Data Independent Pruning of Language Models

Ayesha Qamar

-

Motivation

- Pre-trained LMs are used ubiquitously in NLP but there are computationally expensive.
- Extensive work on reducing the size of those models has been done.
    - The goal is to compress the model without sacrificing a lot on performance.

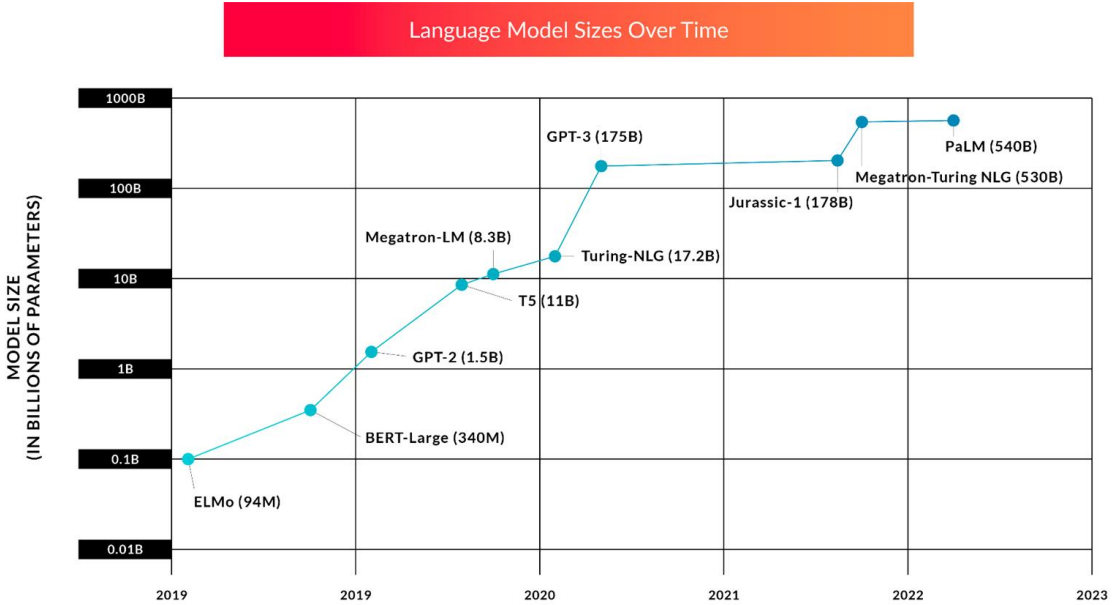
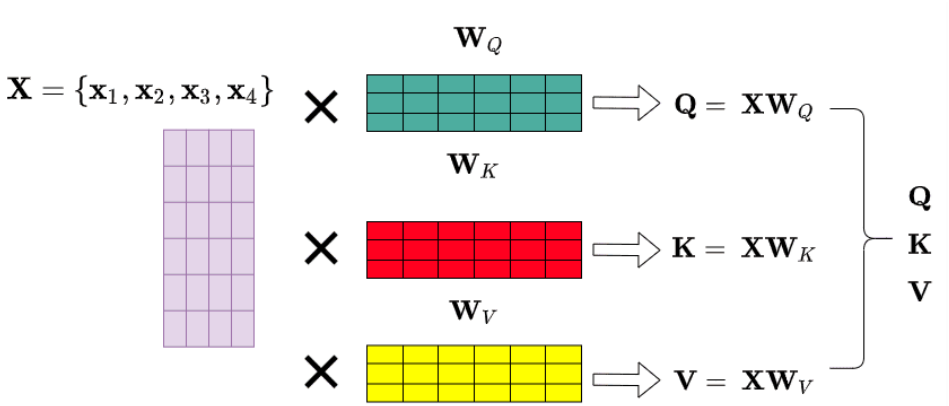
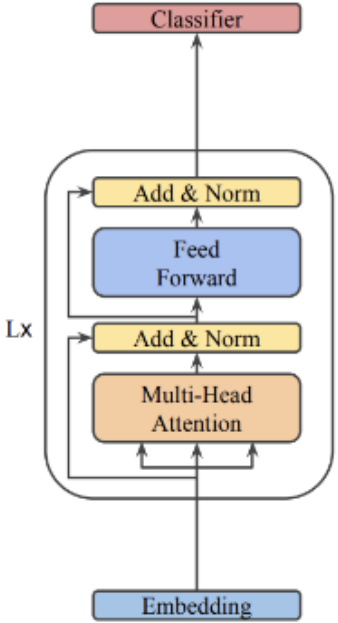


