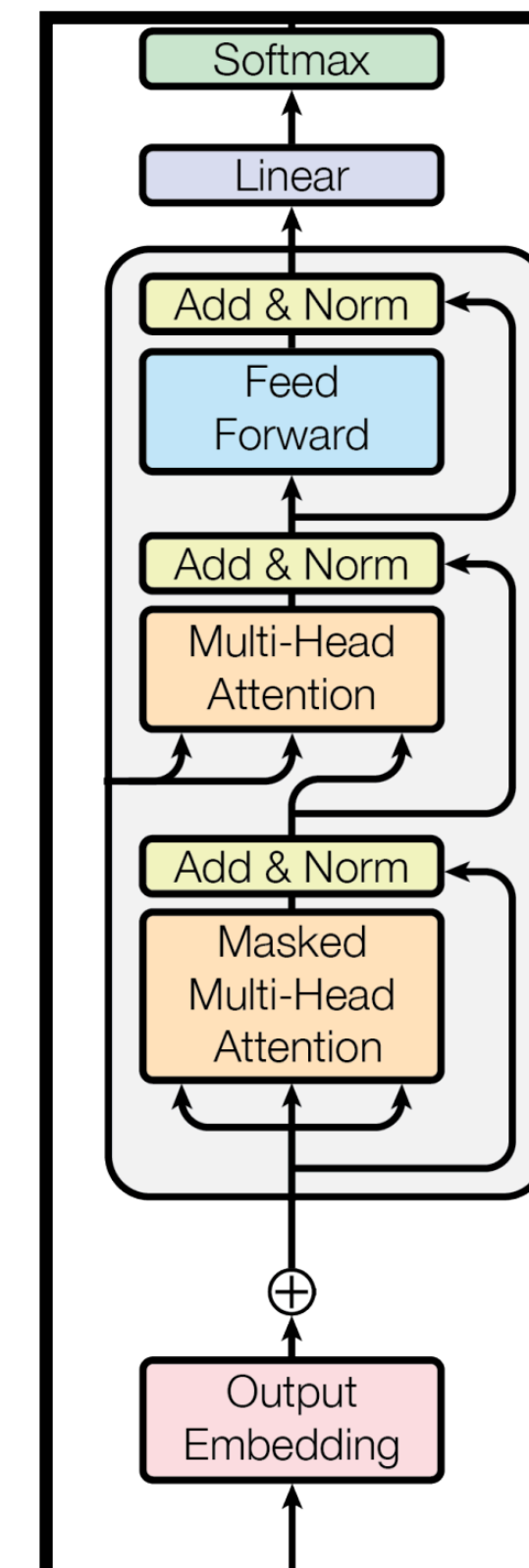
👍 Capture n-grams that were unseen at training time

👍 Longer pointer traversal, without backing up to k NN search

→ ',Robinette,→

RetoMaton

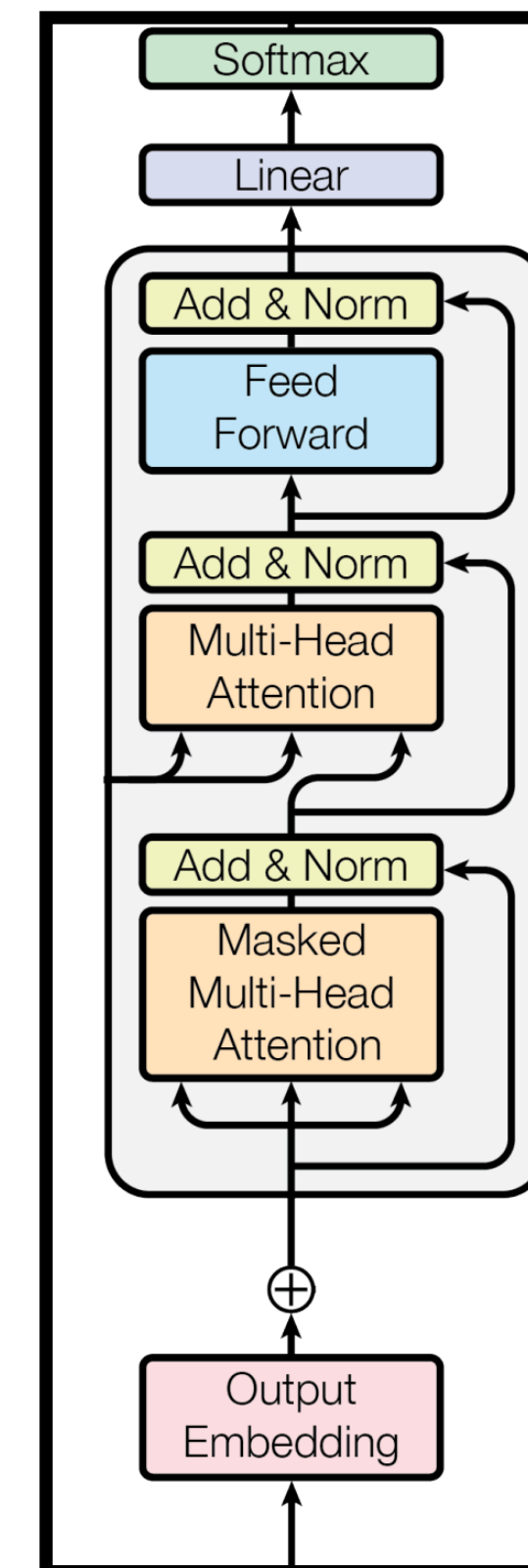
Trained LM



RetoMaton

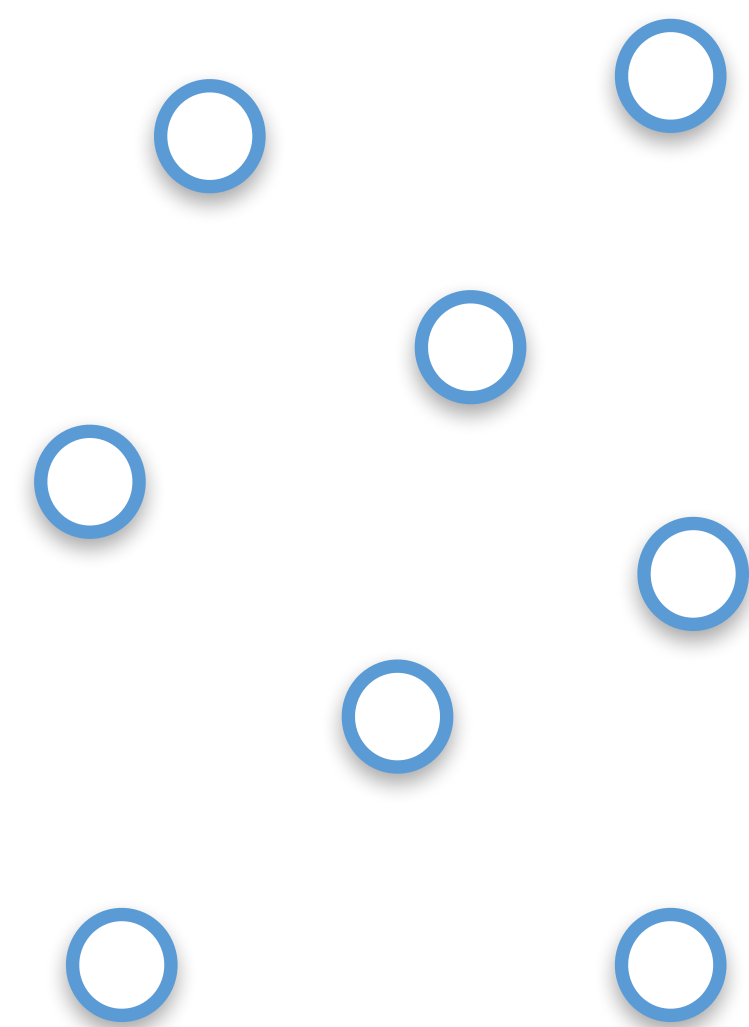
Automaton

Trained LM



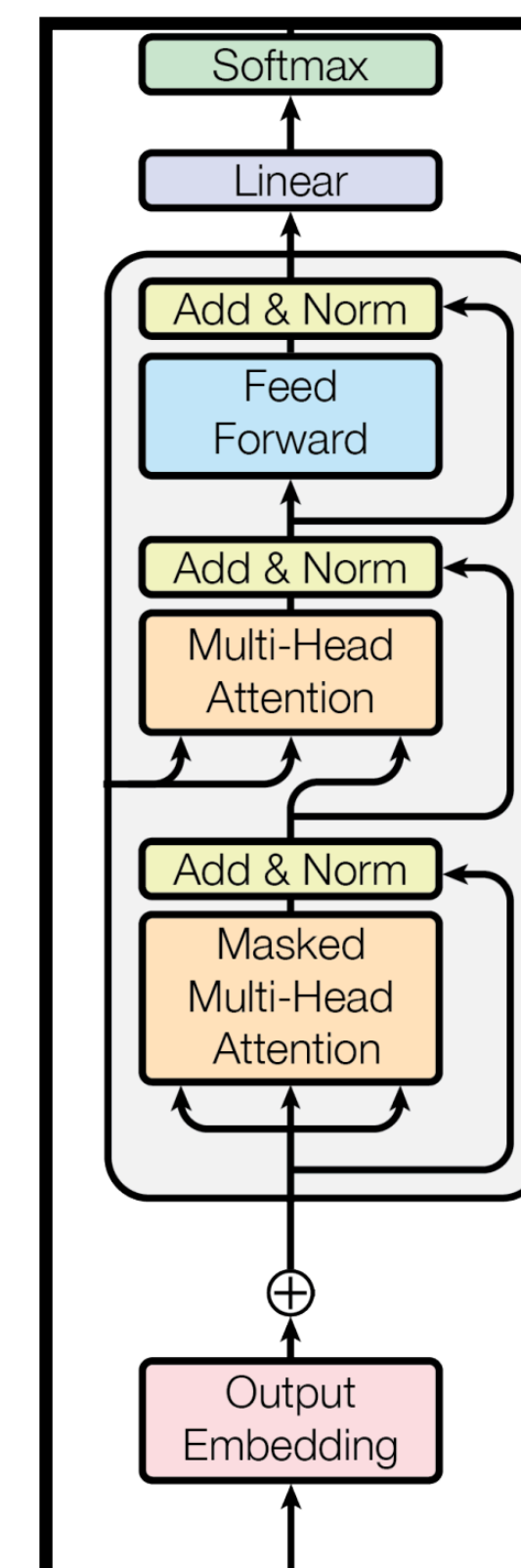
RetoMaton

Automaton



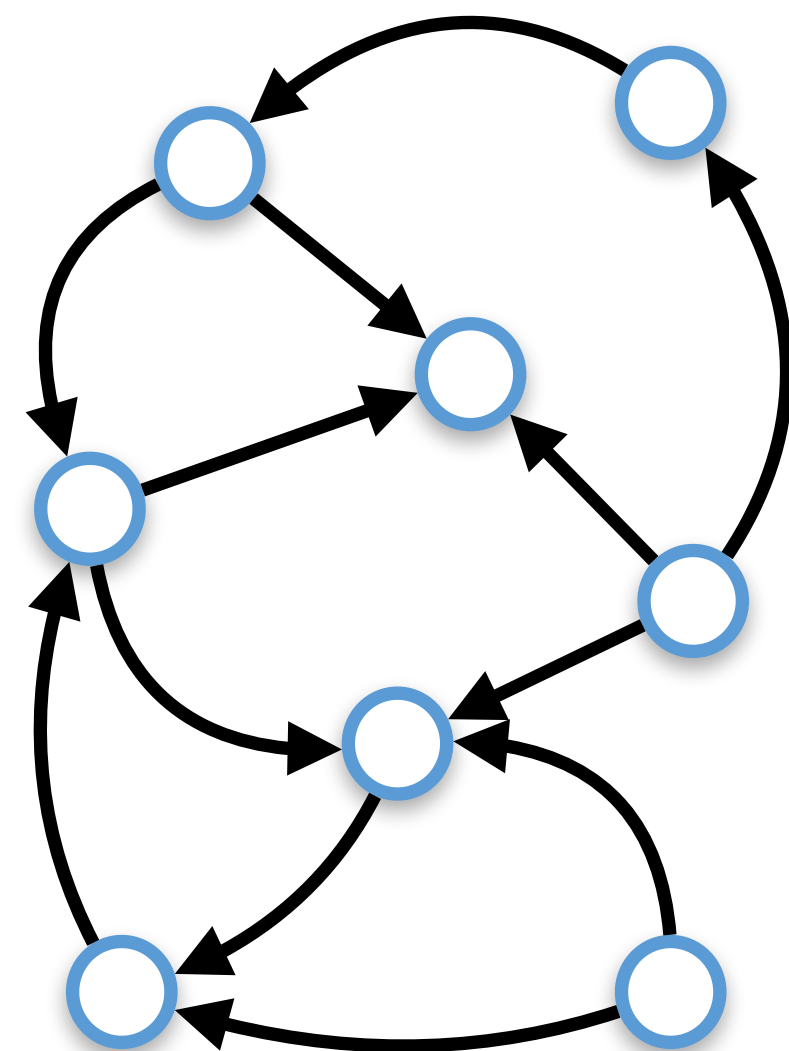
States: clusters of training examples, encoded by the LM

