Introduction to Search Relevance Ranking- Session III – Knowledge Distillation

Tutorial Link: https://dlranking.github.io/dlrr/

Data source: https://huggingface.co/datasets/xglue

XGLUE: A New Benchmark Dataset for Cross-lingual Pre-training, Understanding and Generation)

Presenters: Xue Li, Keng-hao Chang @ Microsoft Ads

Date: August 14th, 2022

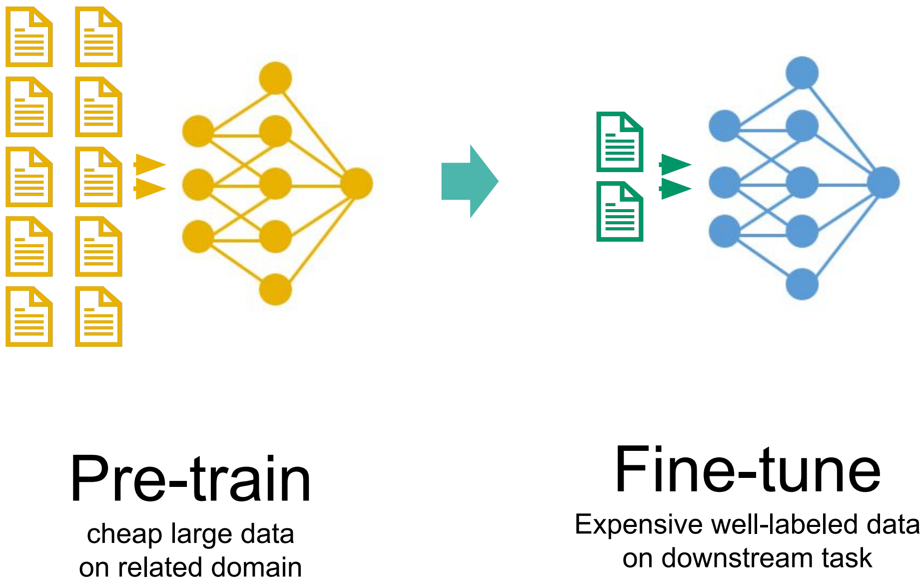
Agenda

- Knowledge distillation
- Case study in DistilBert
- Case study in Microsoft Ads for Ranking
- The colab