Image taken from https://twosigmaventures.com/blog/article/the-promise-and-perils-of-large-language-models/

Recap - Transformers

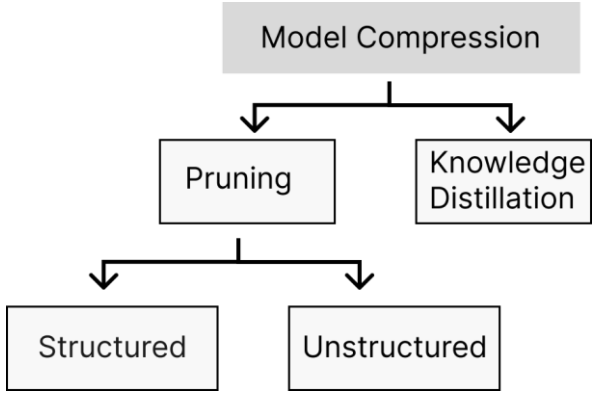
- Transformer based models consist of two main components that contribute to model size.
  - Multiheaded attention - makeup 33% of the total weights
  - Fully connected layers - 67% of the model weights
- Each attention head consists of four matrices.
  - Queries, keys, values, and output.



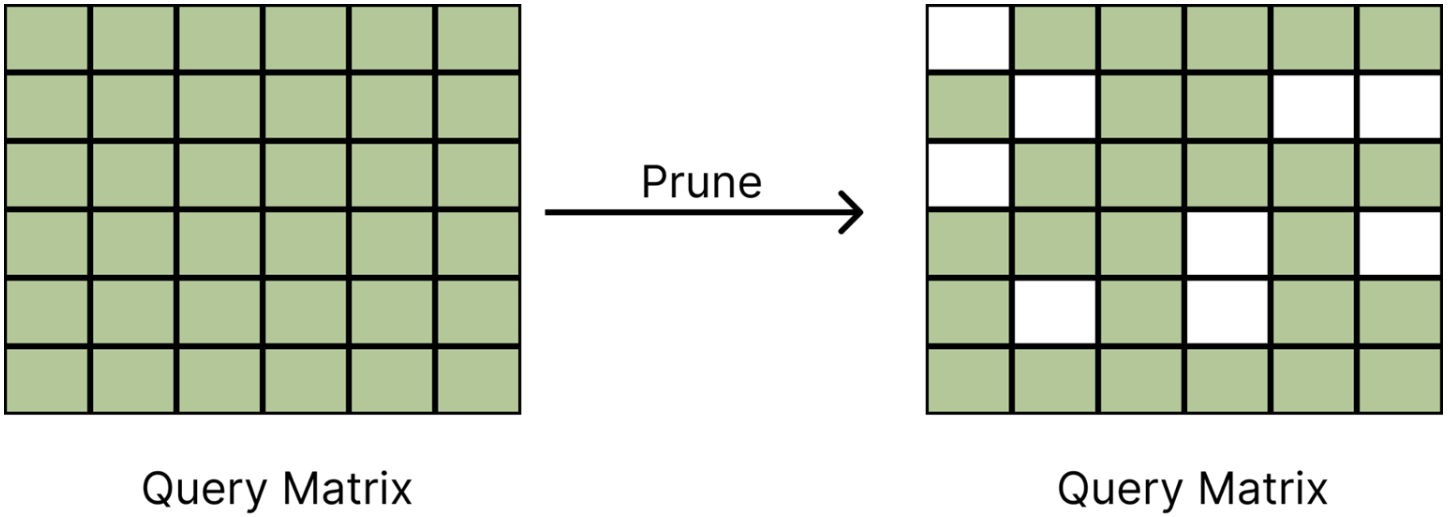


Prior work

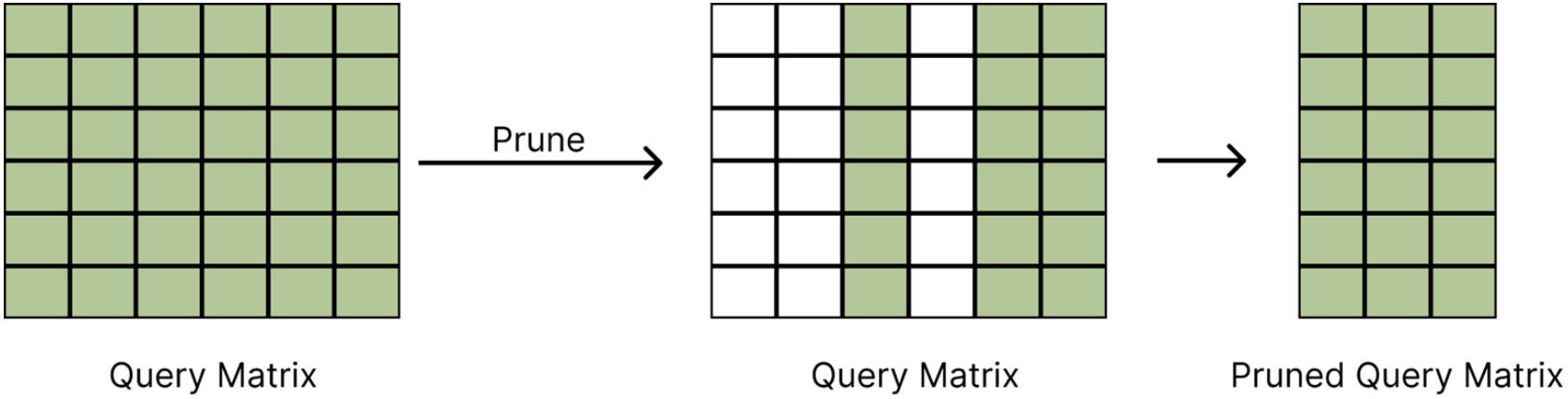
- Knowledge Distillation
  - Use a bigger teacher model to train a distilled version of it.
  - Need to pre-train from scratch!
  - DistilBert
- Pruning
  - Remove some components of the already pre-trained model
  - Structured
    - Remove the component as a whole, eg. some heads from the multihead attention, or even some full layers.
    - Limited choices for pruning.
  - Unstructured
    - Make the model sparse by removing weights by making them zero.
    - Can achieve high sparsity.
    - Does not actually give inference speedup.



Unstructured Pruning



Structured Pruning



How to decide what units to prune?

- Need to define some kind of importance
