SpARC: Sparse Activation Regularization for Logical Consistency

Sarthak Consul, Samar Khanna, Julia Xu
{sarthakc, samar99, juliaxu} [at] stanford.edu

Problem and Motivation

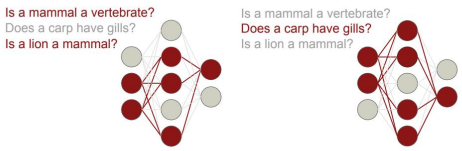
Language models store large amounts of linguistic data and contain extensive world knowledge from which relational knowledge can be captured. Although it is possible to recover factual and commonsense knowledge, the problem that arises is making them consistent with each other. Logical consistency is the notion that a model will reason consistently over a set of implications.

Given a set of implications: "A lion is a mammal" and "A mammal is a vertebrate"

We want to correctly answer the question: "Is a lion a vertebrate?"

Background

Csordas *et. al.* argue that imposing modularity in neural networks would allow for the modules to be reused for a greater consistency in networks. We build upon this idea by finetuning a pretrained language model with an auxiliary loss that penalizes the network for non-sparse activations when given similar prompts.



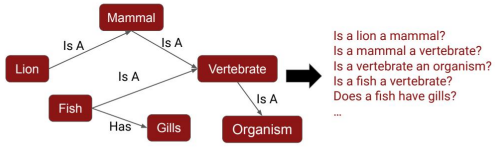
We base our approach on MACAW a pretrained general-purpose question-answering model built on top of T5. We evaluate the performance of our model based on two metrics: accuracy (in terms of the F1-score) and consistency, which is the complement of the constraint violation, τ. The consistency equations is given below, where S_1 and S_2 are boolean formulas constructed from model predictions on x in dataset D.

$$\tau = \frac{\sum_{x \in D} \left[\bigvee_{(S_1, S_2)} \neg (S_1(x) \rightarrow S_2(x)) \right]}{\sum_{x \in D} \left[\bigvee_{(S_1, S_2)} S_1(x) \right]}$$

$$\text{Consistency} = 1 - \tau$$

Dataset

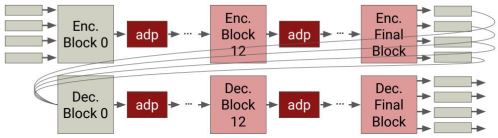
We generate and use a custom QA dataset which is built on constraints and facts given by BeliefBank. We generate questions through traversing a fact constraint graph and design question templates according to English grammar rules.



Methodology

Finetuning

Given the comparatively small size of the dataset, finetuning on all parameters of the transformer is susceptible to overfitting. To mitigate this effect, we introduce small perturbation layers (termed as adapters) in between the transformer layers as the only trainable parameters.

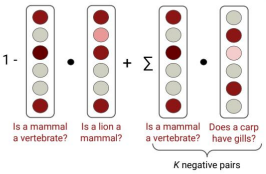


Meng *et. al.* demonstrated that mid-layer token embeddings significantly influence the model's predictive efficacy, which informs our loss function's design to sparsify the model's mid-layer activations.

Sparsity via L1 Loss

$$Loss = CrossEntropy(y, \hat{y}) + \lambda \sum_{k=1}^{N} \sum_{i=1}^{T} ||A_k(token_i)||_1$$

Similarity



We employ a contrastive similarity loss among positive and negative pairs. The loss maximizes the cosine similarity between positive pairs and minimizes between negative pairs.

The positive pairs are generated in 3 separate ways:

- Adjacent connections ("Does a fish have vertebrae?" and "Does a fish have gills?")
- Linked connections ("Is a lion a mammal?" and "Do mammals have vertebrae?")
- Cosine similarity of embeddings given by SimCSE, an NLI model

To mitigate the effect of non-entity tokens, we use activations corresponding to:

- The last token corresponding to the common entity
- The token corresponding to "EOS"
- The last token corresponding to "$answer$"

Future Work

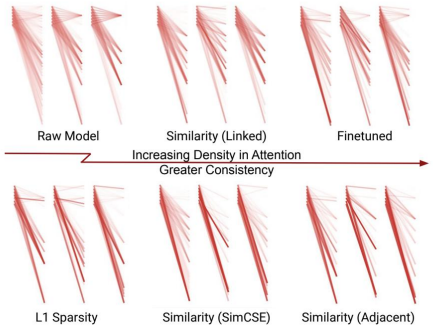
- Directly enforce sparsity through pruning (e.g. magnitude, movement) methods
- Use momentum encoding (MoCo) to increase the number of negative samples
- Run experiments on a larger dataset with more complex entailment questions

Experimental Results.

Method	Layer	λ	F1 (%)	Consistency (%)
Raw Model	-	-	85.75	52.36
Finetune	-	-	93.24	86.70
L1 Sparsity	Dec. Block 12	1×10^{-5}	94.90	89.59
L1 Sparsity	Enc. Final Block	1×10^{-5}	92.58	86.10
Similarity (Linked)	Dec. Block 12	2×10^{-3}	92.88	79.15
Similarity (Linked)	Enc. Final Block	2×10^{-3}	92.29	78.44
Similarity (Adj)	Dec. Block 12	2×10^{-3}	94.85	82.25
Similarity (Adj)	Enc. Final Block	2×10^{-3}	94.79	**91.00**
Similarity (SimCSE)	Dec. Final Block	2×10^{-3}	**96.81**	89.92
Similarity (SimCSE)	Enc. Final Block	2×10^{-3}	94.99	90.93
Common Token	Enc. Block 12	2×10^{-3}	95.22	83.15
EOS Token	Enc. Final Block	2×10^{-3}	93.30	83.51

Discussion and Analysis

- Finetuning the adapter model outperforms finetuning the full model or only the last layer weights (due to overfitting and underfitting, respectively)
- Enforcing L1 sparsity led to minor improvements in performance
- The SimCSE-based dataset is an approximation of the adjacency-based one
- Adjacency and SimCSE similarity experiments led to greatest improvements



- Adjacency-based outperforms linked-based since it has a more diverse set of answer pairs (linked-based can only contain yes-yes and yes-no pairs)
- Token similarity underperforms due to uncertainty in optimal index position
- Due to small batch sizes (64 at maximum), the convergence of our contrastive similarity loss suffers from a dearth of negative samples

Modular activations promise better logical consistency