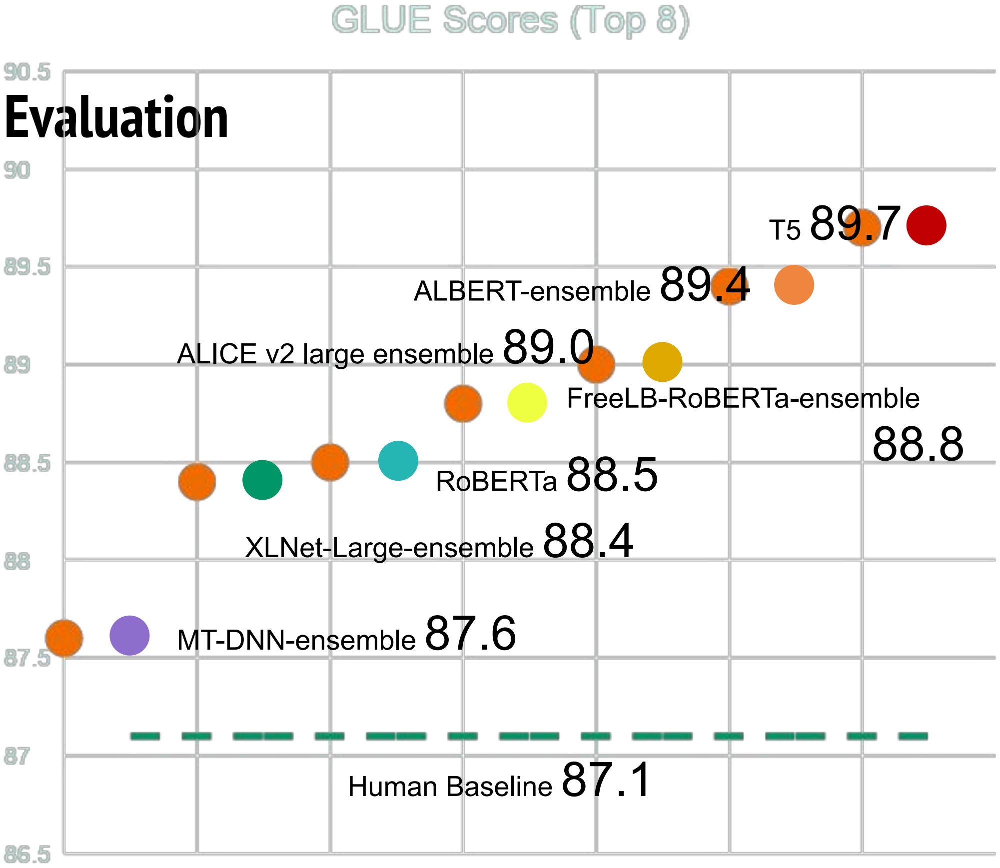
Pre-train and fine-tune

- Natural Language Processing
- Pre-train on unsupervised tasks, e.g., Language Modeling
- Fine-tune on downstream NLP tasks, e.g., Question Answering, search relevance

- Large & powerful NLP models, even beat human!

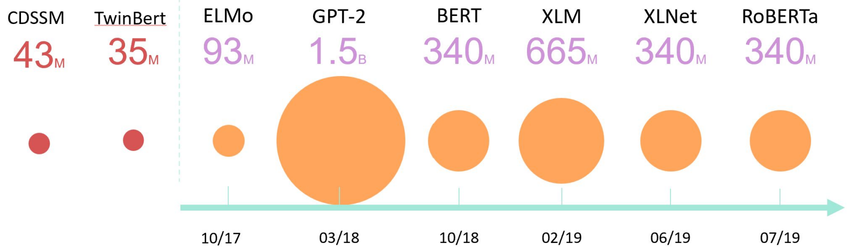


GLUE –
General Language Understanding Evaluation



https://gluebenchmark.com/leaderboard/, 2019-11-12

How large are pre-trained NLP models? (and distilled)



Learning semantic representations using convolutional neural networks for web search | Proceedings of the 23rd International Conference on World Wide Web (acm.org) (CDSSM)
TwinBERT | Proceedings of the 29th ACM International Conference on Information & Knowledge Management

Call outs

Will cover

- Knowledge distillation
- Practices of knowledge distillation

Will not cover

- Other lighter BERT model techniques (Albert, Electra)
- All research directions of knowledge distillation

[2006.05525] Knowledge Distillation: A Survey (arxiv.org)

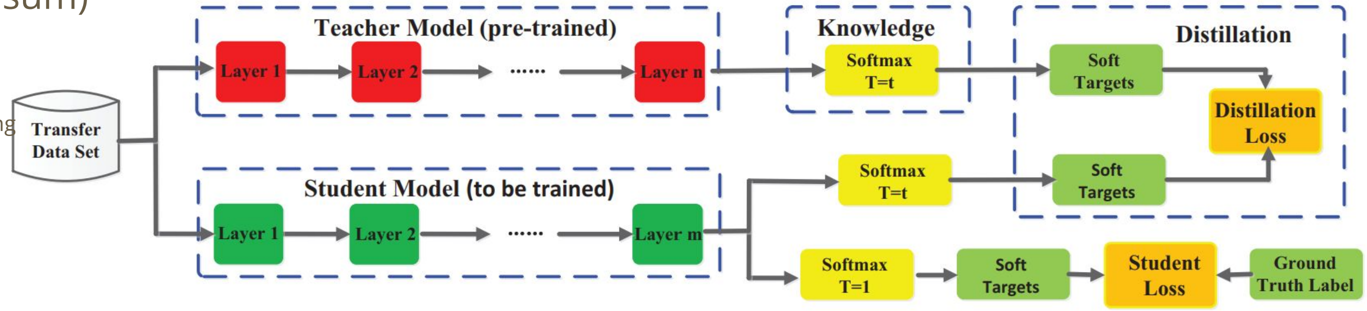
Response level knowledge distillation

- Distil by learning softmax from teacher on a transfer set
  - i.e., soft label, dark knowledge*
  - KL divergence
  - Vs. Logits
- Two losses (weighted sum)
- Temperature
  - Analogous to label smoothing