Trained LM



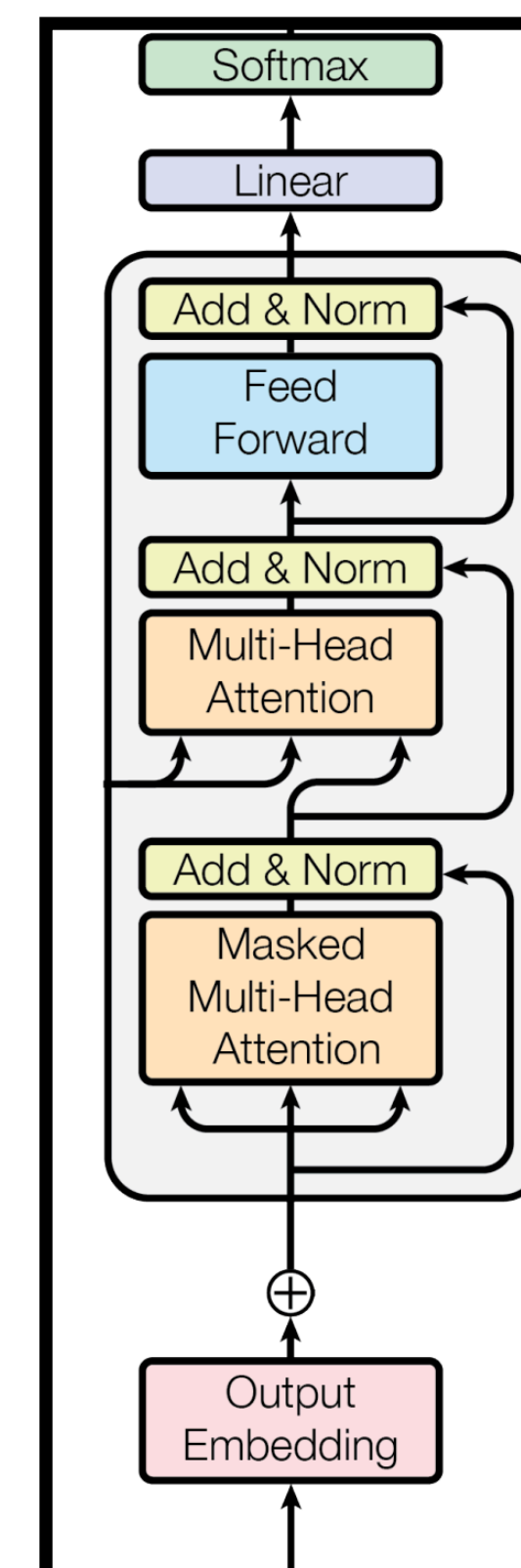
RetoMaton

Automaton



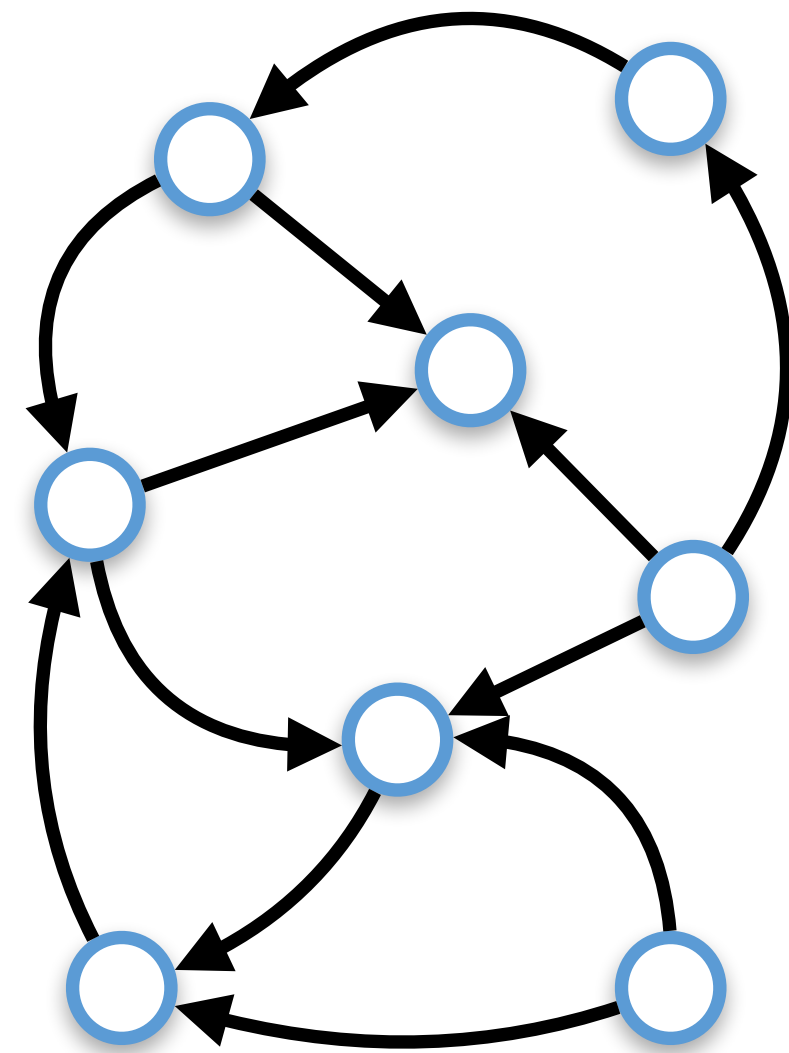
- States: clusters of training examples, encoded by the LM
- Edges: pointers between consecutive examples, shared in cluster

