

Improving Efficiency and Robustness of Transformer-based Information Retrieval Systems



Tutorial presented by Edmon Begoli, PhD, Sudarshan Srinivasan,
PhD and Maria Mahbub

Outline

Transformers for Information Retrieval

Transformer Architecture

Break

Optimization and Efficiency Improvement Techniques

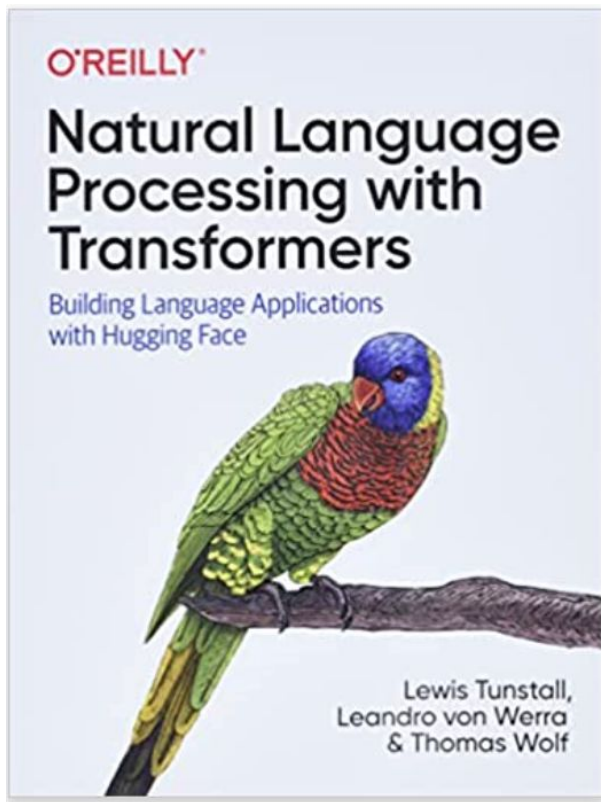
Robustness


Break

Q&A Session

Background

Credits and Recommendations



 **Hugging Face**

[Models](#) [Datasets](#) [Spaces](#) [Docs](#)

Tasks

[Image Classification](#) [Translation](#)
[Image Segmentation](#) [Fill-Mask](#)
[Automatic Speech Recognition](#)
[Token Classification](#) [Sentence Similarity](#)
[Audio Classification](#) [Question Answering](#)
[Summarization](#) [Zero-Shot Classification](#)

+ 16 Tasks

Libraries

[PyTorch](#) [TensorFlow](#) [JAX](#) + 25


Datasets


[common_voice](#) [wikipedia](#) [squad](#)
[glue](#) [bookcorpus](#) [c4](#) [conll2003](#)
[emotion](#) + 1031


Languages


[en](#) [es](#) [fr](#) [de](#) [zh](#) [ja](#) [ru](#) [sv](#) + 178


Models 51,369


bert-base-uncased
 Fill-Mask • Updated 6 days ago • ↓ 14.9M • ♥ 162


distilgpt2
 Text Generation • Updated 11 days ago • ↓ 14.7M • ♥ 67

gpt2
 Text Generation • Updated May 19, 2021 • ↓ 12.3M • ♥ 120

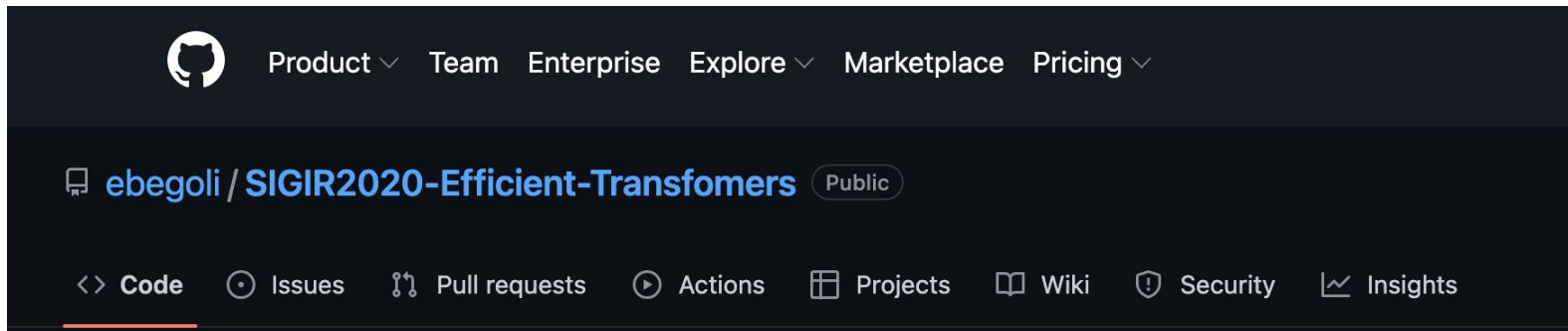
distilbert-base-uncased-finetuned-sst-2-english
 Text Classification • Updated Mar 22 • ↓ 10.2M • ♥ 58

roberta-base
 Fill-Mask • Updated Jul 6, 2021 • ↓ 8.06M • ♥ 39

distilbert-base-uncased
 Fill-Mask • Updated 12 days ago • ↓ 7.32M • ♥ 63

 **hfl/chinese-roberta-wwm-ext**

Source Material



Transformers for Information Retrieval (IR)

