Neuro-Symbolic Language Modeling with Automaton-augmented Retrieval

Appeared in ICML'2022

Uri Alon, Frank F. Xu, Junxian He, Sudipta Sengupta, Dan Roth, Graham Neubig

Carnegie Mellon University
Language Technologies Institute

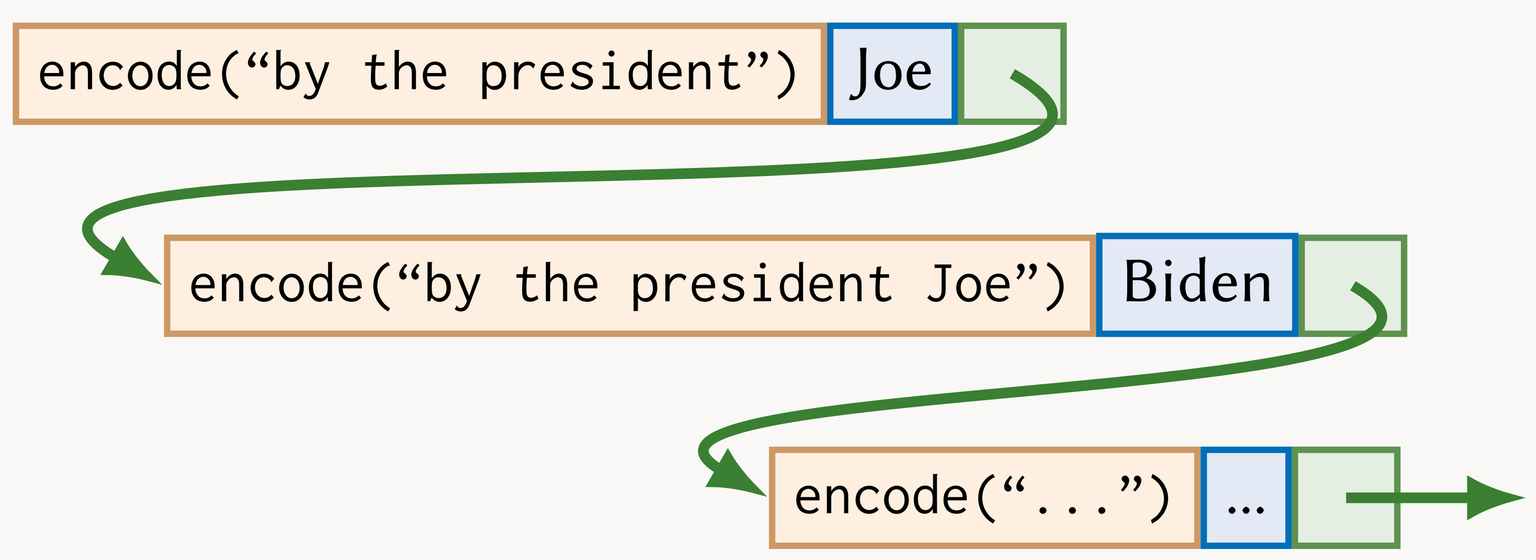
aws

https://github.com/neulab/retomaton