Trained LM



RetoMaton

Automaton

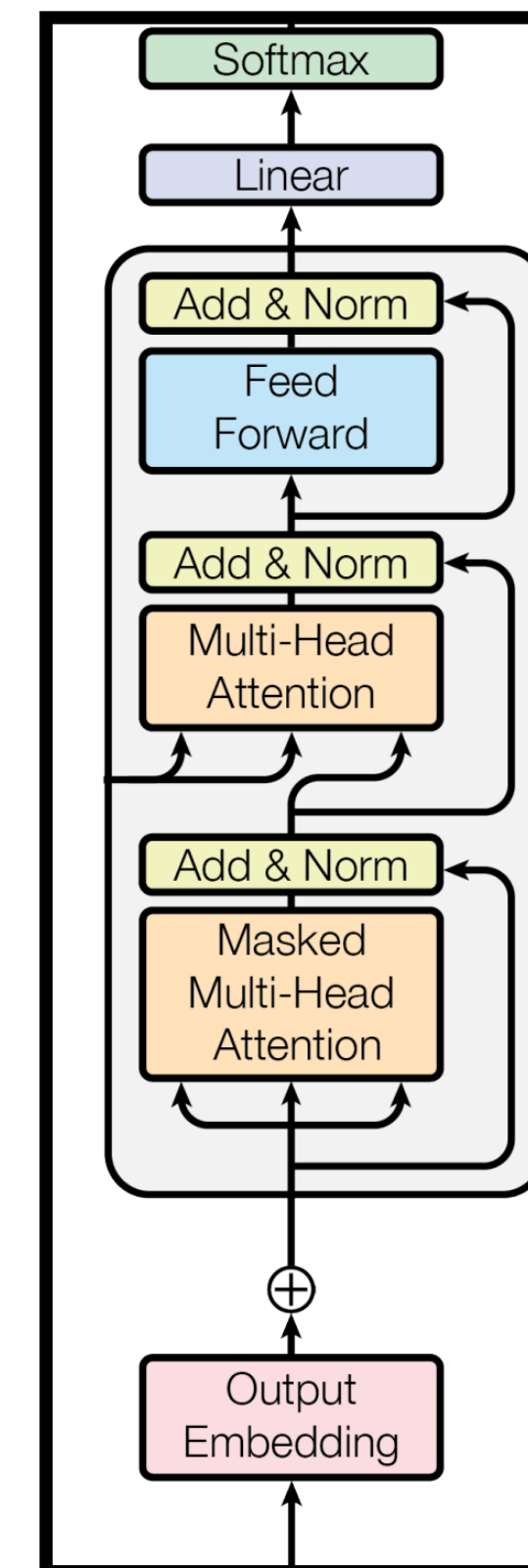


States: clusters of training examples, encoded by the LM

Edges: pointers between consecutive examples, shared in cluster

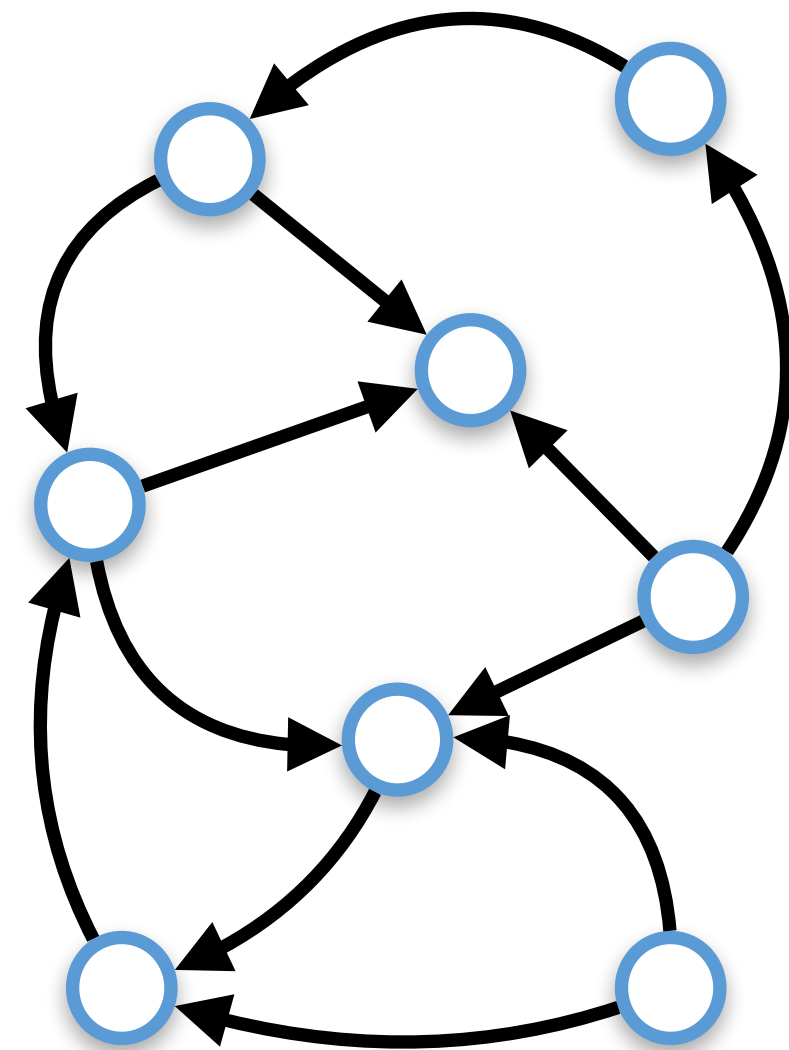
Weights: $-\|h^{(t)}, h_i\|_2$

Trained LM



RetoMaton

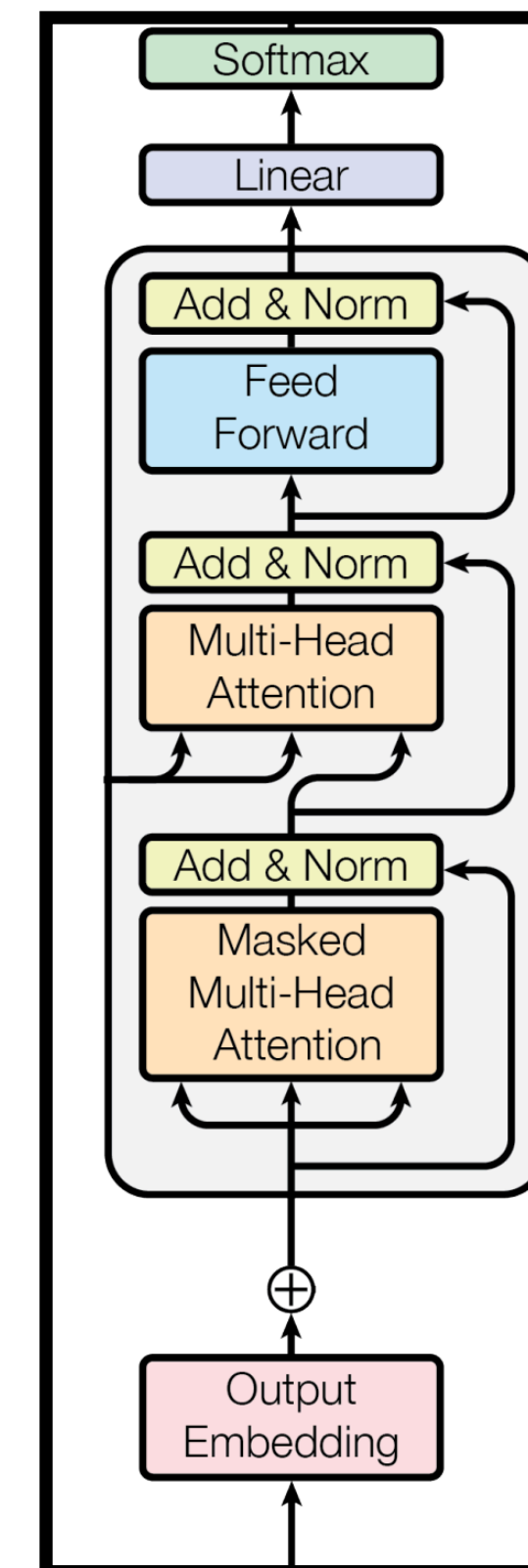
Automaton



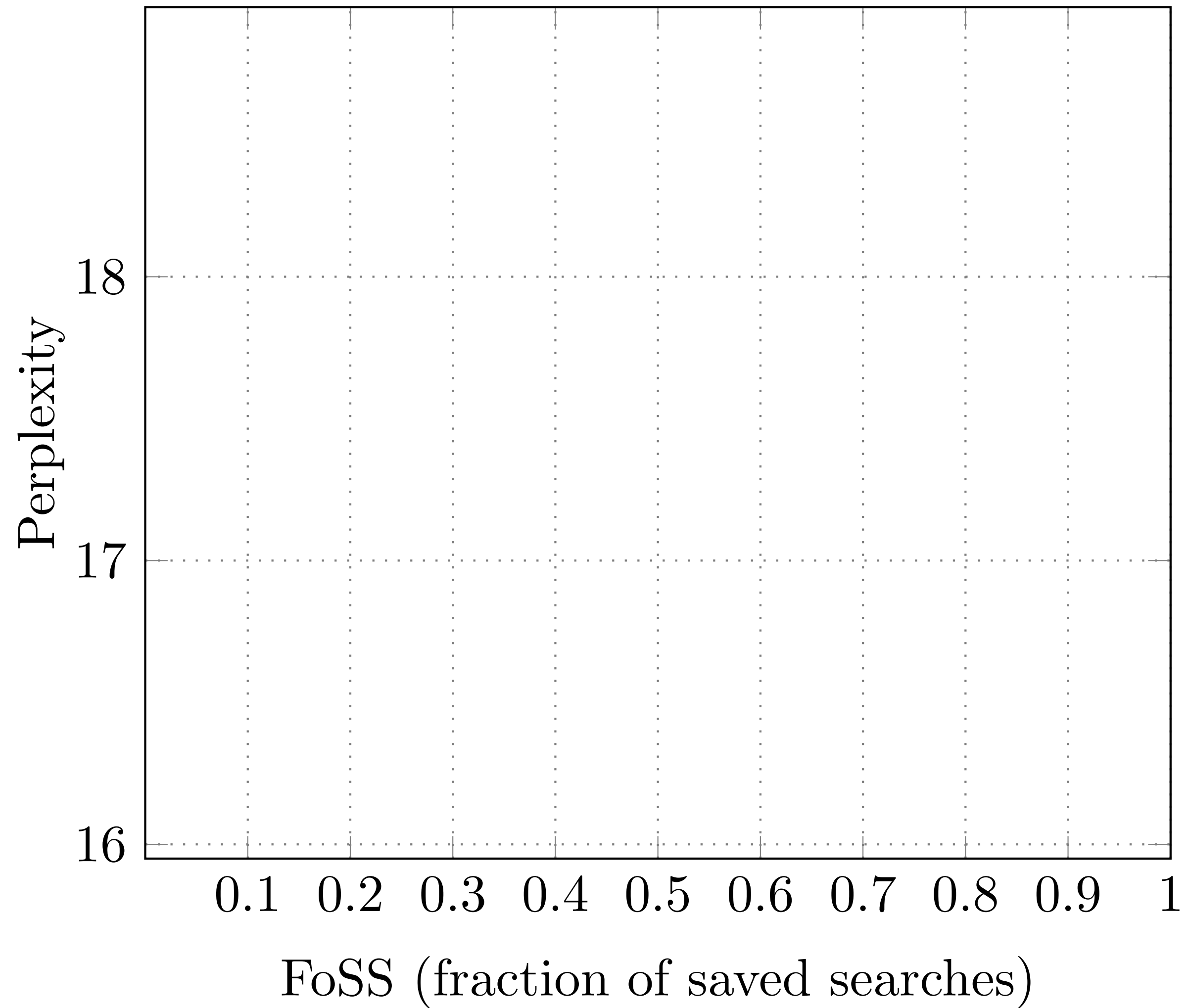
- States: clusters of training examples, encoded by the LM
- Edges: pointers between consecutive examples, shared in cluster
- Weights: $-\|h^{(t)}, h_i\|_2$

$$\lambda P_{auto} + (1 - \lambda) P_{LM}$$

Trained LM

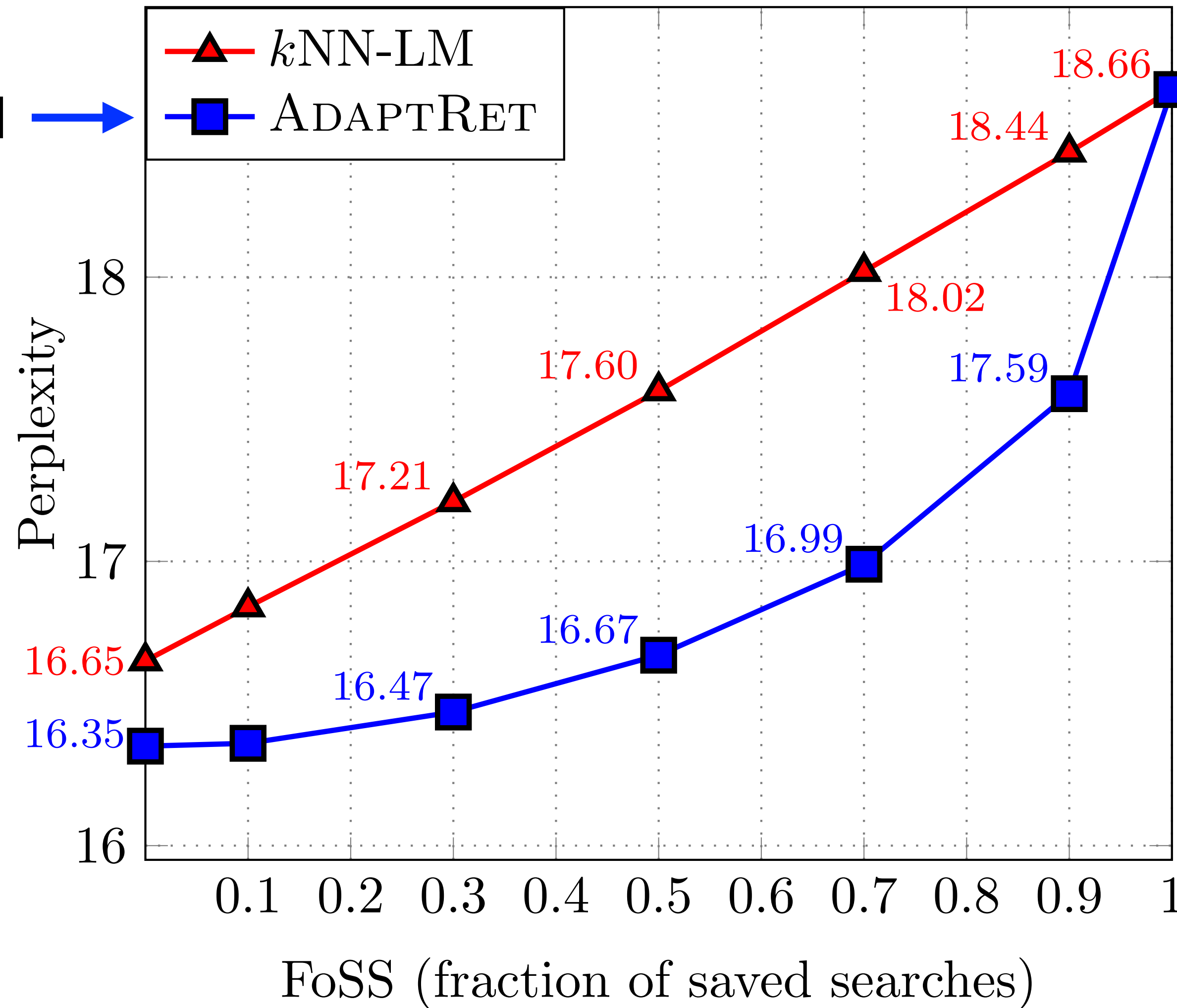


Wikitext-103



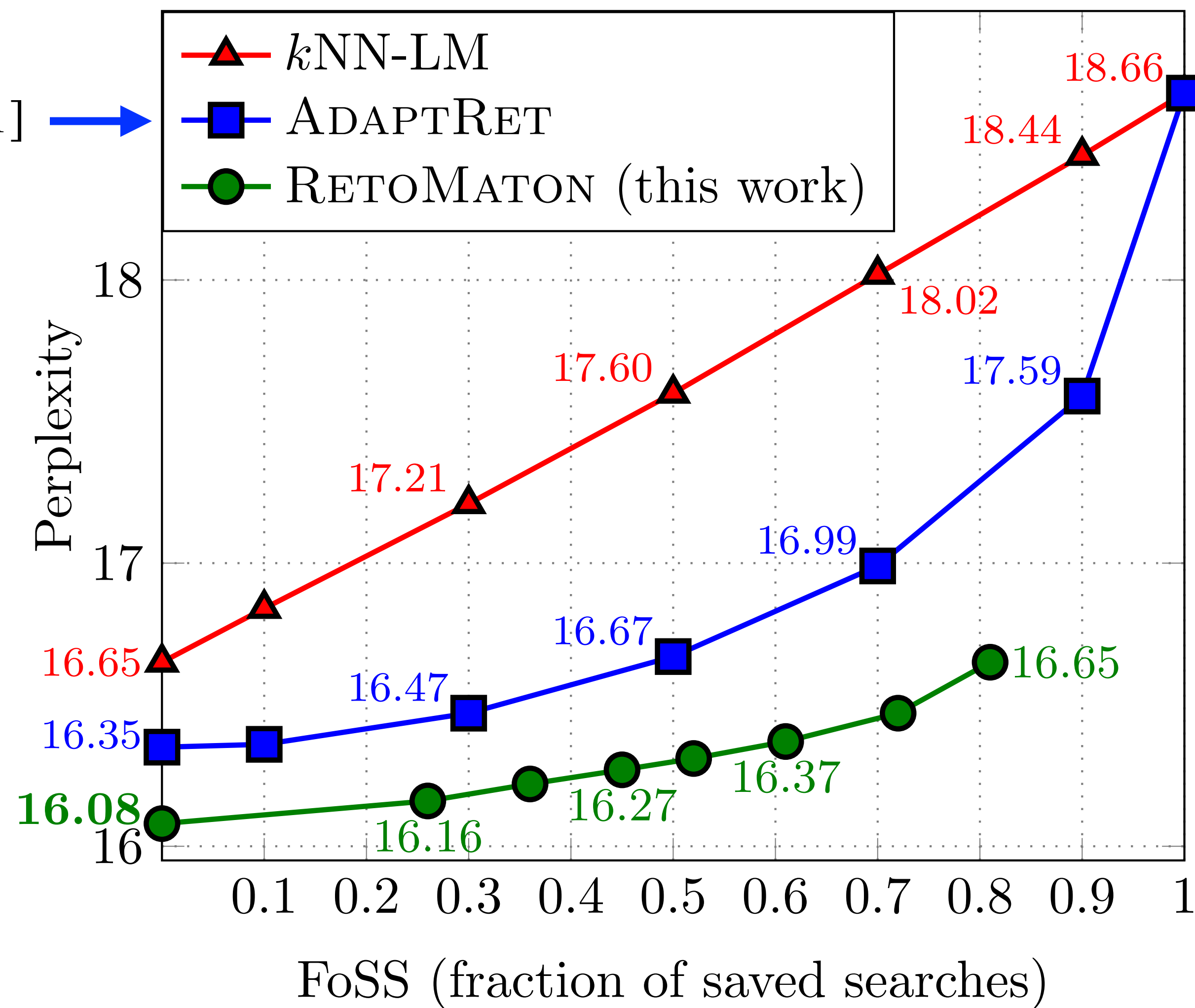
Wikitext-103

[He et al., EMNLP'2021]