$$L_{ResD}(z_t, z_s) = \mathcal{L}_R(z_t, z_s)\ , \tag{1}$$

$$p(z_i, T) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}\ , \tag{2}$$

$$L_{ResD}(p(z_t, T), p(z_s, T)) = \mathcal{L}_R(p(z_t, T), p(z_s, T))\ . \tag{3}$$



[1503.02531] Distilling the Knowledge in a Neural Network (arxiv.org), Hinton

*BERT-base’s predictions for a masked token in "I think this is the beginning of a beautiful [MASK]" comprise two high probability tokens (day and life) and a long tail of valid predictions (future, story, world. . .).

Feature-based knowledge distillation

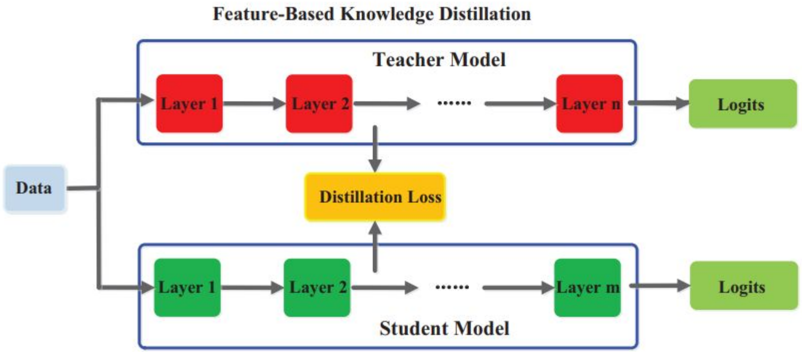
- Learning feature maps of the intermediate layers from teacher to student models
  - L2-norm distance, L1-norm distance, cosine loss etc.
  - Due to the significant differences between sizes of hint and guided layers, how to properly match feature representations of teacher and student also needs to be explored.

$$L_{FeaD}\big(f_t(x), f_s(x)\big) = \mathcal{L}_F\big(\Phi_t(f_t(x)), \Phi_s(f_s(x))\big) \ , \qquad (4)$$

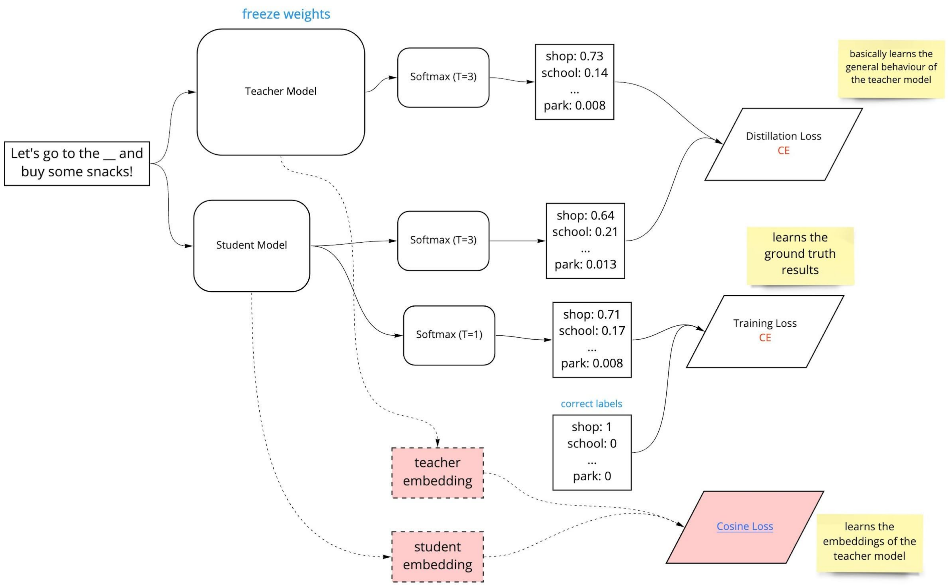
$f_t(x)$ and $f_s(x)$ are the feature maps