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

Key Idea #1: Pointers Between Examples

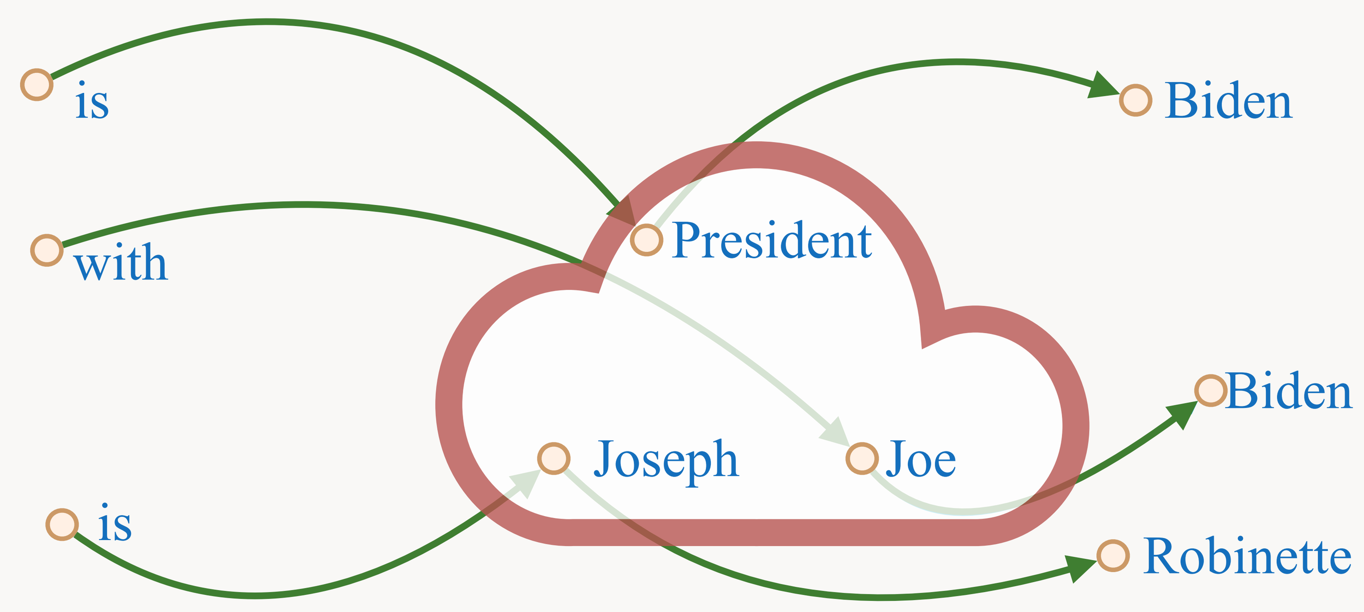
Encode the training set as linked lists of (*key*, *value*, *pointer*):



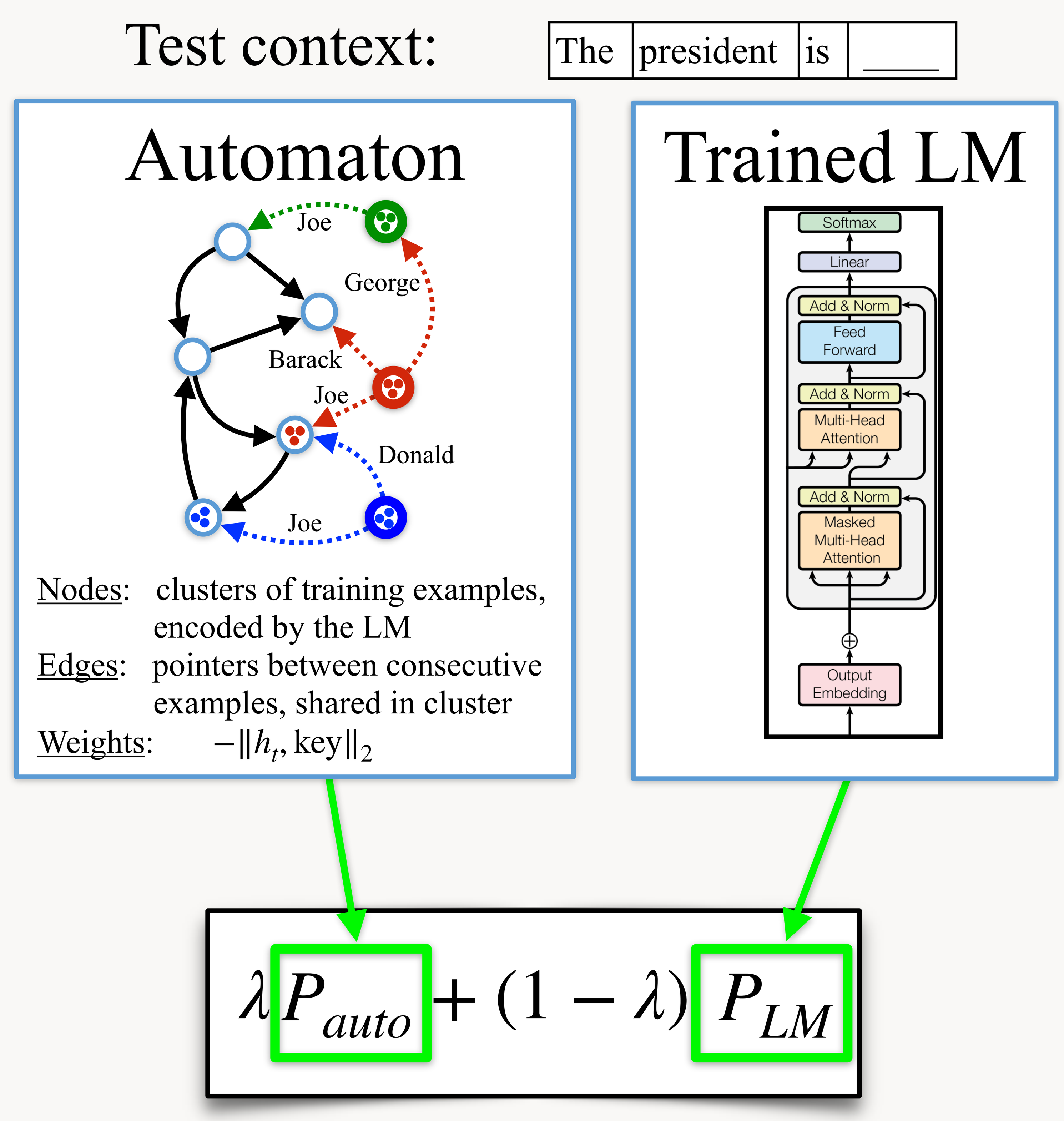
This extends kNN-LM (Khandelwal et al., '2020) by adding *pointers* between consecutive datastore entries.