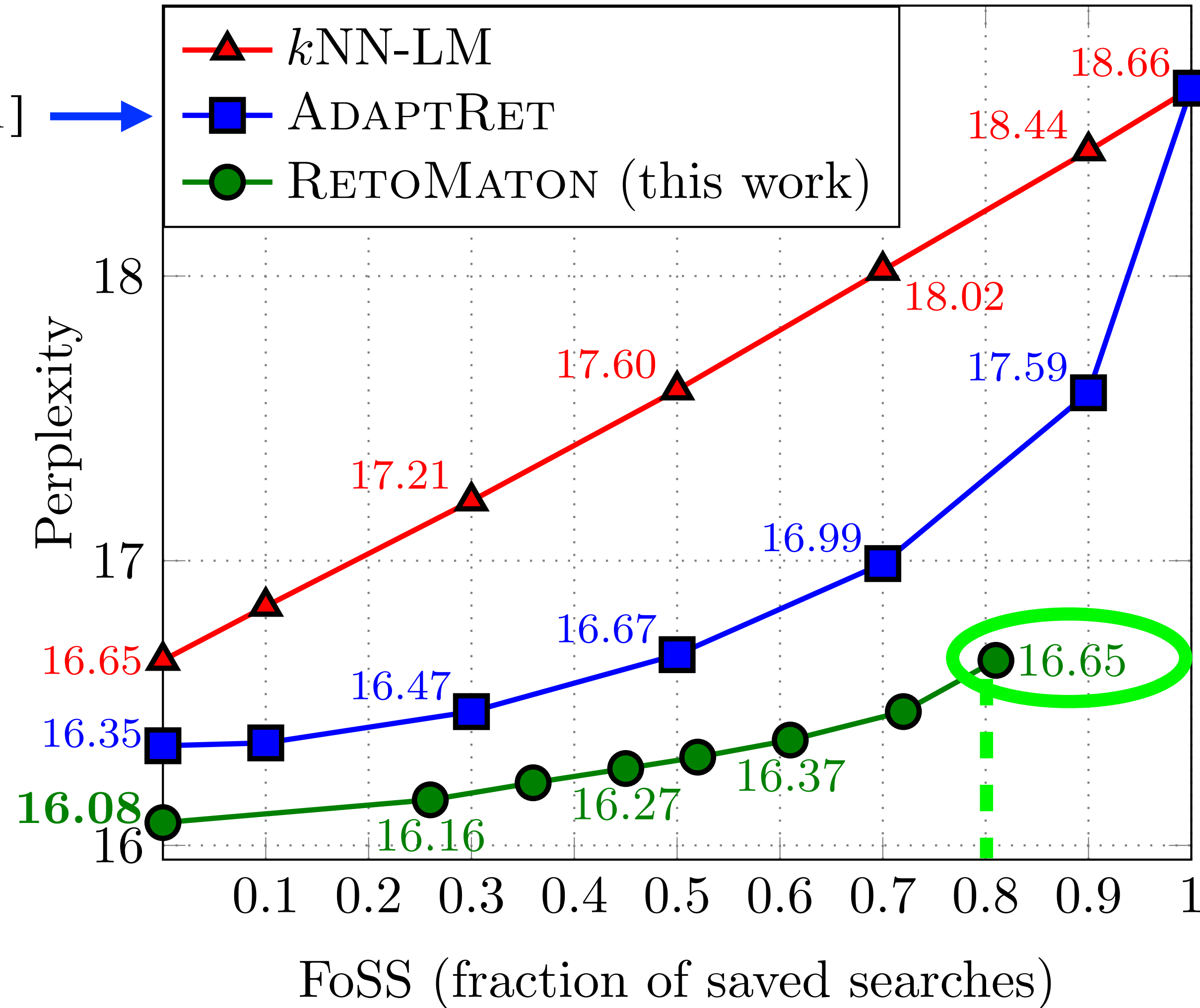
Wikitext-103

[He et al., EMNLP'2021] →



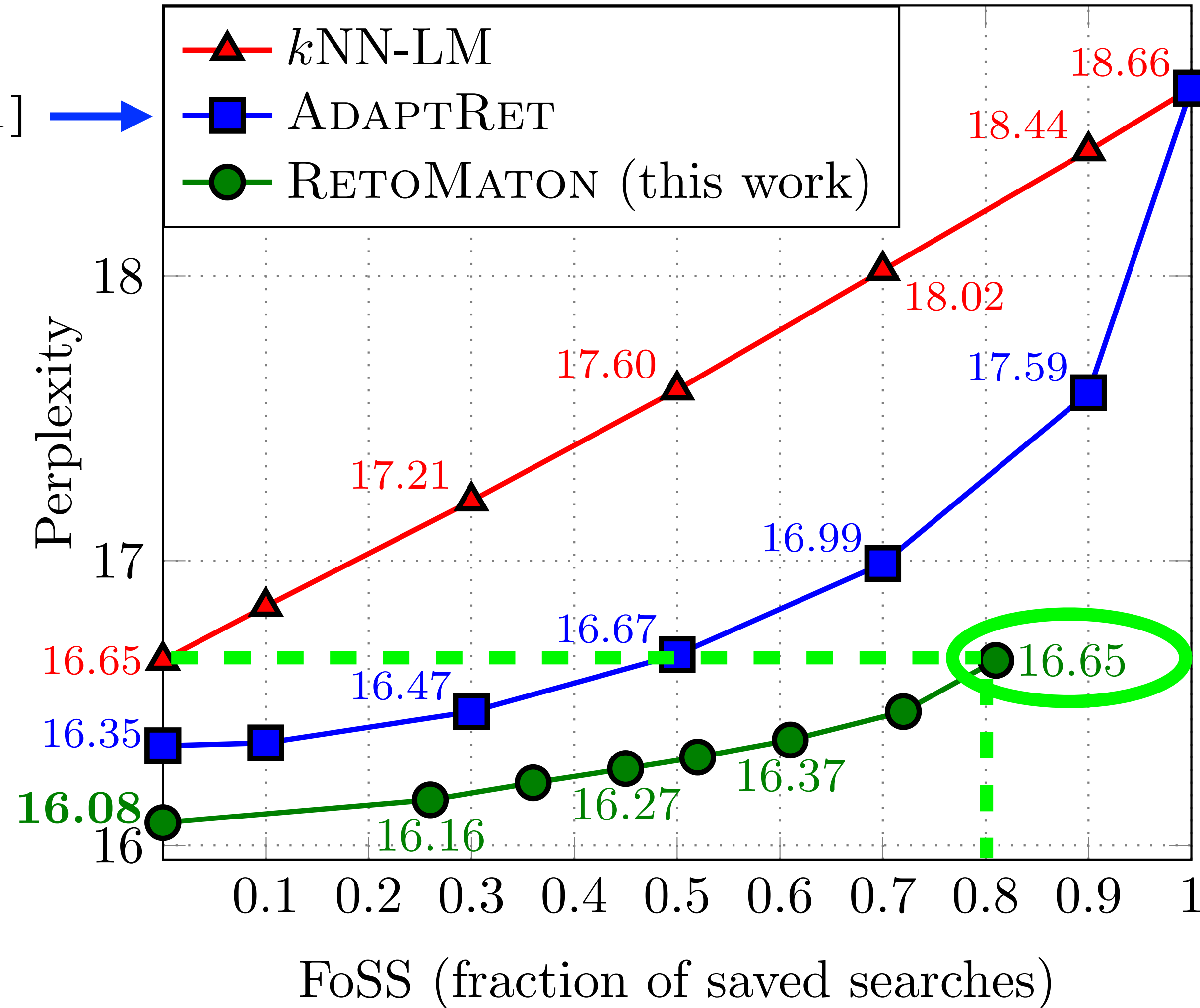
Wikitext-103

[He et al., EMNLP'2021] →



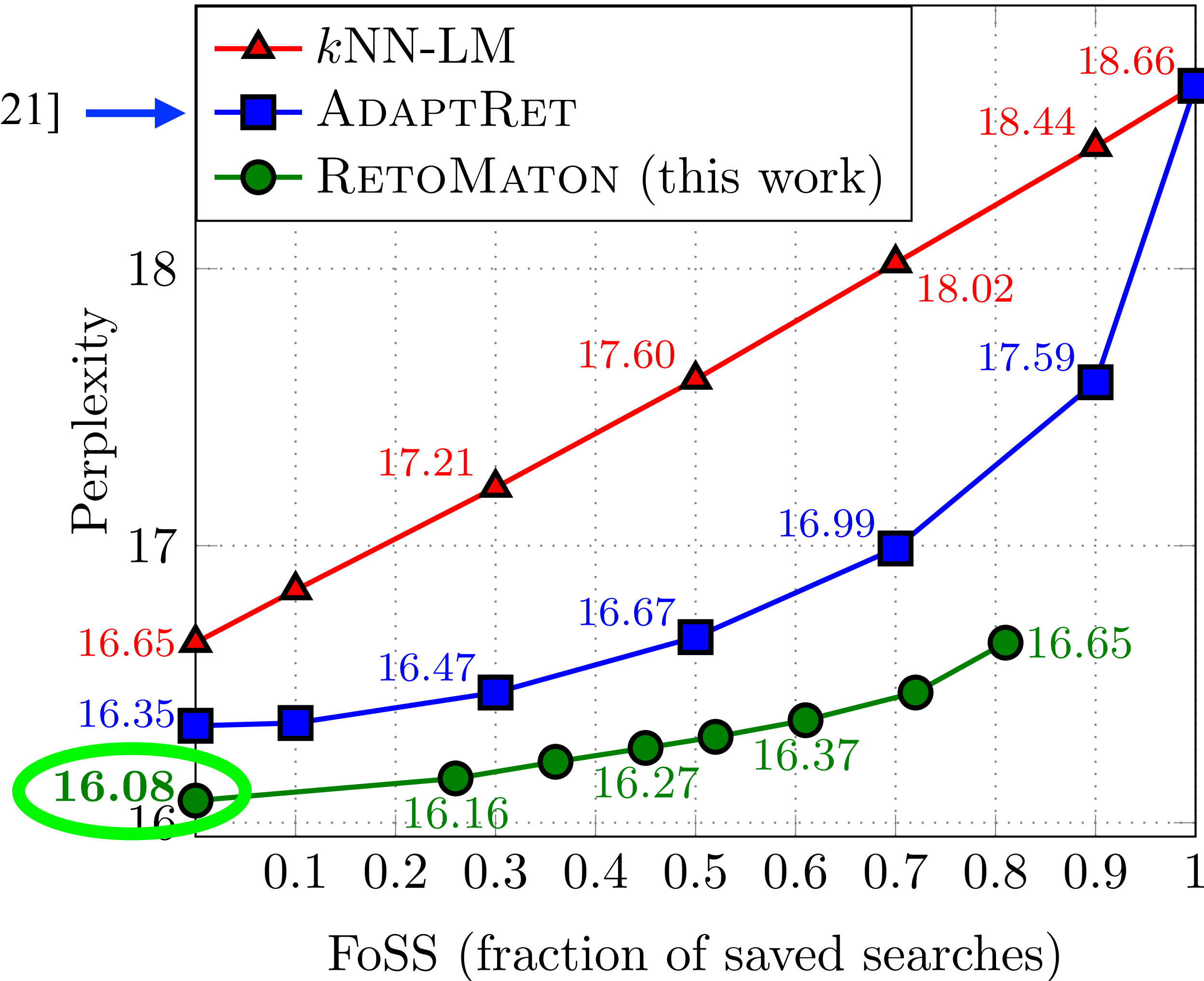
Wikitext-103

[He et al., EMNLP'2021] →