The transformation functions, $\Phi_t(f_t(x))$

$\Phi_s(f_s(x))$,



Case study: distillBert



- 3 losses (for both response & feature)
  - Distillation loss
  - Training loss
  - Cosine loss

$$\mathcal{L}_{cos} = 1 - cos(h_T, h_S)$$

- Architecture
  - the number of layers is reduced by a factor of 2.
  - Initialize by teacher every other layer
- DistillBERT model retains almost 97% of the original BERT-base model's language understanding when evaluated on GLUE benchmarks. In addition to this, it is 40% smaller and 60% faster at inference.
- General-purpose pre-training distillation rather than a task-specific distillation

[[1910.01108] DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter (arxiv.org)]

Case study: distillBert & distillation variant from Internal Representations

(1) KL divergence loss across the self attention probabilities of all the transformer heads
(2) the cosine similarity loss between the [CLS] activation vectors for the given layers

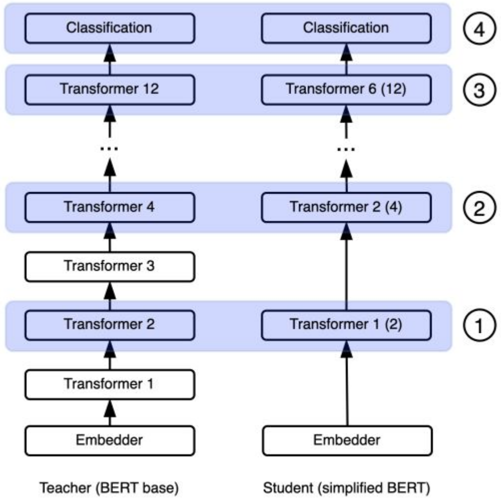


Figure 1: Knowledge distillation from internal representations. We show the internal layers that the teacher (left) distills into the student (right).

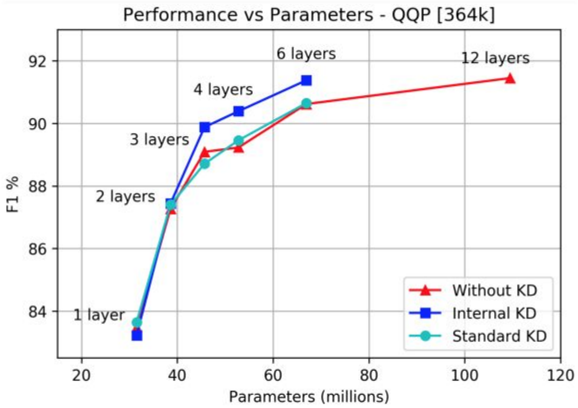


Figure 2: Performance vs. parameters trade-off. The points along the lines denote the number of layers used in BERT, which is reflected by the number of parameters in the x-axis.

[1910.03723] Knowledge Distillation from Internal Representations (arxiv.org)

Variants of Knowledge Distillation

- Offline distillation
  - Two steps: Pretrain teacher then distill student; one way; capacity gap
- Online distillation
  - Both teacher and student are updated simultaneously
  - E.g., multi-branch architecture, in which each branch indicates a student model and different branches share the same backbone network.
  - E.g., Any one network can be the student model and other models can be the teacher during the training process.
- Self distillation
  - the same networks are used for the teacher and the student model