- Magnitude based pruning: use the absolute magnitude of the unit as its importance.
  - Works well on low sparsity but gives performance degradation for high sparsity settings.
- Gradient based pruning: for any task, look at how the gradients of that unit change during fine-tuning.
  - The bigger the change for a unit, the more important it is.
  - Data dependent!

Data independent structured pruning

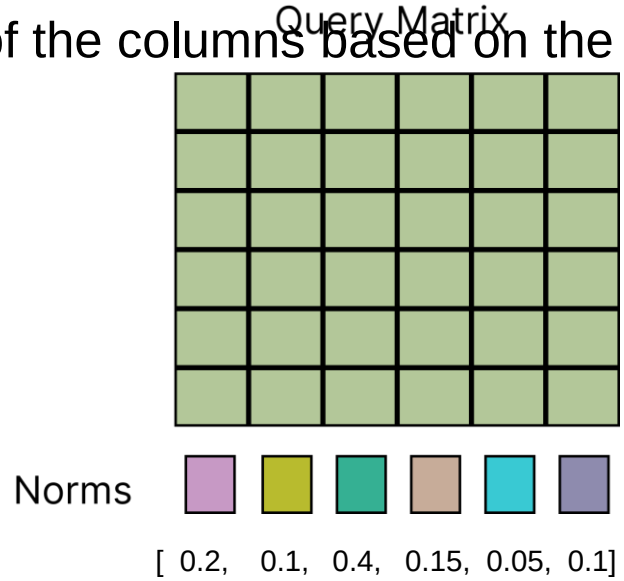
- We want inference speed up - structured pruning.
  - Is a much harder setting, limited choices for pruning units.
- Data independent pruning - magnitude based.
  - Does not perform as well as gradient based in high compression setting.

Coresets

- A coreset is a small, weighted subset of the original input set of items.
- A coreset would return us a smaller matrix
  - Retaining columns based on their importance.

Coreset for Query Matrix

- The importance of each column is measured based on its norm.
- Intuition: the larger the norm the more important that column is.
- Sample a subset of the columns based on the importance.



Dataset and Model

- SST-2
  - Single sentence sentiment analysis for movie reviews.
  - Positive, negative.
- Bert base model used in all experiments.