Key Idea #2: Clustering Similar Keys

Cluster similar keys into automaton states:



RetoMaton



$$P_{auto}\left(w \mid h_t, states_t\right) \propto \sum_{s \in states_t} \sum_{(key,val) \in s, val=w} exp\left(-\|h_t, \mathrm{key}\|_2\right)$$

Results

In-domain Datastore

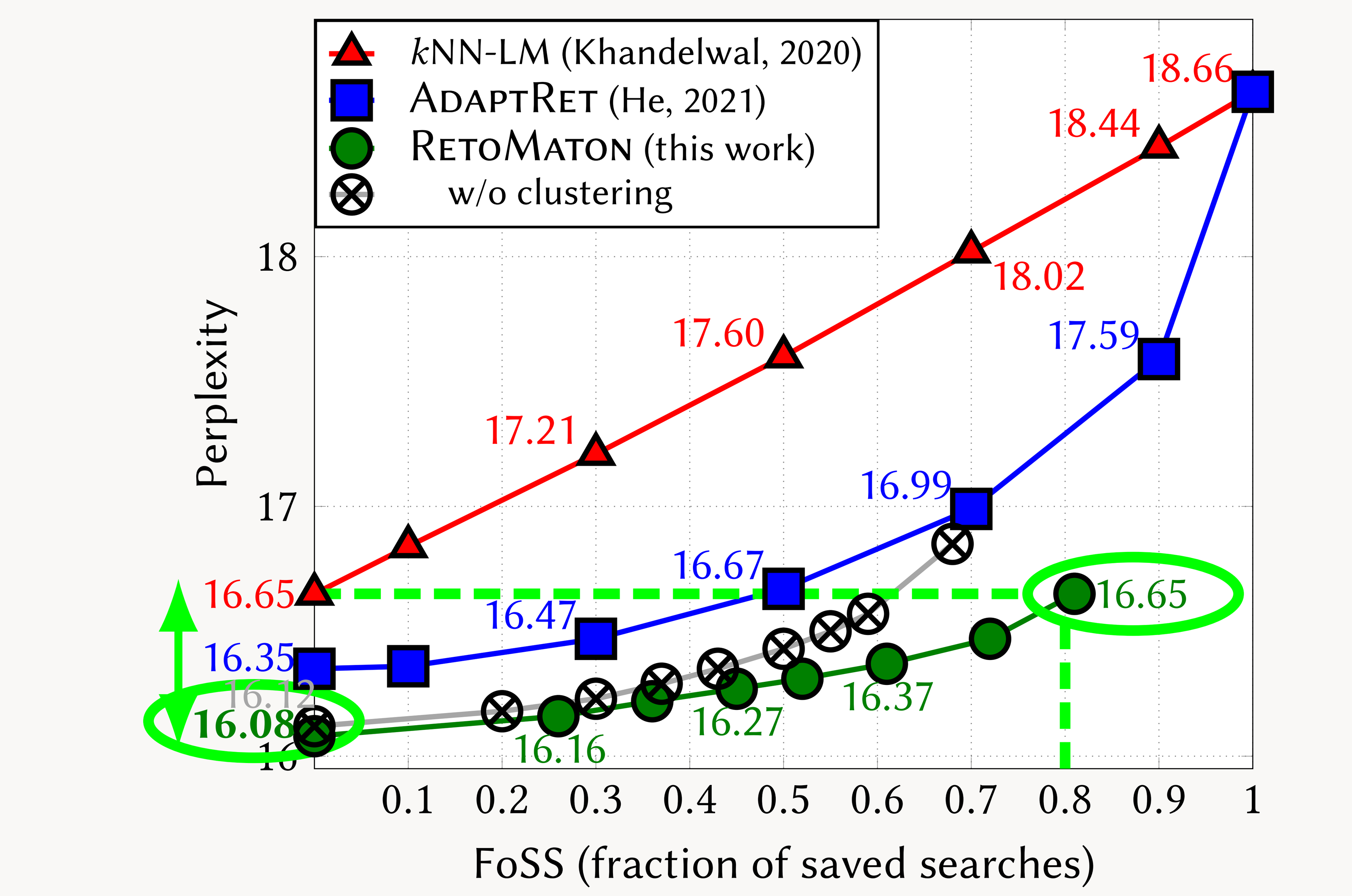


Figure: Experiments on Wikitext-103, where the datastore is created from the same training set that the base LM was trained on.