- Teacher-Student Architecture

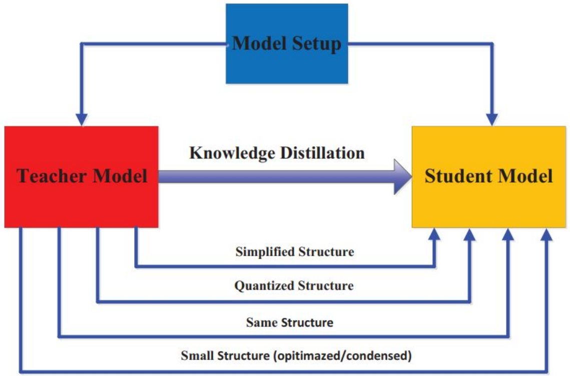
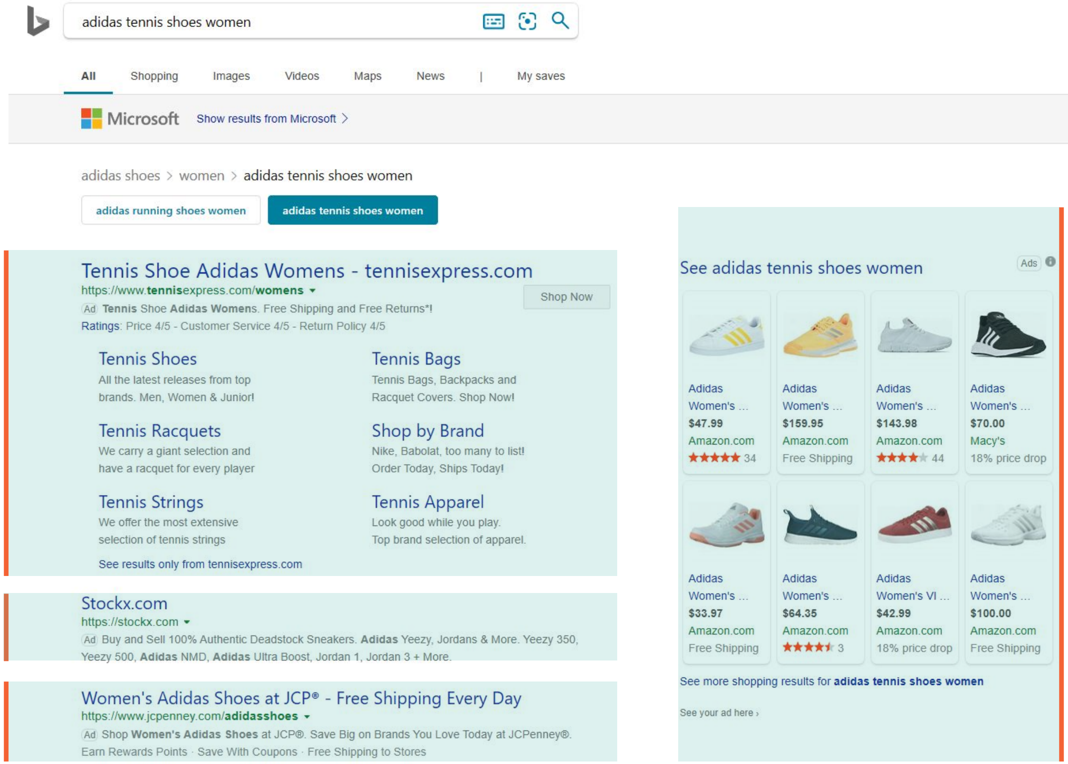


Fig. 9 Relationship of the teacher and student models.

- Adversarial Distillation
- Multi-Teacher Distillation
- Data-Free Distillation

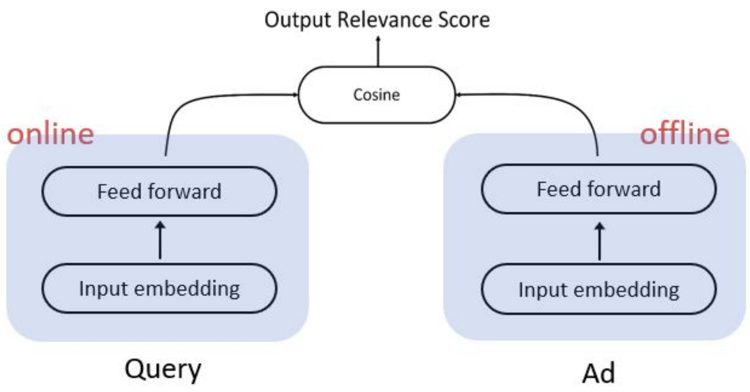
Case study: search relevance ranking at Microsoft Ads

