What Do Transformers Learn in NLP? Recent Insights from Model Analysis

Rico Sennrich

joint work with Elena Voita, Ivan Titov, David Talbot, Fedor Moiseev
Recent Developments in NLP Leaderboard Race

3.2 Attention

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

3.2.1 Scaled Dot-Product Attention

We call our particular attention “Scaled Dot-Product Attention” (Figure 2). The input consists of queries and keys of dimension $d_k$, and values of dimension $d_v$. We compute the dot products of the query with all keys, divide each by $\sqrt{d_k}$, and apply a softmax function to obtain the weights on the values.

new neural architectures

pre-training becomes mainstream
how do neural architectures work?
(why) does pre-training objective matter?

- BERT-style masked language modeling better than causal language model
  [Lample and Conneau, 2019]
- “Language Modeling Teaches You More Syntax than Translation Does”
  [Zhang and Bowman, 2018]
- ...but multilingual NMT may allow better cross-lingual transfer than mBERT
  [Siddhant et al., 2019]
What Do Transformers Learn?

1. how do neural architectures work?

2. (why) does pre-training objective matter?
- multi-head self-attention is key Transformer component
- questions:
  - how to identify important attention heads?
  - can we prune unimportant ones?
  - which functions do attention heads have?
- spoiler (paper title): “Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned”
Determining Importance of Self-Attention Heads

- method 1: assume that “confident” heads (low entropy weight distribution) are important
- method 2: layerwise relevance propagation (LRP) [Bach et al., 2015, Ding et al., 2017]
- method 3: pruning: add regularization term to training objective which deactivates unimportant heads ($L_0$ norm on scalar gates drawn from Hard Concrete Distribution)
Determining Function of Self-Attention Heads

function of attention head is determined via simple rules
Determining Function of Self-Attention Heads

positional: maximum attention weight is given to specific relative position

![Diagram](image)
Determining Function of Self-Attention Heads

syntactic: maximum attention weight is given to token in specific dependency relation

![Attention Weights Diagram](image)
Determining Function of Self-Attention Heads

rare tokens: maximum attention weight is given to least frequent token
Important Self-Attention Heads Are Specialized

important heads tend to be **positional**, **syntactic**, or focus on **rare tokens**.
Pruning Self-Attention Heads

most heads (in encoder) can be pruned with little quality loss

most heads that survive pruning have one of the functions we identified
1. How do neural architectures work?

2. (Why) does pre-training objective matter?
The Evolution of Representations in the Transformer

[Voita, Sennrich, Titov, EMNLP 2019]

compare representations of models only differing in objective function:

- same architecture (Transformer encoder)
- same (source-side) training data (WMT EN→{DE,FR})

background: information bottleneck principle

[Tishby et al., 1999, Tishby and Zaslavsky, 2015]

hypothesis: deep neural model learns to compress input representation, retaining information necessary to:

- predict the output label
- build representations of other tokens
Same Architecture, Different Objective Functions

- **Language Model (causal, LM)**
  - saw a cat<eos>
  - I saw a cat

- **Masked Language Model (MLM, aka BERT)**
  - I saw a cat
  - I[MASK] a cat

- **Machine Translation (MT)**
  - Я видел кису
  - I saw a cat

Rico Sennrich
Is (Input) Token Identity Preserved?

How can this be non-monotonic?

- We measure MI/acc per position
- In MLM, information about token is distributed across sentence
Is Token Identity Preserved?

representations of *is, are, were, was*  
(t-sne projection)

- MT
- LM
- MLM

→ layers
distance between position of a representation and its k-nearest neighbors.
lower layers in (causal) LM represent input (left token)
higher layers form representations predictive of output (right token)
Conclusions

- analysis of pruning of self-attention heads could lead to:
  - model interpretability
  - efficiency
- learning objective affects information flow in Transformer
- analysis of representations complements probing experiments
  → can be used to explain why:
  - some pre-training objectives are more successful
  - lower layers may perform better in some probing tasks than higher ones
Thank you for your attention

more content in blog posts and papers!

https://lena-voita.github.io/posts/acl19_heads.html
https://lena-voita.github.io/posts/emnlp19_evolution.html

t-sne clustering of CCG tags


High Importance of Rare Tokens: Overfitting?

[Voita et al. ACL 2019]: some heads specialize on rare tokens.

[Voita et al. EMNLP 2019]: rare tokens highly influential but effect goes away after randomly swapping 10% of tokens.