Results on SST-2 for Pruning Attention Layer

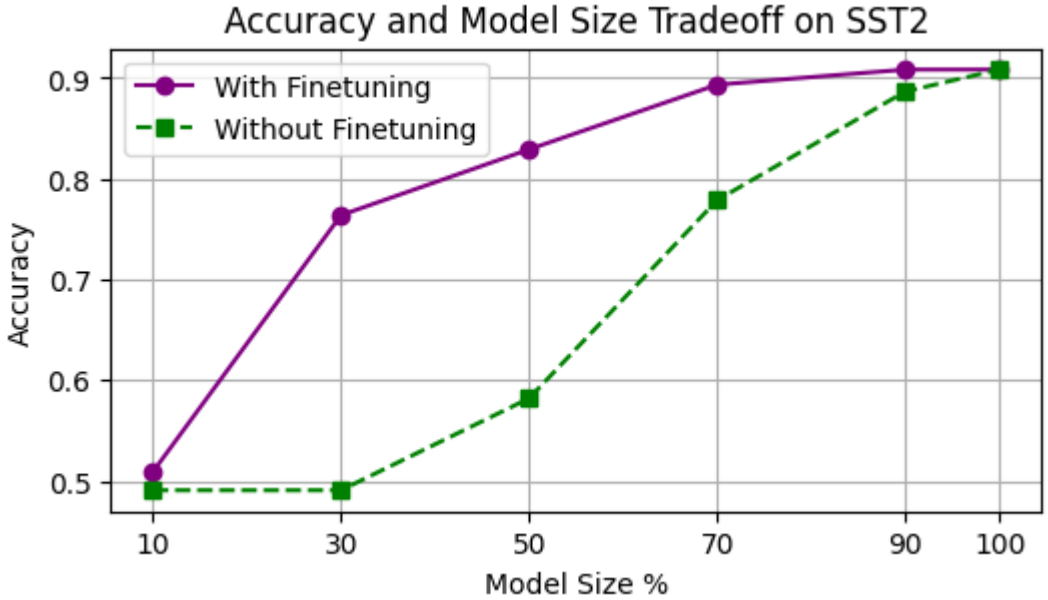
- Density = Pruned model size/Full model size
- *Pruning the Self-Attention layer -- the 4 matrices*

Density	Accuracy w/o finetuning	Accuracy with finetuning
1	-	0.9083
0.9	0.9151	0.9128
0.7	0.8601	0.9002
0.5	0.797	0.8475
0.3	0.5952	0.7947

Results on SST-2

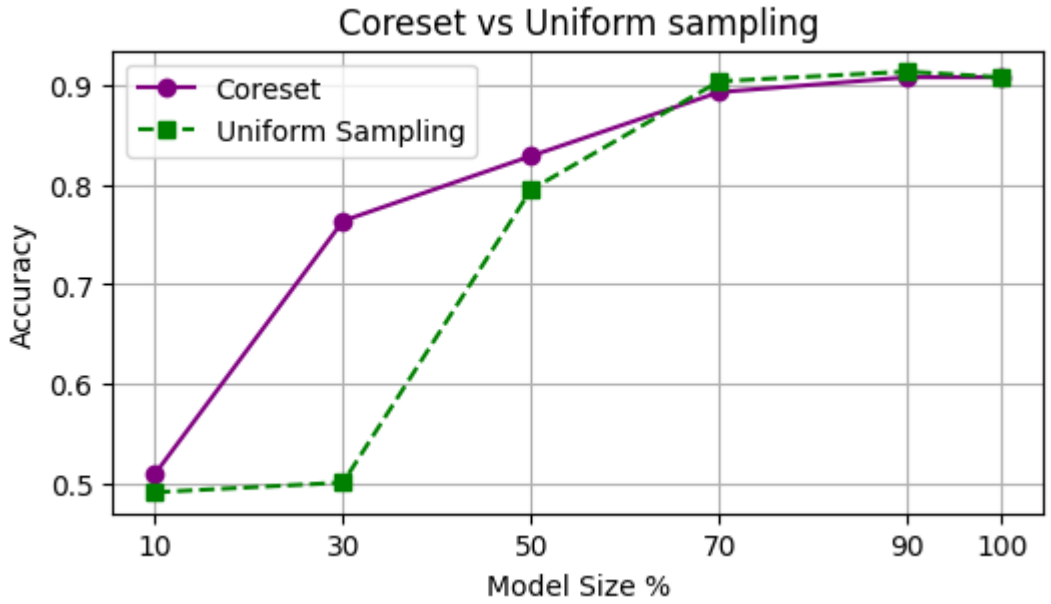
- Pruning the full model

Density	Inference time
1	940.13ms
0.9	916.69ms
0.7	794.17ms
0.5	633.06ms
0.3	519.61ms
0.1	397.20ms



Results on SST-2

- Coreset based pruning performs better at higher compression rates



Conclusion

- Coreset based importance pruning of LMs performs better at high compression rates.
- Pruning structured units in self-attention can give inference speedup.

Thank you!