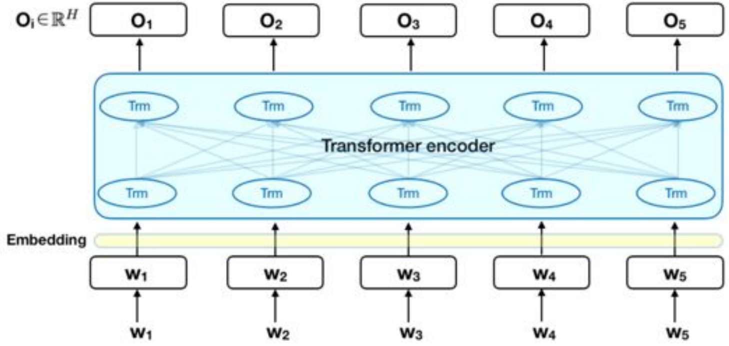
- Point-wise relevance score by against human labels
- Used as externality in ranking
  - pDefect = 1-Relevance
  - RankScore = Bid * pClick – w*pDefect



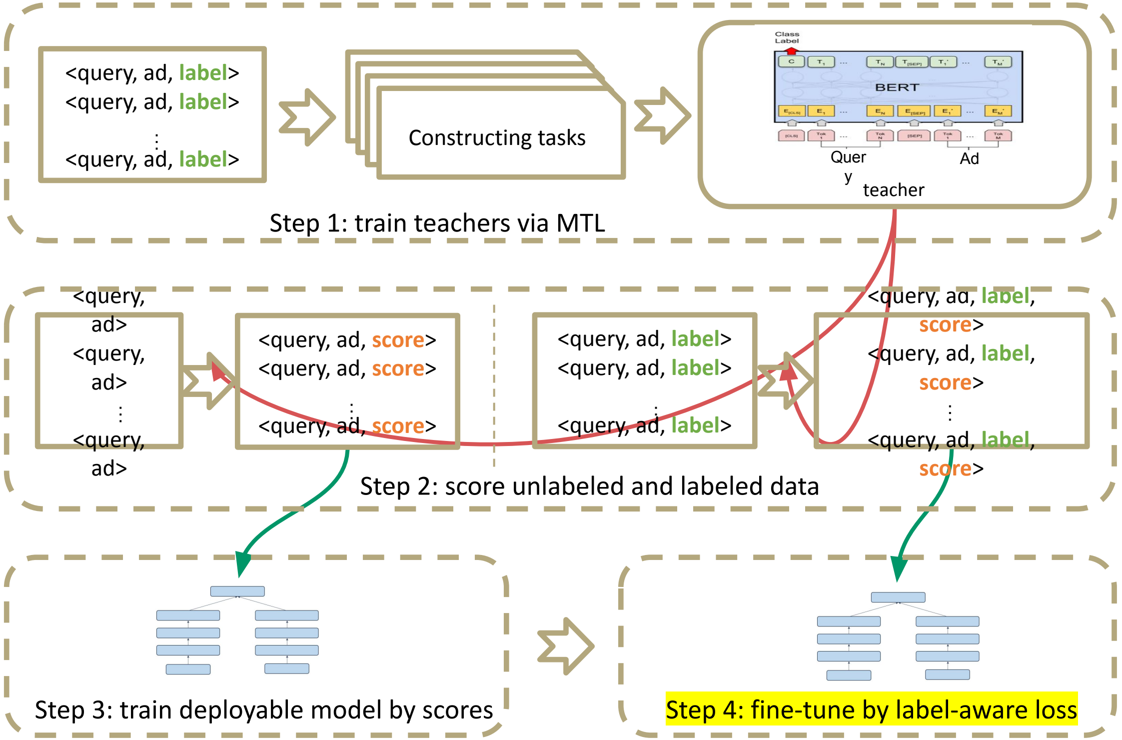
Choice of student model

- Two tower
    - CDSSM, TwinBert
    - Doc embedding is offline computed
- BERT-like
    - cannot support fast compute, latency prohibitive.





