Curriculum Learning & Selective Reliance in Knowledge Distillation for Language Models

Hasan Khan
New York University
hk3550@nyu.edu

Abstract

Knowledge distillation (KD) is a powerful, well-established model compression technique that can face performance limitations when the capacity difference between student and teacher models is severely mismatched (Cho and Hariharan, 2019), or when multiple teachers cause competing distillation objectives (Du et al., 2020). To address these issues and improve performance in KD for large language modeling, I explore the implementation of two ideas: using curriculum learning during KD, where training data is sorted based on difficulty, and using Selective Reliance during KD, where a student language model selectively leverages teacher distillation loss for data samples deemed difficult by the curriculum.

1 Introduction

Knowledge Distillation (Hinton et al., 2015) is a model compression technique commonly deployed in settings where large models are difficult to store and run. Vanilla KD comprises of a dual model student-teacher modeling framework, where a small capacity student model aims to mimic the performance of a larger capacity teacher model by learning the distribution of the output labels generated from the teacher model trained on the same dataset. Specifically, the student model uses a bipartite loss function $L_{student}$ that incorporates both L_{kd} (the KD loss measured by the KL divergence between the softmax of the student output logits P_S and the softmax of the teacher output logits P_T, scaled by the temperature parameter τ) and L_{ce} (the standard cross entropy training loss using the true labels y_{true}). The parameter λ controls the weight given to each component loss. The student loss and its component losses are defined below:

$$L_{ce} = CE(y_{true}, P_S)$$

$$L_{kd} = \tau^2 KL(P_T, P_S)$$

$$L_{student} = (1 - \lambda)L_{ce} + \lambda L_{kd}$$

The general framework is visualized in Figure 1.

However, KD has been shown to provide minimal to no performance gains for certain tasks, such as image recognition on ImageNet (Zagoruyko and Komodakis, 2016). Cho and Hariharan note that empirical results establishing the generality of KD efficacy for various tasks are nonexistent (Cho and Hariharan, 2019). They examine the reasons behind these failures, noting that large differences in student teacher model capacities may limit the student model's ability to minimize both training loss and KD loss, forcing the student to minimize the KD loss over the train loss. In addition to capacity gaps between student and teachers, KD frameworks that involve distillation from an ensemble of teachers require n-paritite loss functions, in which conflict can exists between different teacher models, which can adversely effect distillation loss (Du et al., 2020).

From these findings, I propose *selective reliance*, a KD technique for dynamically changing the student's reliance on KD loss, in order to improve model accuracy. Selective reliance is implemented by updating λ in $L_{Student}$ at the sample level, where λ is determined by the difficulty of the sample. Difficulty is defined in the context of Curriculum Learning (CL), a training strategy for improving model convergence speed and accuracy that involves training on easily learnable samples before more difficult ones (Bengio et al., 2009).

Difficulty rankings for samples can be determined using the confidence scores generated by either the teacher model (teacher-generated curriculum) or the student model (student-generated curriculum). In the former, the teacher model would also act as the difficulty scoring function (Hacohen and Weinshall, 2019) for each datapoint in the the

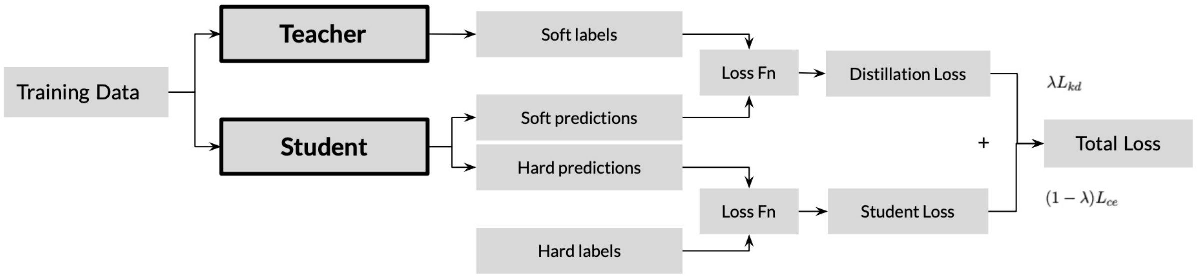

Figure 1: Knowledge Distillation: General Framework

student's curriculum, calculated before the student training process. In the latter, the student model would generate its own curriculum during student training by looking at confidence scores generated during previous epochs in training, as described by snapshot learning (Zhao et al., 2021). For this experiment, I use the confidence scores generated by the student model (student-generated curriculum) during training. I hypothesize that relying on the teacher model is only in the student's best interest when the sample being evaluated is "difficult" as determined by the curriculum, and that accuracy yielded from vanilla knowledge distillation (KD) and knowledge distillation with curriculum learning (KD-CL) can be improved upon with selective reliance (KD-CL-SR) techniques built into the distillation framework.