An overview of the uses of
transformer-based deep neural
architectures in information retrieval
(IR)

List common tasks

Some resources to be included

SBERT:

<https://www.sbert.net/examples/applications/information-retrieval/README.html>

<https://jalamar.github.io/illustrated-retrieval-transformer/>

<https://blog.vespa.ai/pretrained-transformer-language-models-for-search-part-1/>

Transformer Architecture

Motivation for Transformers

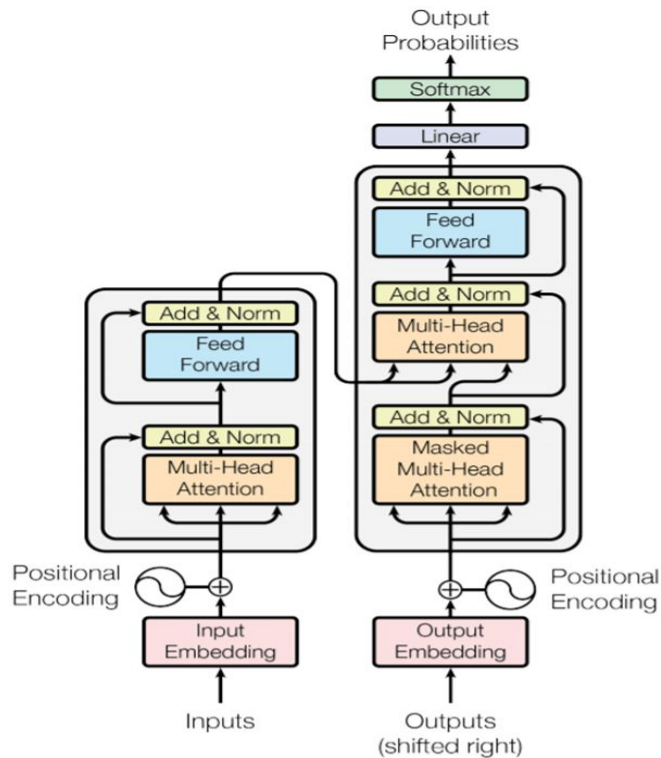
An evolution in distributional semantics-based NLP

Performance on NLP tasks

Parallelization

Benefits of Deep Learning

Transformer Architecture



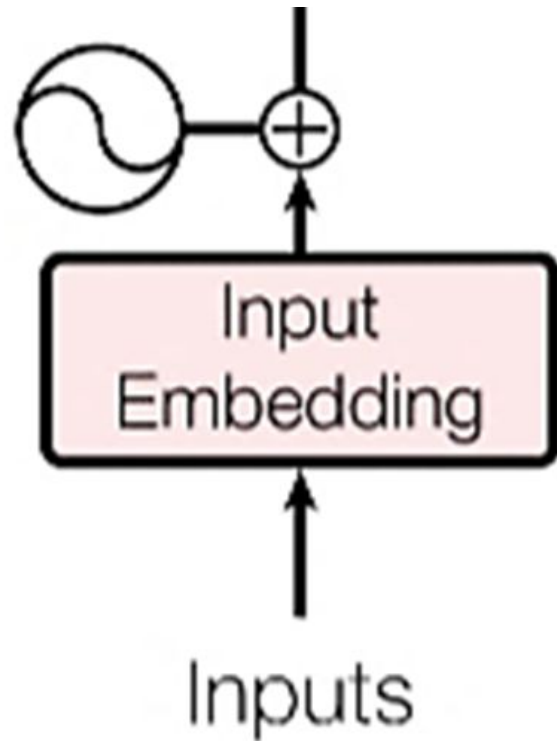
Positional Encoding

- Embeddings do not carry any information about the relative order of the tokens in the sentence
- Requires a way to encode information about a token's position in a sentence
- “I **google** for information. Thanks **Google**.”
- Variables:
 - pos : Position of the token *within* the input sequence
 - i : Position of the *embedding dimension* within the vector representation of the token