Knowledge distillation for search Relevance



Teacher Training

- Narrow the gap between pre-trained models and target tasks

Inference Data

- Score both labeled / unlabeled data

- Train a deployable student model
- Using scored unlabeled data

Student Fine-tuning

- Fine-tune student model
- Using scored labeled data

Learning Fast Matching Models from Weak Annotations, WWW'19

Recipe of AdsBERT Distillation

Initialization

Pre-trained BERT
340M params

Pretrain

MLM/NSP
400M Ads data

MTL Finetune

8 ad tasks
40M samples

Inference

Vast amount
Proper distribution

Distillation

CDSSM keep **70%**
AUC gain

Case study: TwinBert

- Two tower Bert
- Pooling & crossing layer

Table 2: ROC-AUC of TwinBERT models comparing with C-DSSM, $BERT_3$ and $BERT_{12}$ on two test sets

Model	AUC_1	AUC_2
C-DSSM	0.8713	0.8571
$BERT_3$	0.8995	0.9107
$TwinBERT_{cos}$	0.8883	0.8743
$TwinBERT_{res}$	0.9010	0.9113
$BERT_{12}$	0.9011	0.9137

Table 3: Density differences of all 4 labels by comparing top 5 results from $TwinBERT_{cos}$ and C-DSSM

bad	fair	good	excellent
-7.4%	-2.6%	1.9%	18.8%

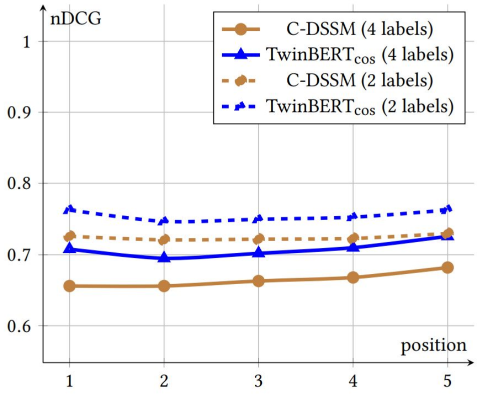


Figure 3: nDCGs of $TwinBERT_{cos}$ and C-DSSM

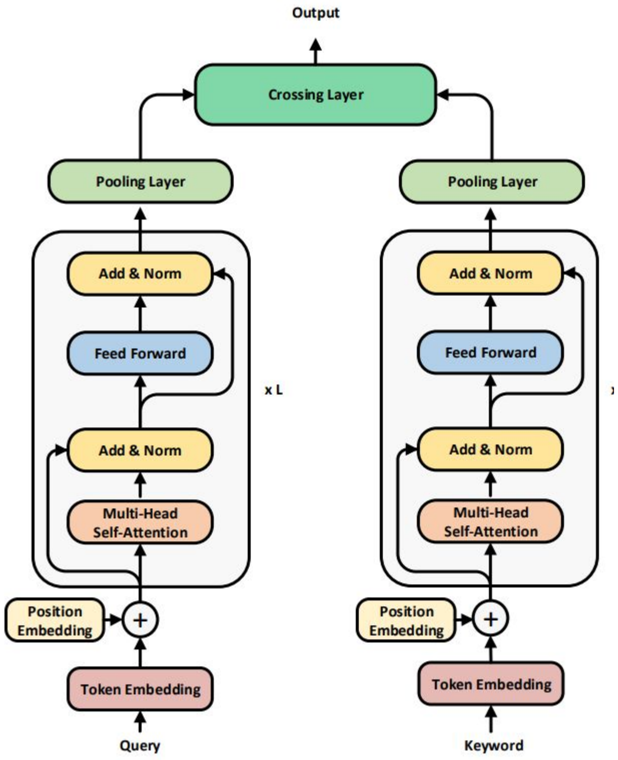


Figure 1: TwinBERT Architecture

TwinBERT | Proceedings of the 29th ACM International Conference on Information & Knowledge Management

Colab

- QADSM task in xGLUE dataset, which is extracted from real Bing Ads traffic.