2 Related Works

2.1 Efficacy of Knowledge Distillation

Cho and Hariharan test the notion of generality in Knowledge Distillation, which states that students of *any* size or architecture can learn from teachers of *any* size or architecture (Cho and Hariharan, 2019).They experiment with various student and teacher network architectures on CIFAR and ImageNet datasets, and conclude that while larger teacher models have increased accuracy, student models trained on larger teachers may become less accurate due to large capacity gaps between student and teacher models. Larger models do not make inherently better teachers, and matching student and teacher capacities must be taken into consideration for proper knowledge distillation. In exploring ways around this capacity gap, they find that early stopping of teacher model training can help mitigate the effect of mismatched capacities.

2.2 Curriculum Learning for Deep Networks

Haconen and Weinshall analyze the effect of curriculum learning on deep vision network training by examining the challenges of sorting datapoint difficulties and computing mini-batches such that they exhibited increasing levels of difficulty (Hacohen and Weinshall, 2019). They sort the difficulty of training examples by using confidence scores for each training image, computed using two methods. The *transfer scoring function* method consists of using the confidence scores from an external classifier trained on feature vectors from the penultimate layer activations of a pre-trained ImageNet inception network to assign the difficulty of the data. The second method consists of training the deep network with the help of uniformly sampled mini-batches, then defining a scoring function by computing the confidence score for each image, also called the *self-taught scoring function*. They found that both methods, combined with pacing functions meant for mini-batch computation, had similar performance.

2.3 Knowledge Distillation Via Instance Level Sequence Learning

Zhao et al. propose a knowledge distillation framework in the image classification setting where the student constructs curriculums throughout training by utilizing "snapshots" of itself at early training checkpoints to rank image difficulty (Zhao et al., 2021). These difficulty scores then allow for on the fly curriculum generation during KD, yielding slightly improved accuracies and faster convergence time for the student model.

3 Method

The overall approach of this work is to determine whether KD, and specifically KD-CL and KD-CL-SR, can improve the accuracy of a small model on the biomedical question answering task,