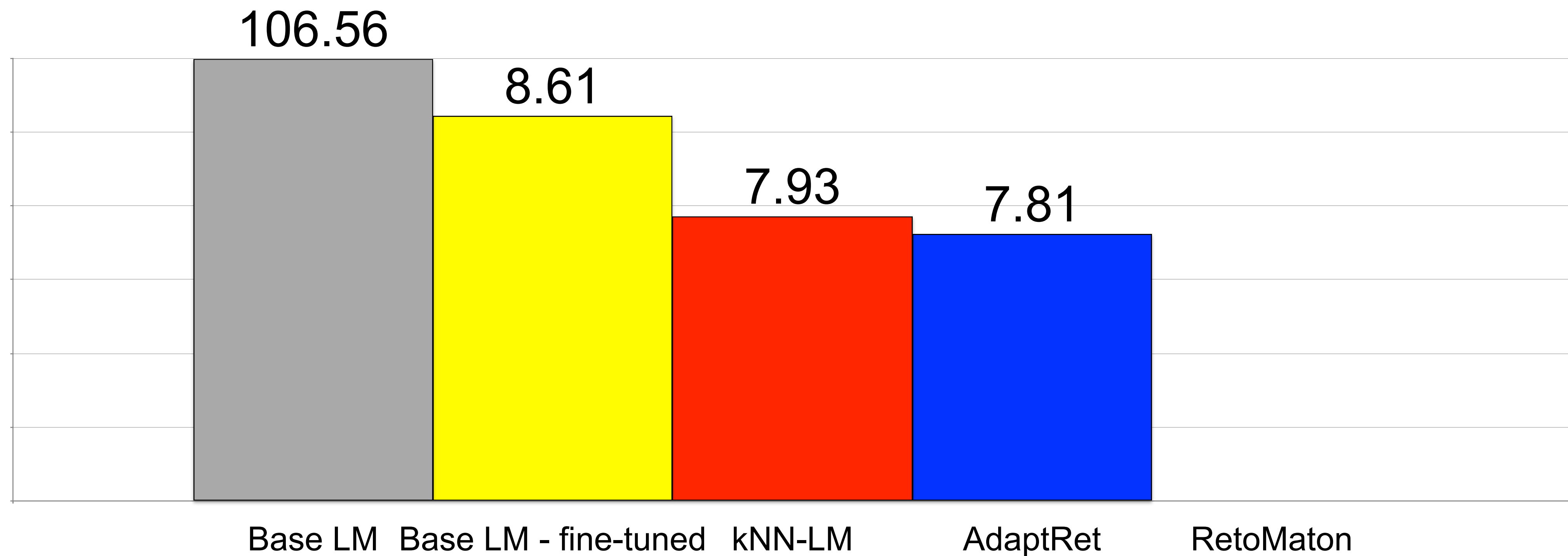
Wikitext-103

[He et al., EMNLP'2021] →



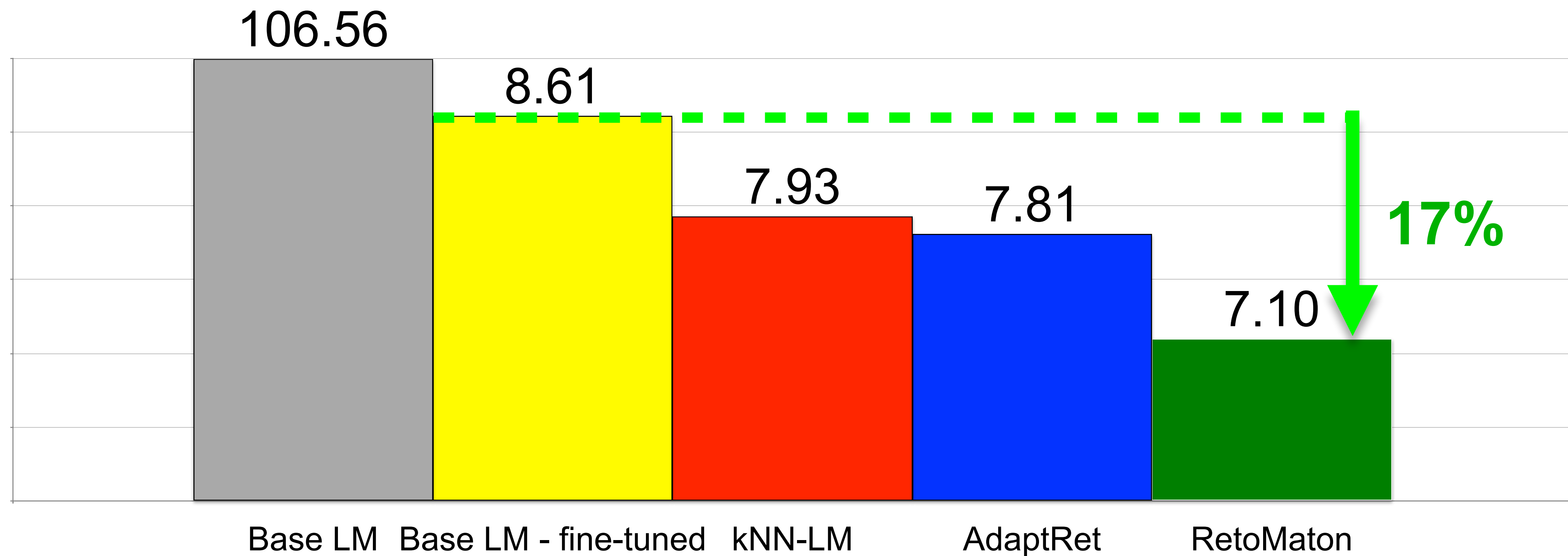
Domain Adaptation

Train on WMT News Crawl; Test+build datastore on Law

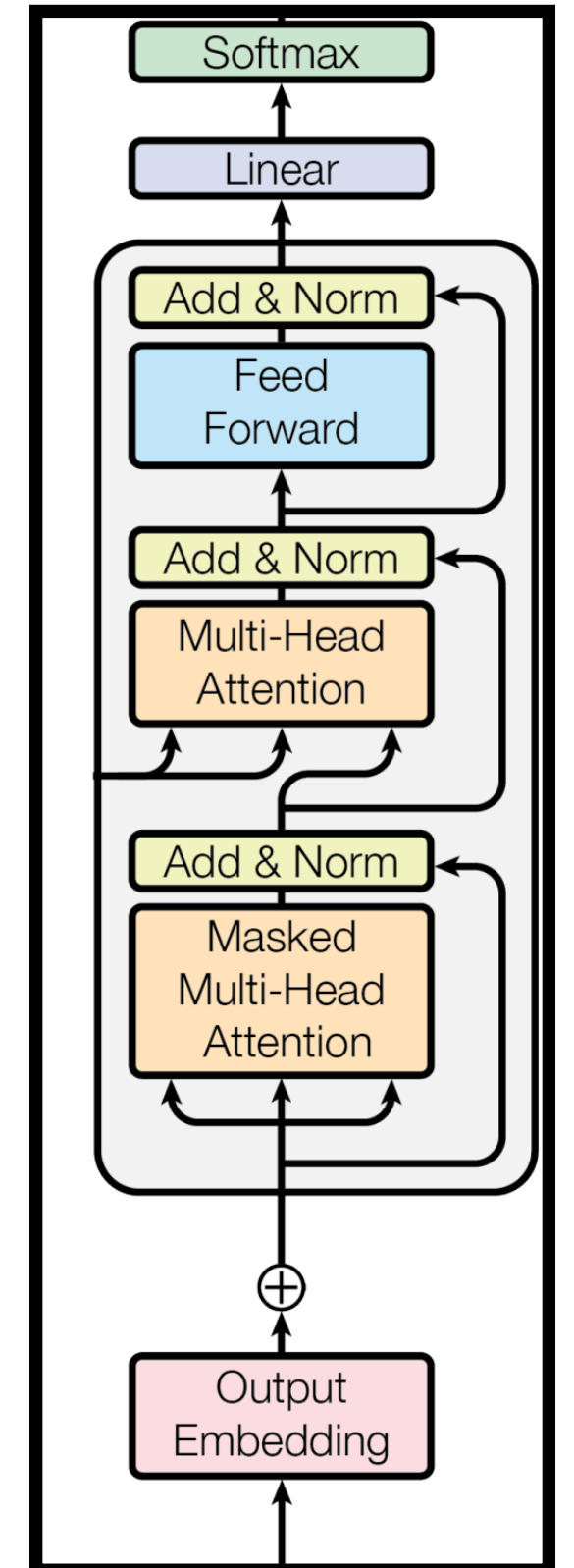
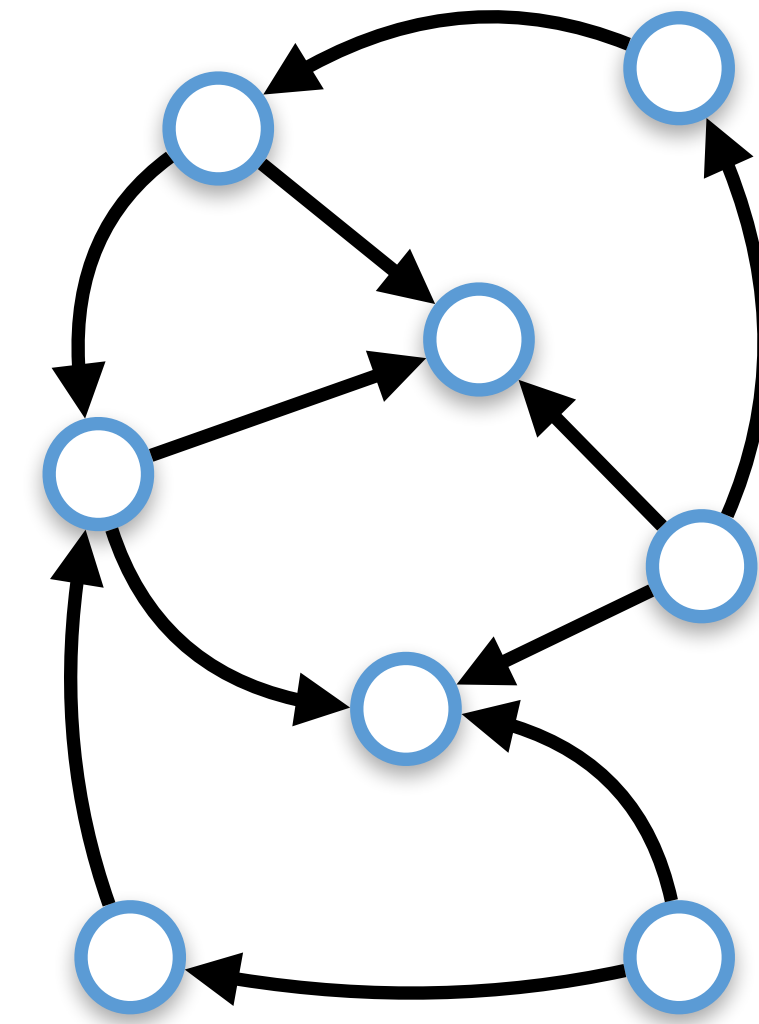