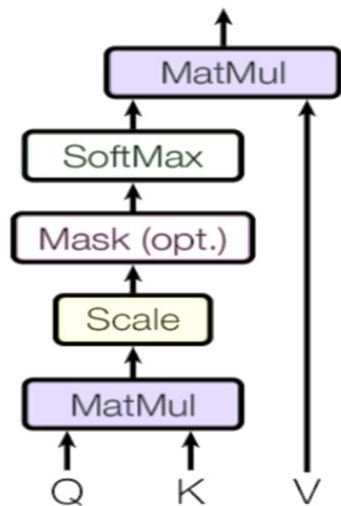
$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$

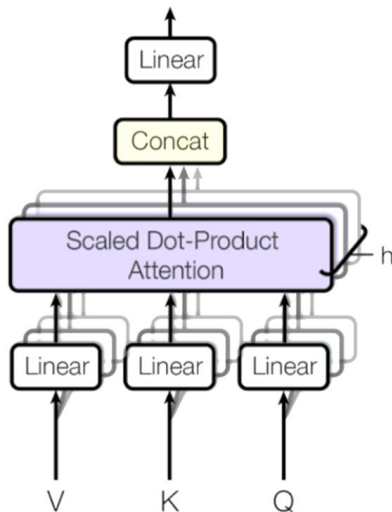
Positional
Encoding



Attention for Transformers



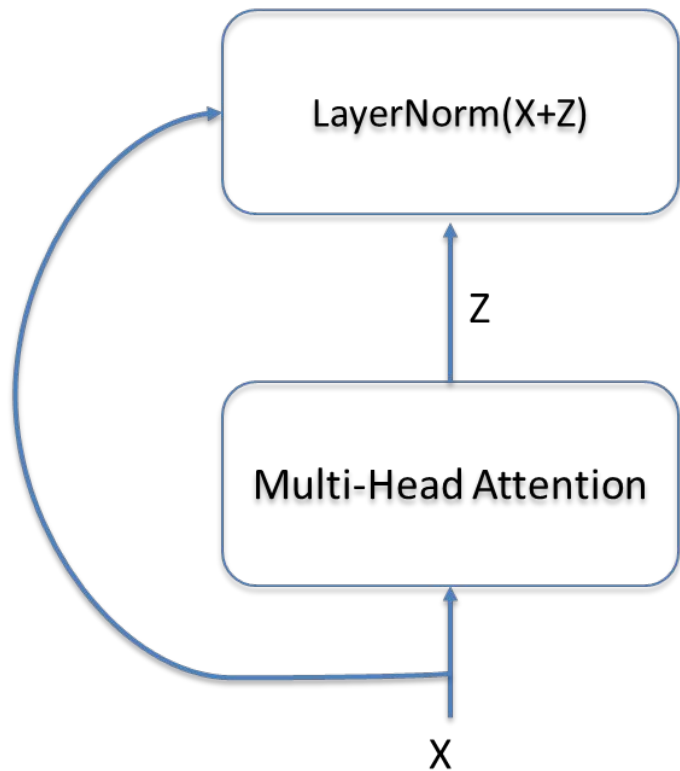
- Attention can be described as mapping a query and set of key-value pairs to an output
- Tells the model which part of the input it should focus on



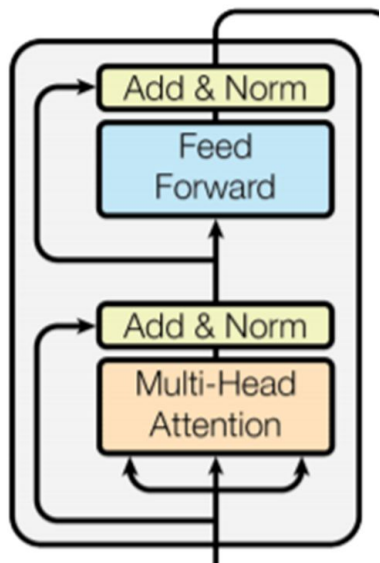
- Consists of h parallel self-attention layers, each one is called a *head*
- Multi-head attention allows the model to jointly attend to information from different representation subspaces at different positions

Add & Norm

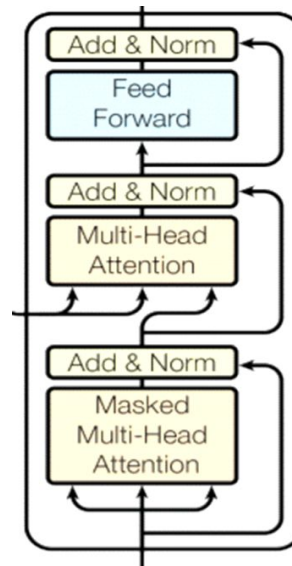
- Residual (skip) connection to allow gradients to flow through the network directly
- Layer normalization to “tame” the gradients to the mean and standard deviation of each layer
- Each layer output is shifted and scaled by their collective mean and standard deviation respectively to normalize that layer