Domain Adaptation

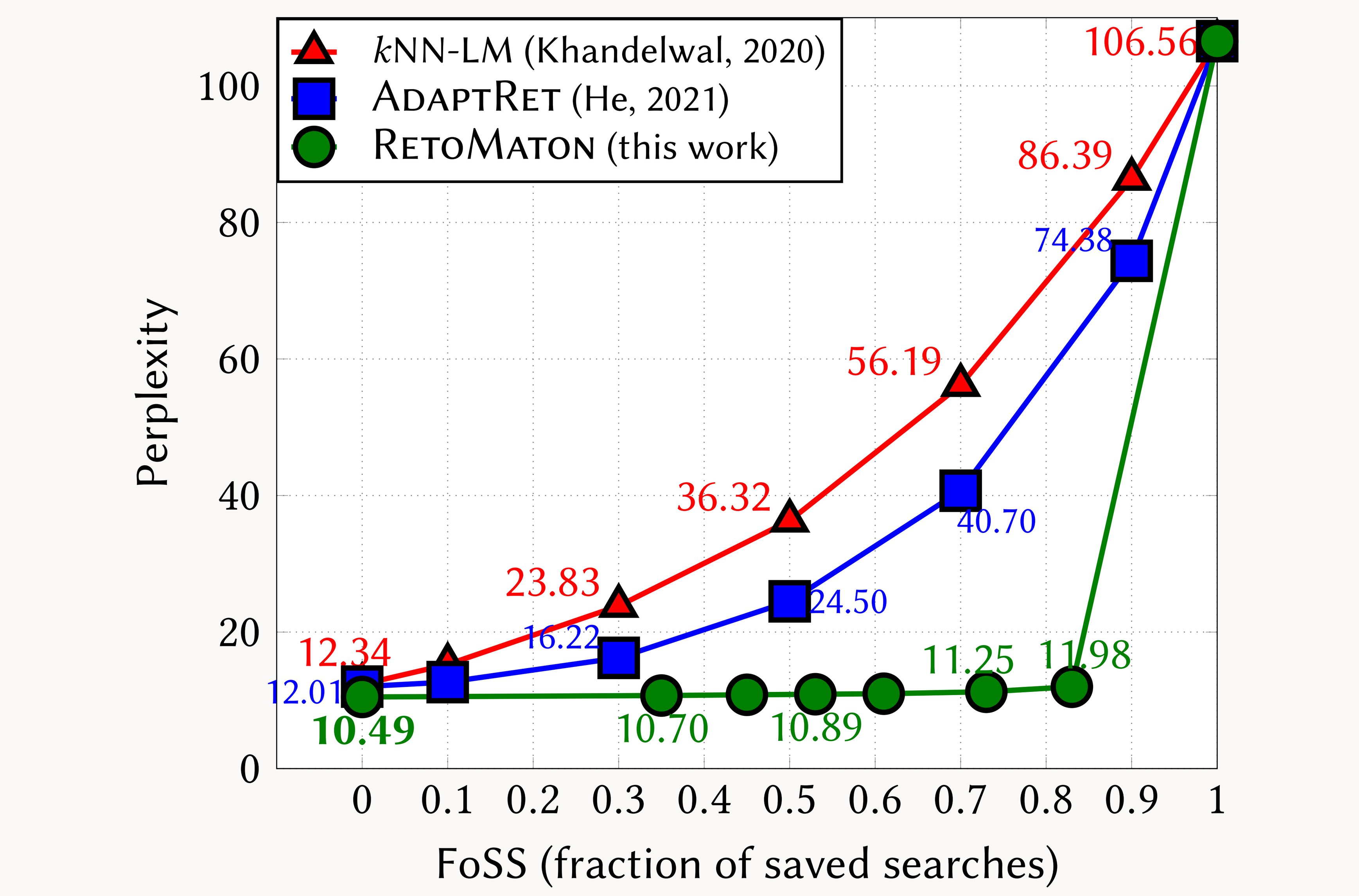


Figure: Domain adaptation experiments: the model was trained on News Crawl, and the datastore is constructed from Law-MT.