Domain Adaptation

Train on WMT News Crawl; Test+build datastore on Law

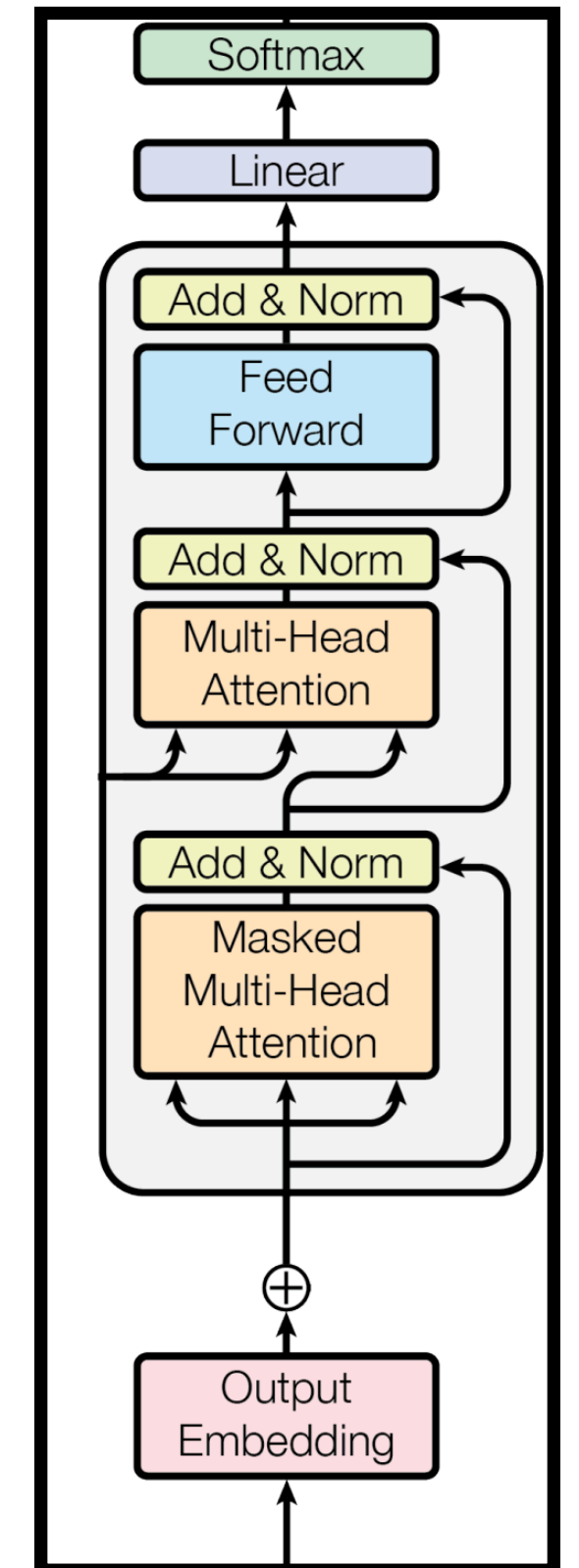
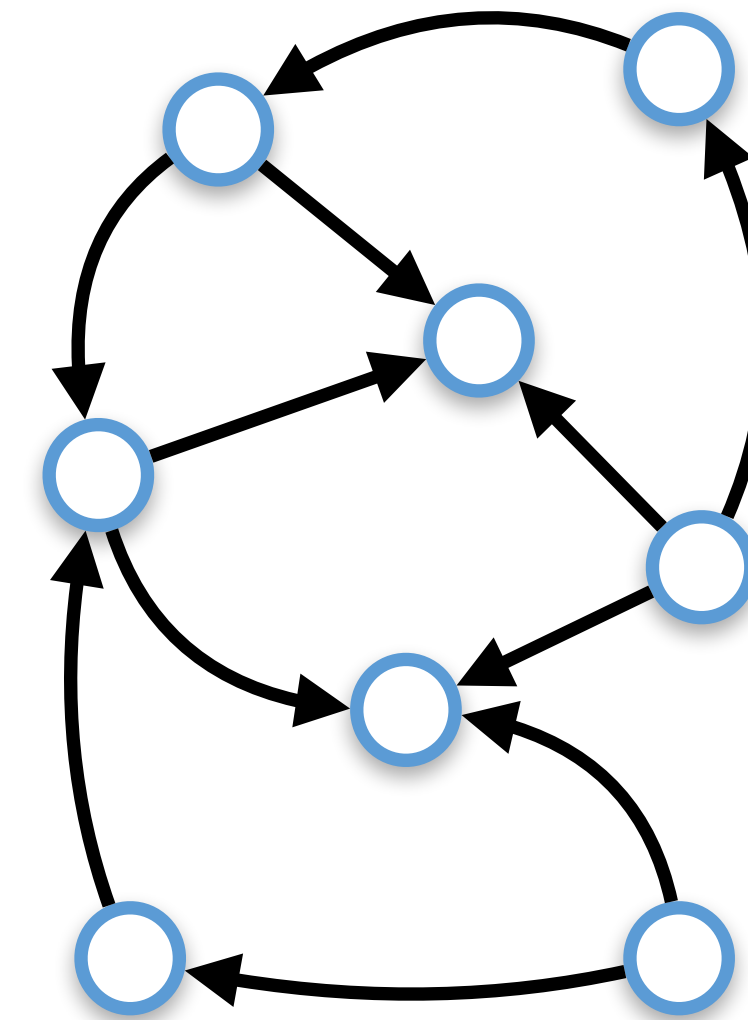


RetoMaton



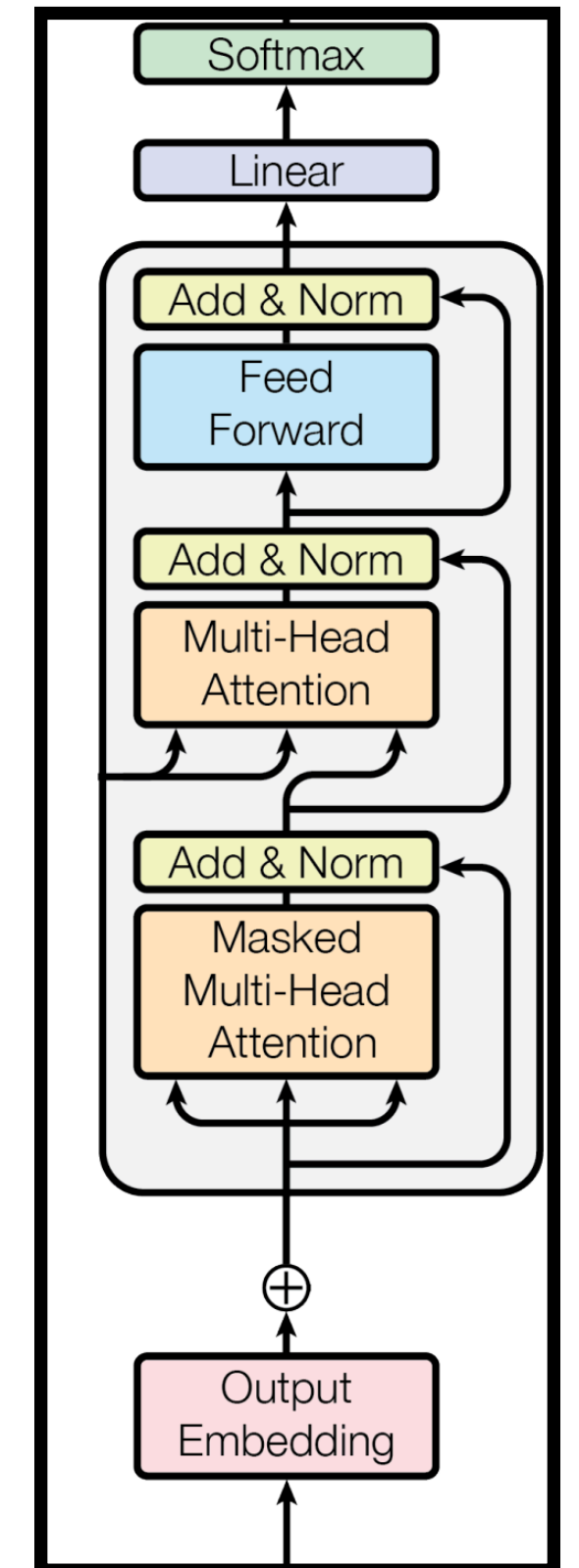
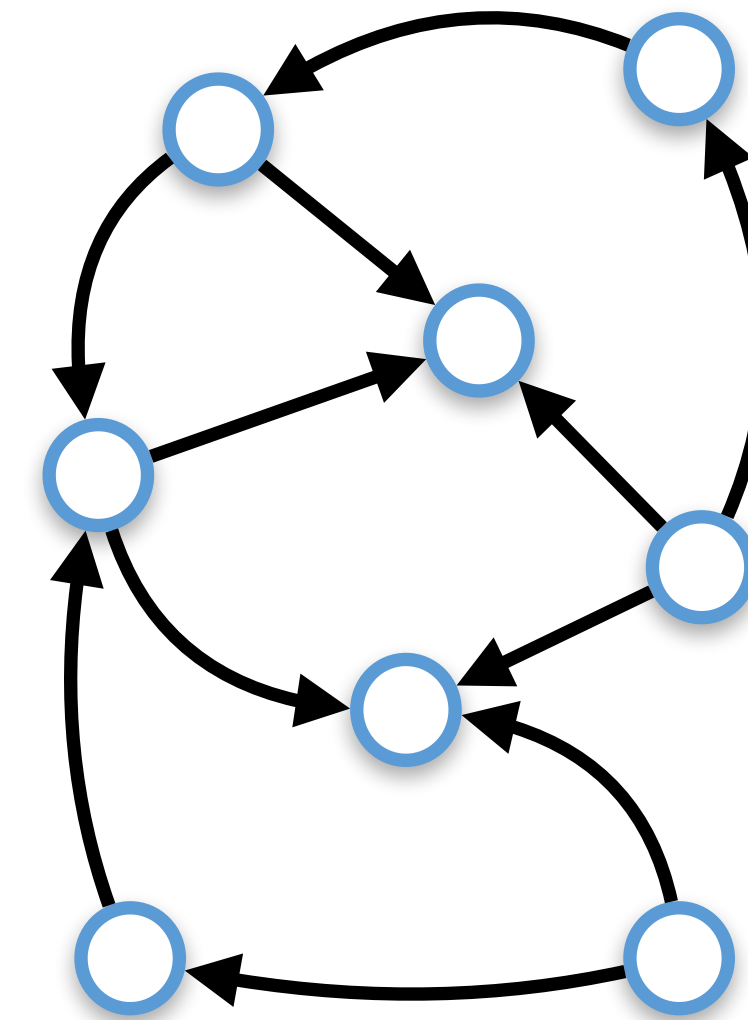
RetoMaton

- Synergy between a **symbolic** automaton and a **neural** LM



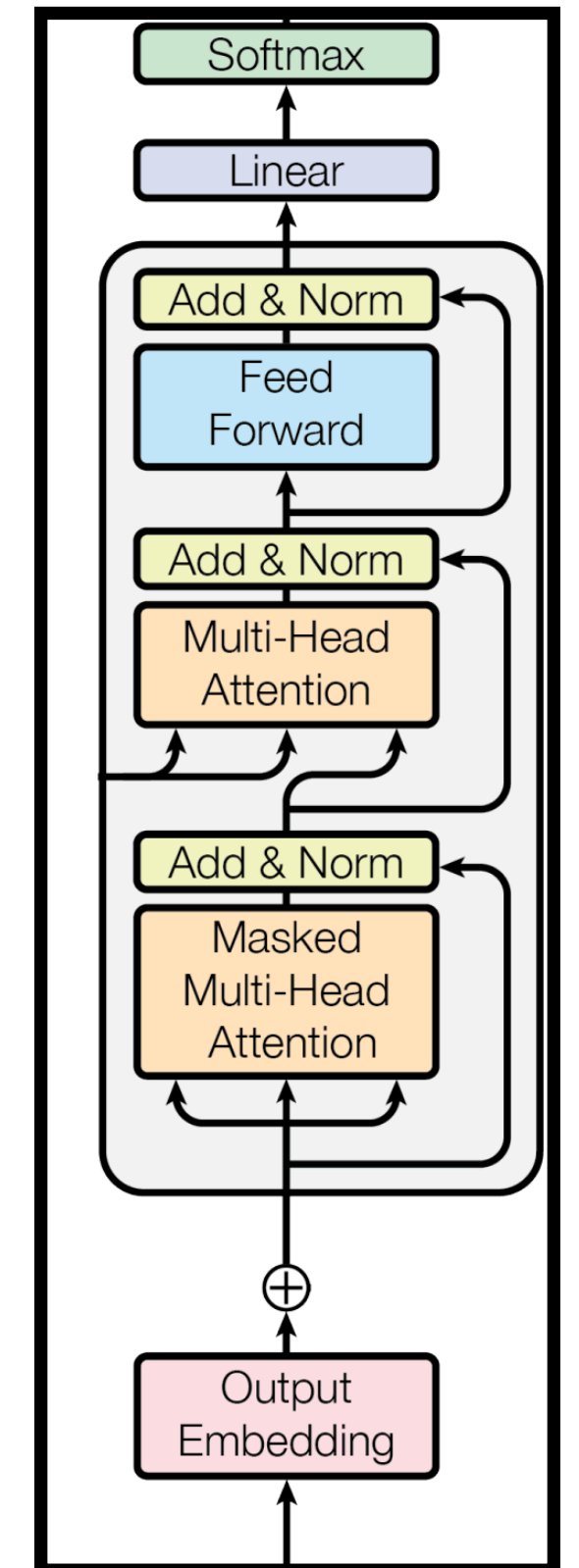
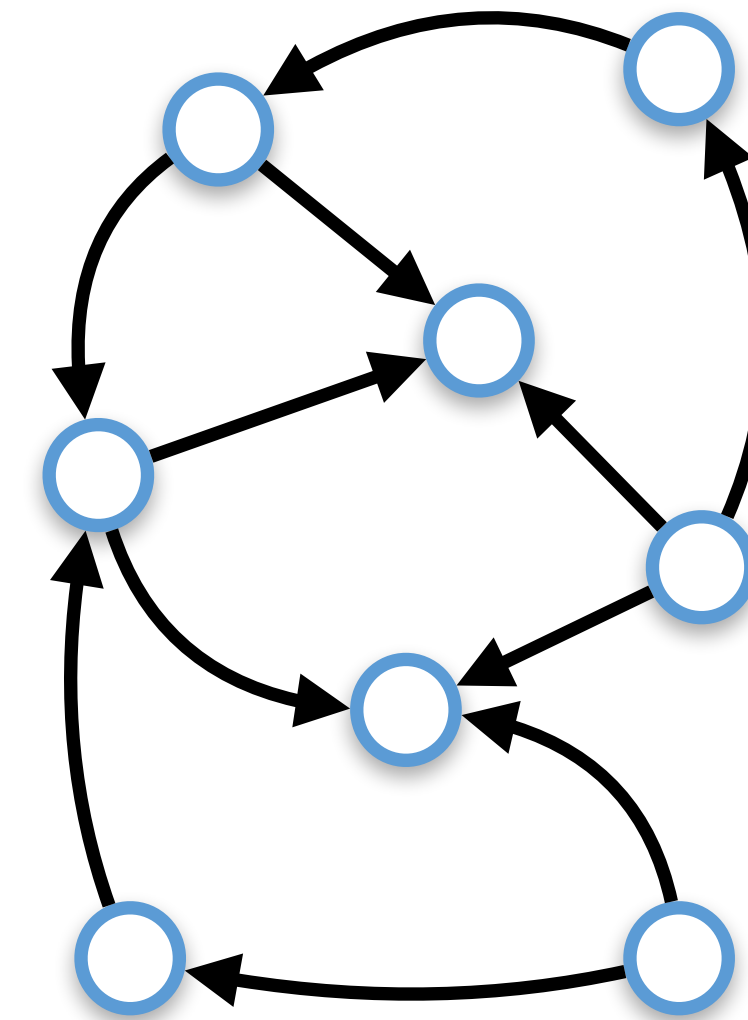
RetoMaton

