Neuro-Symbolic Language Modeling with Automaton-augmented Retrieval

Carnegie Mellon University
Language Technologies Institute

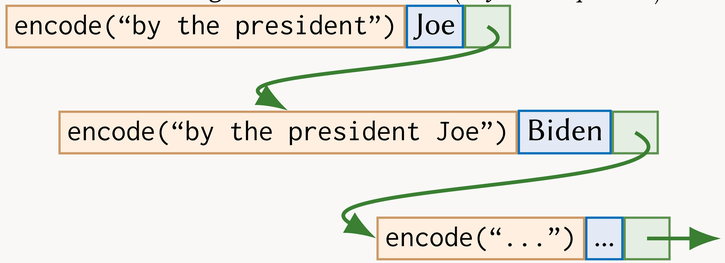
aws

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

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

Key Idea #1: Pointers Between Examples

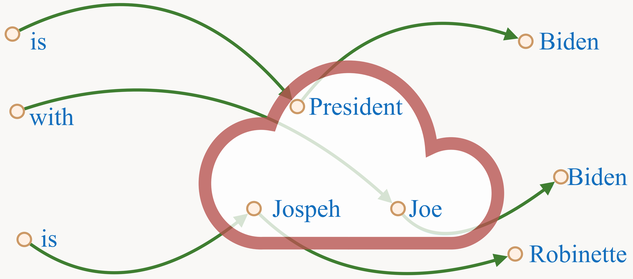
Encode the training set as linked lists of (*key*, *value*, *pointer*):



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

Key Idea #2: Clustering Similar Keys

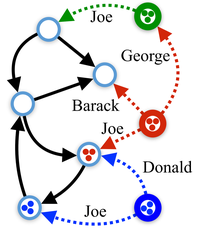
Cluster similar keys into automaton states:



RetoMaton

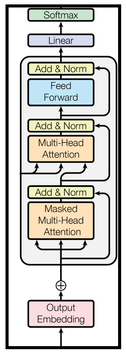
Test context: The | president | is | ____

Automaton



Nodes: clusters of training examples, encoded by the LM
Edges: pointers between consecutive examples, shared in cluster
Weights: $-\|h_t, \text{key}\|_2$

Trained LM



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

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

In-domain Datastore

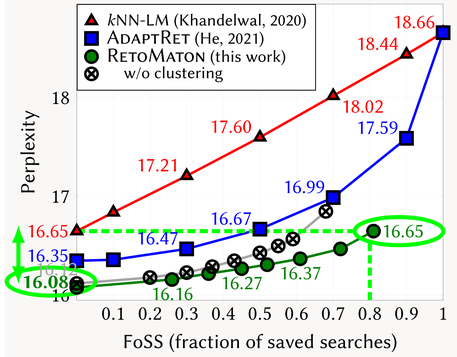


Figure: Experiments on WikiText-103, where the datastore is created from the same training set that the base LM was trained on.

Domain Adaptation

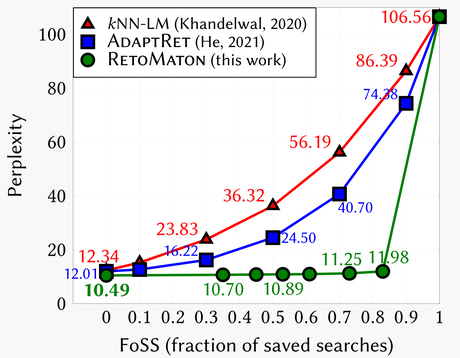


Figure: Domain adaptation experiments: the model was trained on News Crawl, and the datastore is constructed from Law-MT.

Improving Fine-Tuning

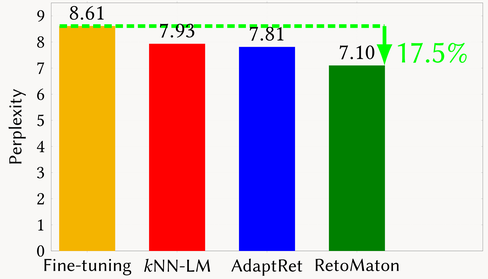


Figure: When constructing RetoMaton on top of a fine-tuned model, RetoMaton reduces perplexity by 17.5%.

Sample

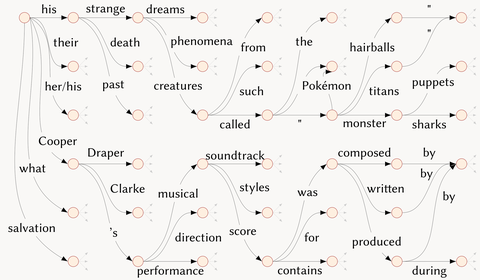


Figure: A random sample of the automaton constructed from the training set of WikiText-103