Improving Fine-tuning

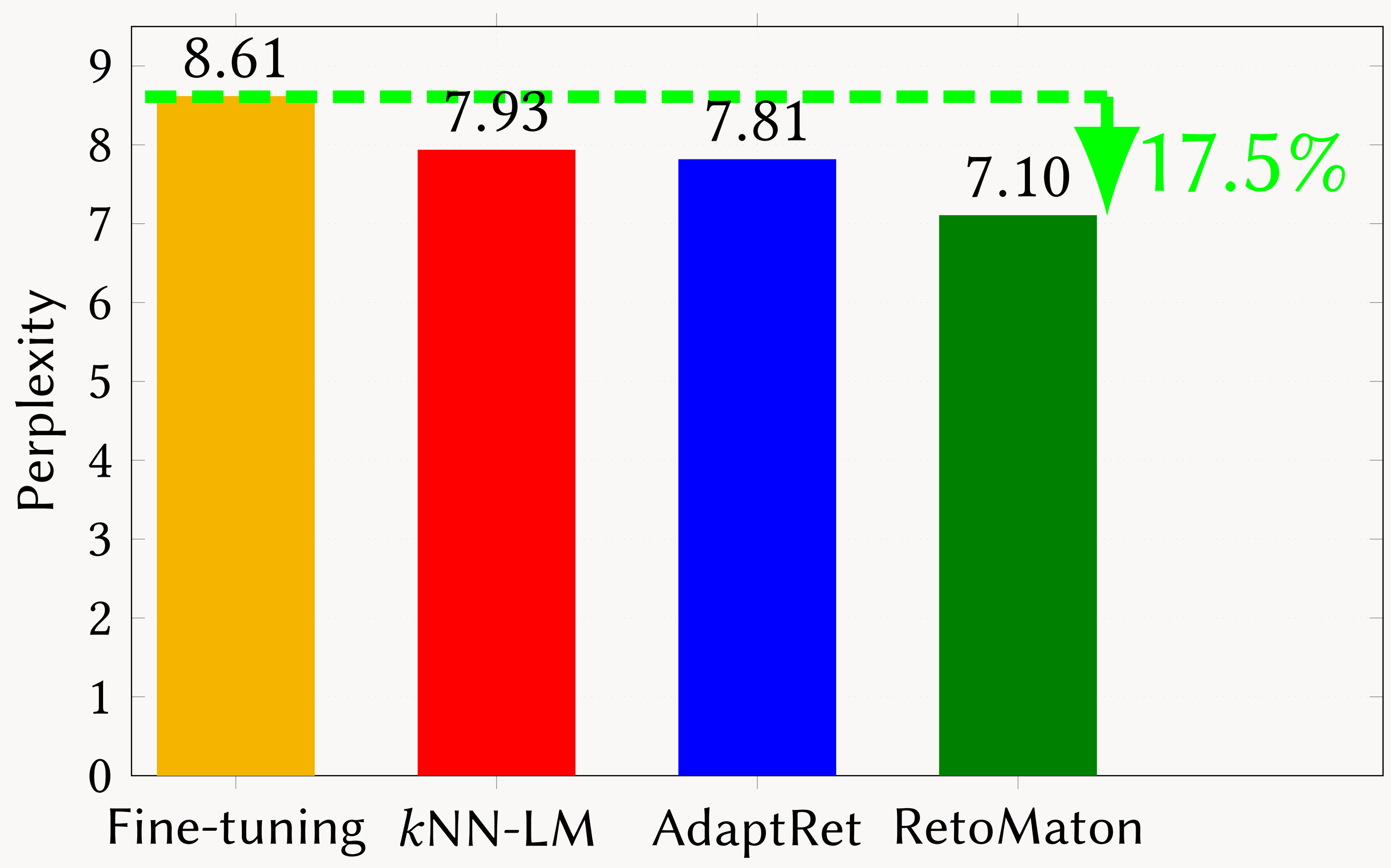


Figure: When constructing RetoMaton on top of a fine-tuned model, RetoMaton reduces perplexity by 17.5%.