- Synergy between a **symbolic** automaton and a **neural** LM
- Saving **pointers** between training entries
- **Clustering** of entries into automaton states



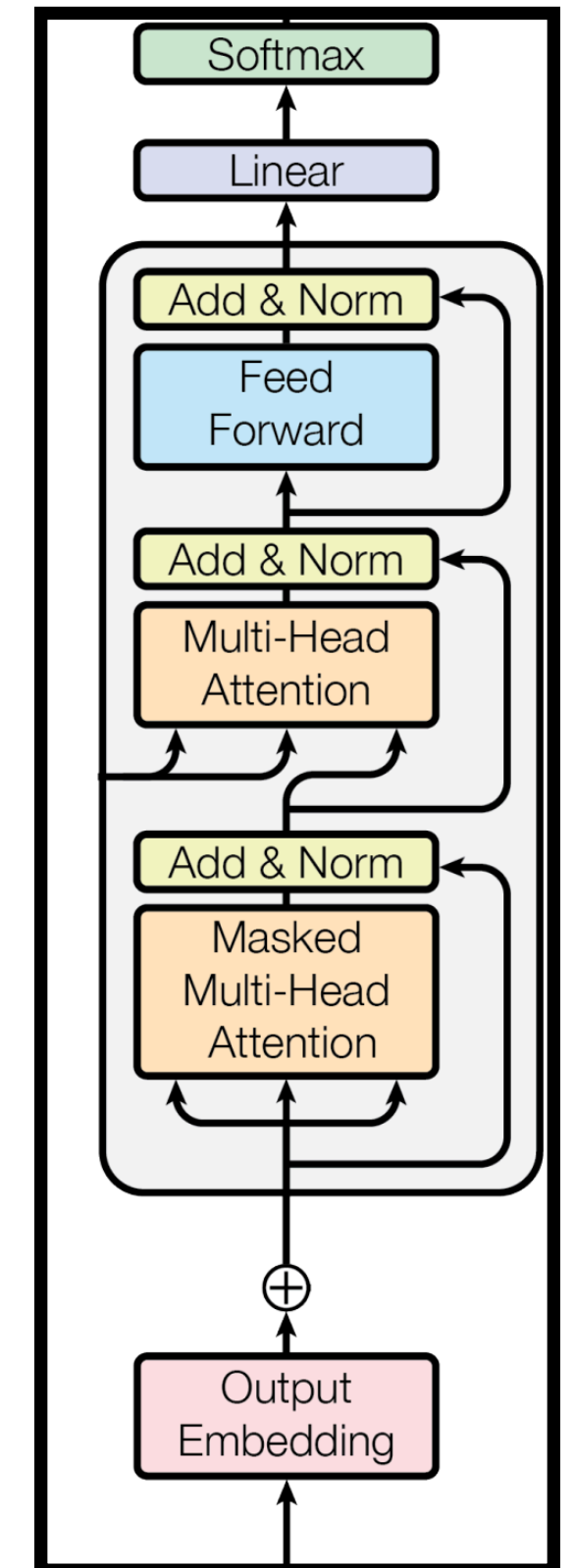
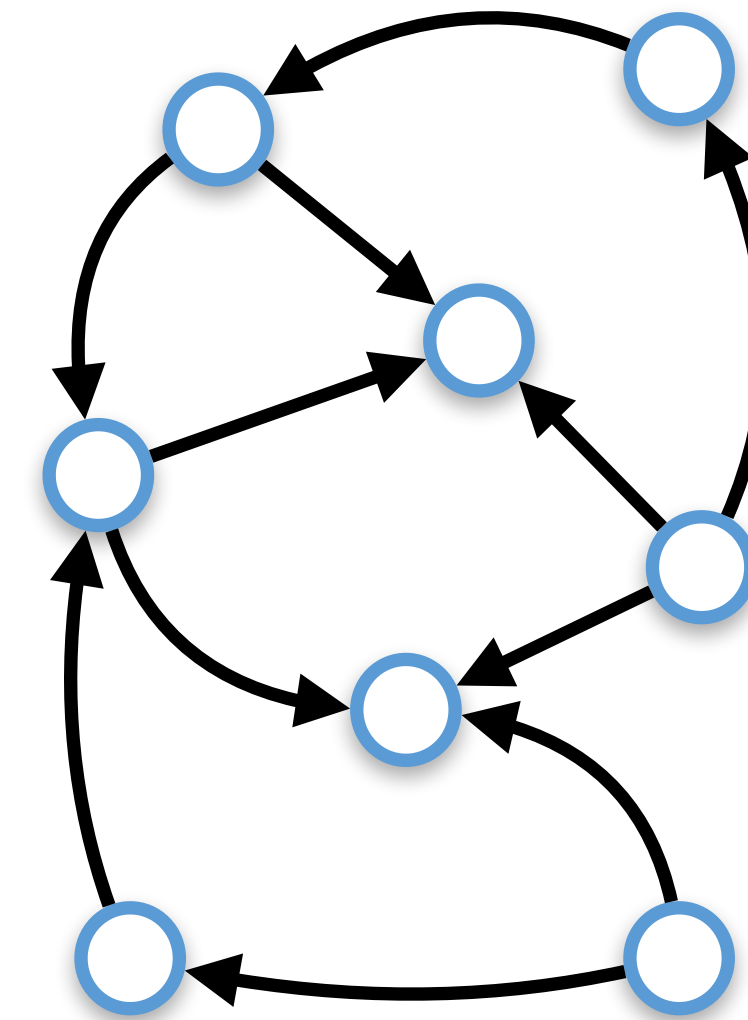
RetoMaton

- Synergy between a **symbolic** automaton and a **neural** LM
- Saving **pointers** between training entries
- **Clustering** of entries into automaton states
- **Dynamic** transition scores
- Lower perplexity than the base LM, while saving up to **83%** of the k NN searches compared to k NN-LM



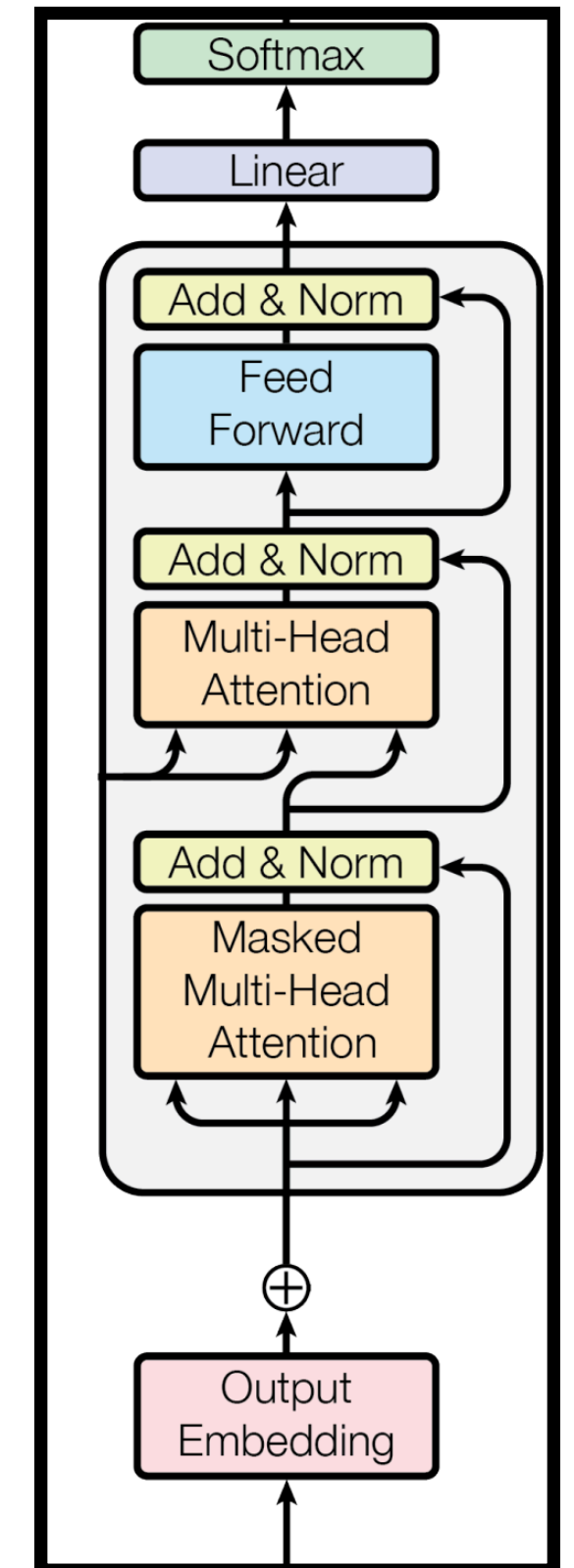
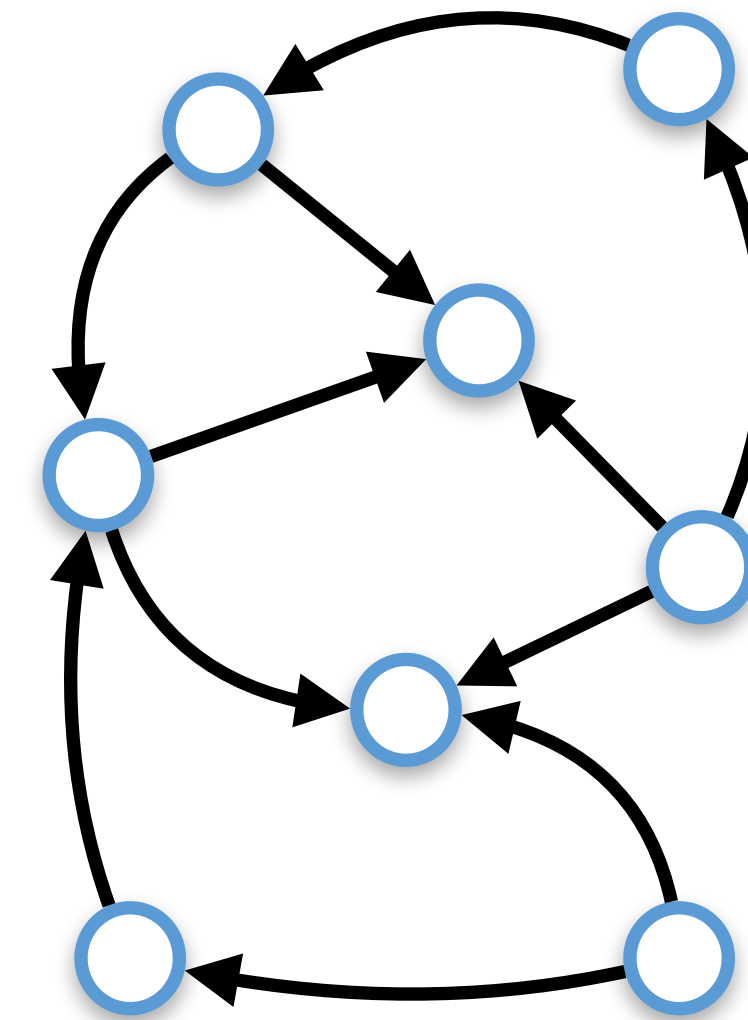
RetoMaton