Encoder-Decoder Architecture



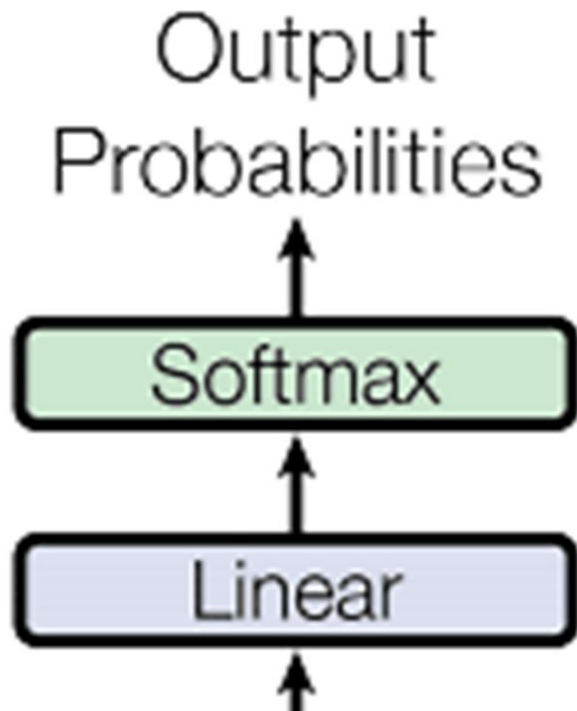
- Encoder block contains two “sub-layers”
- There can be many encoder blocks stacked together in an encoder layer



- Contains an extra layer of multi-head attention and Add & Norm
- Keys and values for the additional multi-head attention comes from the encoder which carries the states of the input words
- Input is the generated output of the translated word

Final Layer

- Final linear takes in the output from the last decoder block of the decoder stack
- This is fed into a softmax layer to get output probabilities of each word in the vocabulary
- The word with the highest probability is selected as the translated word and is fed into the next position
- Process ends when an end-of-sentence (EOS) token is generated



Bi-directional Encoder Representations from Transformers (BERT)

Break

Performance and Efficiency

Practice and Examples

Improvements to Robustness

Break

Discussion

Draft material to be used in the slides

1. http://ceur-ws.org/Vol-2621/CIRCLE20_05.pdf - “Given the recent success of transformer encoders for NLP, we propose KTRel: a NLTR model that uses word embeddings, Knowledge bases and Transformer encoders for IR.” --- **uses transformers for simple IR**
2. <https://dl.acm.org/doi/pdf/10.1145/3331184.3331303> - “. This paper studies leveraging a recently-proposed contextual neural language model, BERT, to provide deeper text understanding for IR. Experimental results demonstrate that the contextual text representations from BERT are more effective than traditional word embeddings.” --- **studies the importance of transformers in IR**
3. http://ceur-ws.org/Vol-2696/paper_241.pdf - “In particular, we investigate three approaches: (1) query expansion using GPT-2, (2) query expansion using BERT, and orthogonal to these approaches, (3) embedding of documents using Google’s BERT-like universal sentence encoder (USE) combined with a subsequent retrieval step based on a nearest-neighbor search in the embedding space.” --- **uses transformers to enrich argument retrieval at various stages of the IR pipeline**
4. https://dl.acm.org/doi/pdf/10.1145/3409256.3409829?casa_token=17_ORNp2RoAAAAA:Niw0NgI4VS-nB0AxRiDHH_QY8AhfiZFJe1iqZW8cWLUJkR9v8LdACXhDkHS98XBar6QJWYedA_9N - “propose a framework called PyTerrier that allows advanced retrieval pipelines to be expressed, and evaluated, in a declarative manner close to their conceptual design” --- **proposes an IR tool (like PyTorch) that uses Transformers**
5. https://link.springer.com/chapter/10.1007/978-3-030-99736-6_27 - “Motivated by those observations we aim to answer the following question: how robust are retrieval pipelines with respect to different variations in queries that do not change the queries’ semantics?” --- **analyzes robustness of transformer-based IR system**
6. https://dl.acm.org/doi/pdf/10.1145/3459637.3482452?casa_token=LQRHKWATm5wAAAAA:2_G_sJv8x-Ugzy0X1mVOMM91G_LNDOqj-7eZwyQwvWmlQ-Y4Py0ViuxFv6tugbhN9LXpGoKrU0Wd - “.. in the multilingual pre-trained models that the words in different languages are projected into the same hyperspace, the model tends to “translate” query terms into related terms – i.e., terms that appear in a similar context – in addition to or sometimes rather than synonyms in the target language. This property is creating difficulties for the model to connect terms that co-occur in both query and document. To address this issue, we propose a novel Mixed Attention Transformer (MAT) that incorporates external word-level knowledge, such as a dictionary or translation table.” --- **uses transformers in IR in a multilingual setting**