Sample

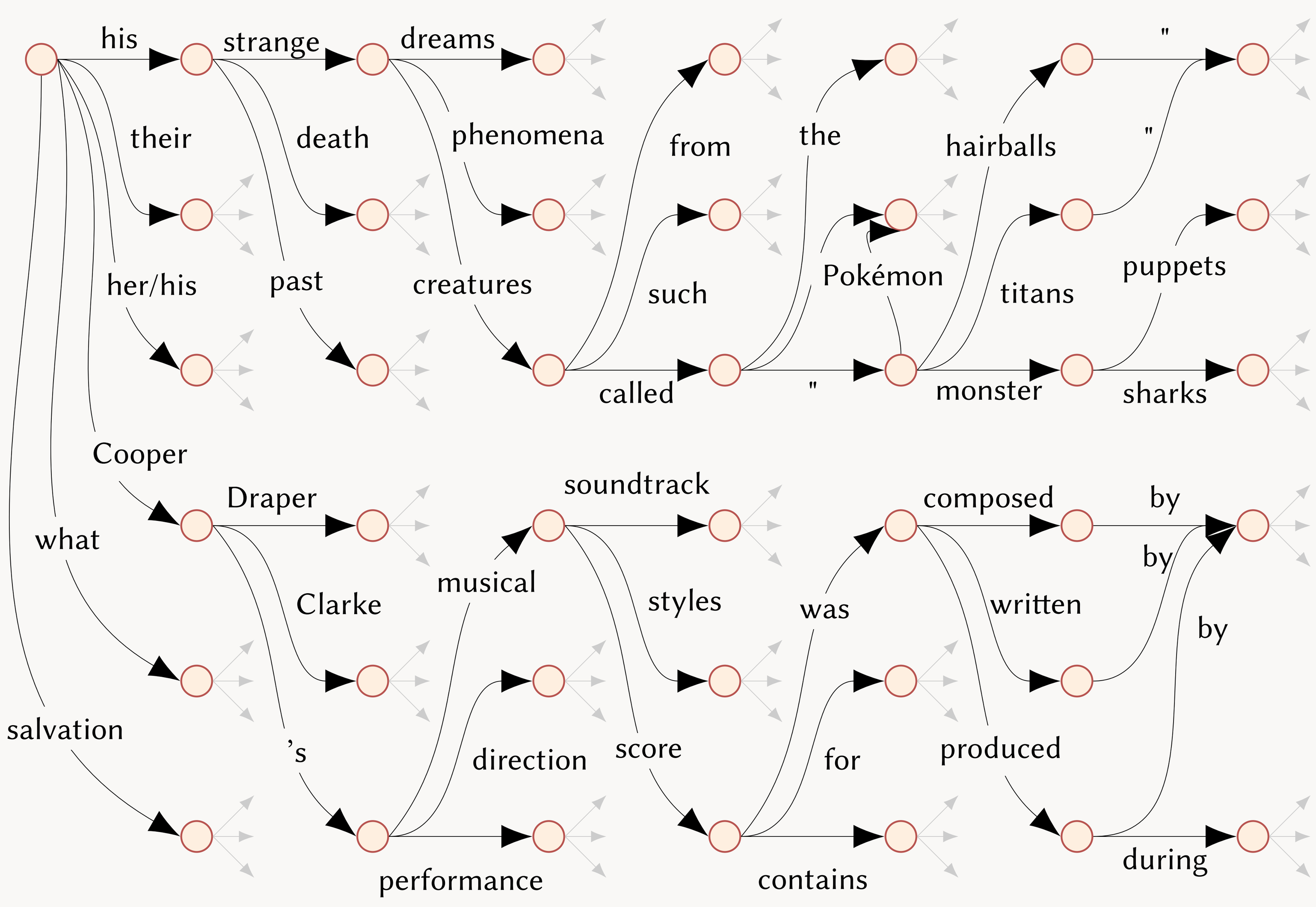


Figure: A sample of the automaton constructed from Wikitext-103