- Synergy between a **symbolic** automaton and a **neural** LM
- Saving **pointers** between training entries
- **Clustering** of entries into automaton states
- **Dynamic** transition scores
- Lower perplexity than the base LM, while saving up to **83%** of the k NN searches compared to k NN-LM
- The creation of the automaton is **unsupervised**



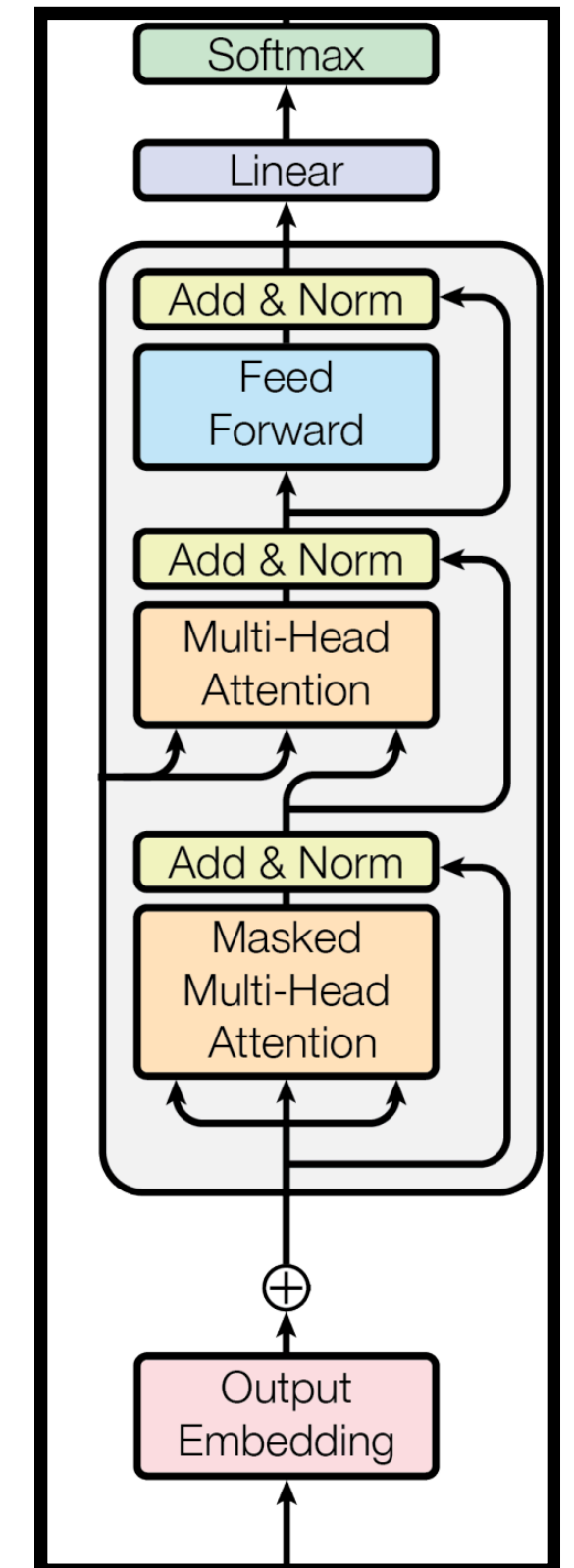
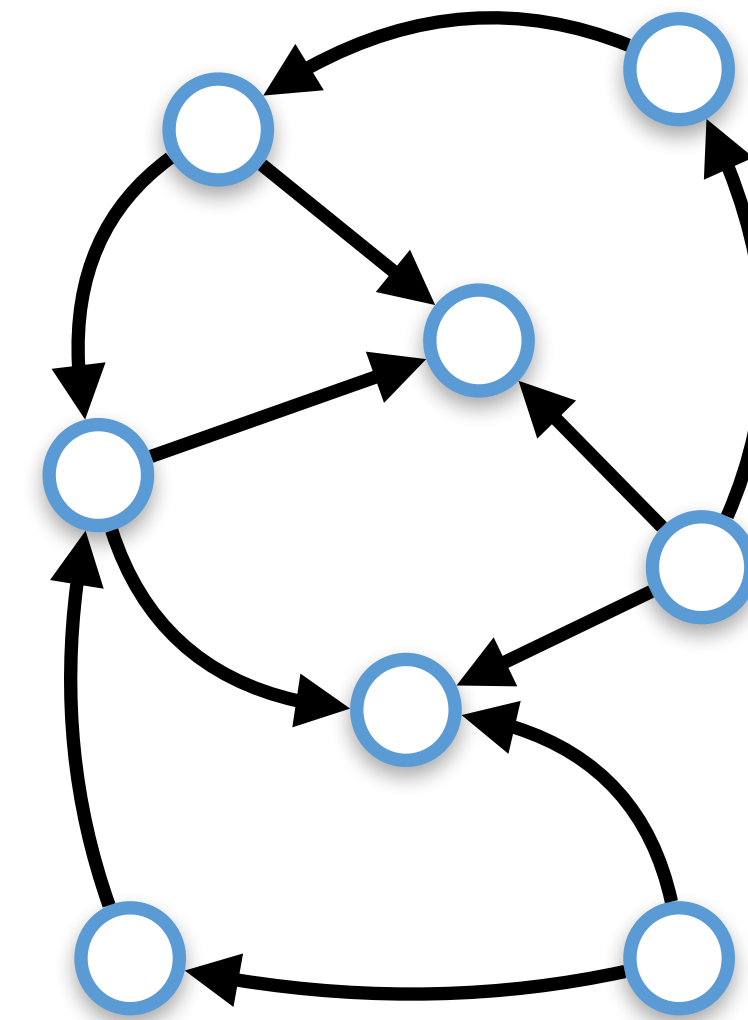
RetoMaton

- Synergy between a **symbolic** automaton and a **neural** LM
- Saving **pointers** between training entries
- **Clustering** of entries into automaton states
- **Dynamic** transition scores
- Lower perplexity than the base LM, while saving up to **83%** of the k NN searches compared to k NN-LM
- The creation of the automaton is **unsupervised**
- Constructed from the original training data



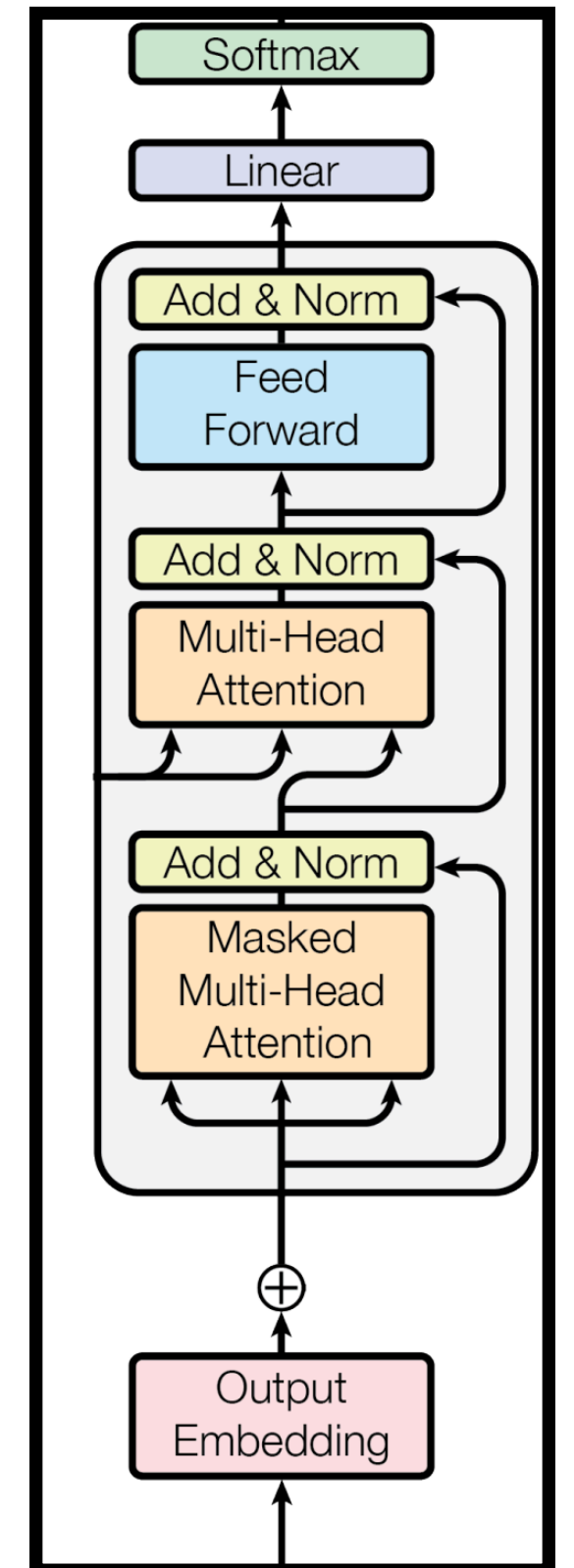
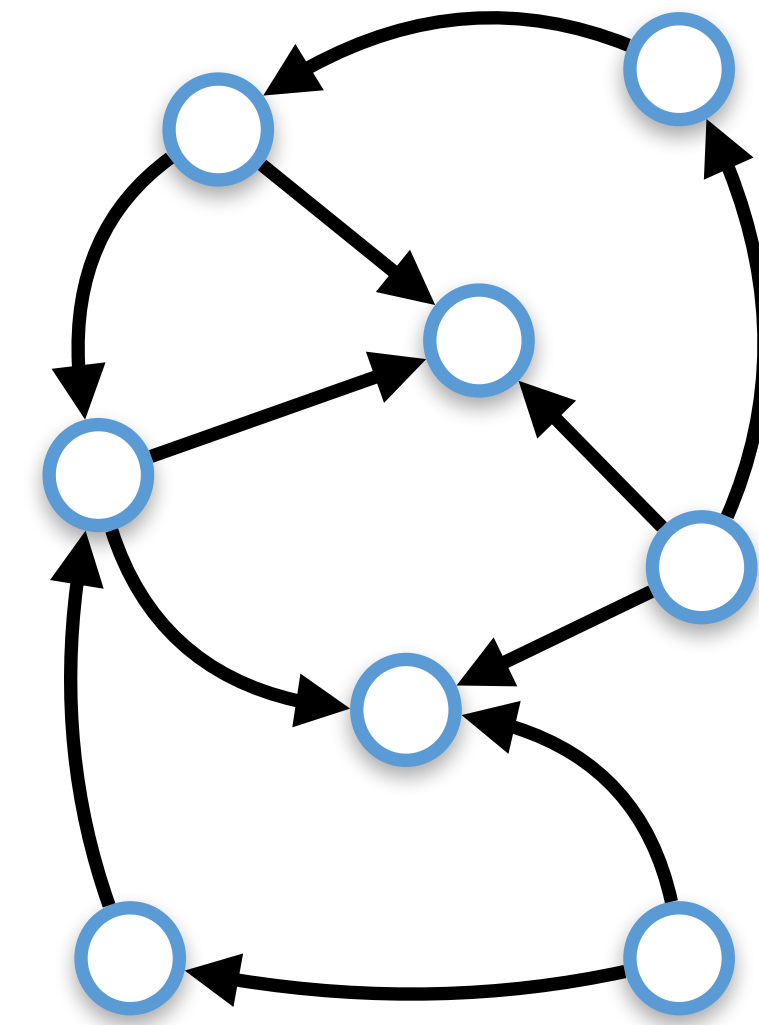
RetoMaton

- Synergy between a **symbolic** automaton and a **neural** LM
- Saving **pointers** between training entries
- **Clustering** of entries into automaton states
- **Dynamic** transition scores
- Lower perplexity than the base LM, while saving up to **83%** of the k NN searches compared to k NN-LM
- The creation of the automaton is **unsupervised**
 - Constructed from the original training data
 - Another domain



RetoMaton

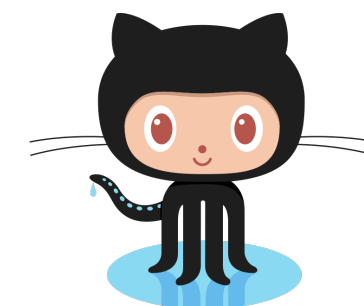
- Synergy between a **symbolic** automaton and a **neural** LM
- Saving **pointers** between training entries
- **Clustering** of entries into automaton states
- **Dynamic** transition scores
- Lower perplexity than the base LM, while saving up to **83%** of the k NN searches compared to k NN-LM
- The creation of the automaton is **unsupervised**
- Constructed from the original training data
- Another domain



Please visit our poster session 6-8PM!

<http://urialon.ml>

ualon@cs.cmu.edu



<https://github.com/neulab/retomaton>

<https://github.com/neulab/knn-transformers>