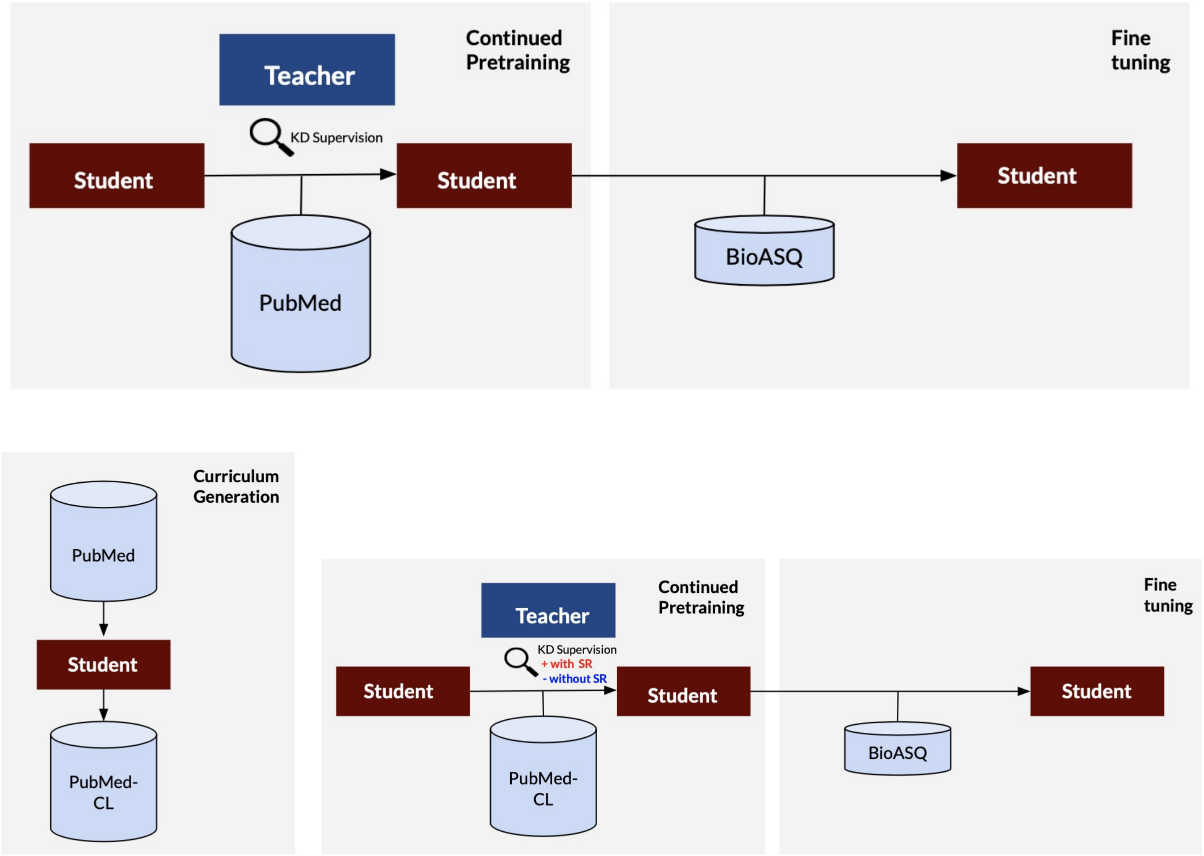

Figure 2: Experimental Setup: Training regime for three experiments, with steps progressing sequentially from left to right. The top figure depicts the training regime for vanilla KD, where a student is further pre-trained on PubMed abstracts under supervision by the BioBert teacher, and then fine tuned on BioASQ data for evaluation. The bottom figure depicts the training regime for KD-CL and KD-CL-SR. A curriculum is first generated on the PubMed abstracts by passing them through the student model at inference, yielding PubMed-CL. Then, the student is further pre-trained on that ordered curriculum, and finally fine tuned on BioASQ data for evaluation as before. Selective reliance is applied during the continued pre-training for KD-CL-SR.

BioASQ. As a benchmark for comparison, two baseline models are fine tuned on the BioASQ task; a dmoain naive, unguided DistillBert, which achieves a fairly low accuracy (and serves as the base architecture for student models in remaining experiments), and the domain adapted BioBert model, which achieves a relatively high accuracy (and serves as the base architecture for teacher models in remaining experiments). I hypothesise that the scores of these baseline models will serve as loose lower and upper bounds for the three knowledge distillation experiments (KD, KD-CL, and KD-CL-SR), which should fall between these bounds; where KD is expected to improve upon the unguided DistillBert, KD-CL over KD, KD-CL-SR over KD-CL, and that all student models will remain less accurate than BioBert, the designated expert.

All experiments were run on Google Collab GPUs. See Appendix A for data details, B for task specific settings, C for evaluation metrics, and D for experiment hyperparameter configurations. All source code for experiments can found in this Github repository.

3.1 Knowledge Distillation Framework

To test the effects of curriculum learning and selective reliance on knowledge distillation in language models, I conduct additional pre-training (Gururangan et al., 2020) on the small DistillBert student, with distillation supervision by the larger BioBert teacher model, using specialized biomedical data from PubMed abstracts (Lee et al., 2019). Standard masked language modeling (MLM) is the task chosen for additional pretraining. Depending on the experiment, the PubMed abstract corpus may be ordered into a curriculum (KD-CL, KD-CL-SR), and selective reliance may be applied (KD-CL-SR). Once further pretrained, the student model is fine tuned and evaluated on the BioASQ question answering task selected from BLURB (Gu et al., 2021), a broad LM benchmark for general Biomedical language understanding, and compared to scores attained by the naive student and domain adapted teacher models. This KD setup adapts and builds on recent work implementing KD for biomedical models (Rohanian et al., 2022). The decision to conduct KD at the pre-training stage, rather than at the fine tuning stage, follows from research and applied work done showing higher efficacy of knowledge distillation during the pre-training phase w/ subsequent fine tuning, rather than conducting KD on just the fine tuning phase. (?). See Figure 2 for a visual of the KD-CL framework.

3.2 Student Curriculum Generation

As a preliminary step to running any curriculum learning experiments, a curriculum for the student models must be generated. To do this, the PubMed abstracts are fed through the student model, and inverse confidence scores are derived for each text sample by calculating the MLM loss via model inference for each abstract. Once these scores have been assigned to each abstract, abstract texts are sorted by increasing inverse confidence (increasing loss) into an easy-to-hard curriculum, with the hardest samples having performed the worst during MLM inference. See Figure 2 for a visual of the generated curriculum, PubMed-CL.

3.3 Student Training & Experiments

I conduct three training experiments on the DistillBert student models, with the student with no teacher (plain DistillBert) model serving as the student baseline. The remaining three experiments test the effects of using curriculum ordered training and selective reliance implementations, listed as follows: student w/ knowledge distillation (Student-KD), student w/ knowledge distillation using a student-generated curriculum (Student-KD-CL), student w/ selectively reliant knowledge distillation using a student-generated curriculum (Student-KD-CL-SR). See Figure 2 above for a chart depicting the experiments using curriculums and selective reliance.

3.4 Implementing Selective Reliance

Since the curriculum is - by definition - ordered on difficulty, selective reliance was to be implemented by scaling up λ from 0 through the duration of an epoch. One lambda value would be selected per batch. Values would be based on the average confidence score of the batch C_b, normalized by the maximum global confidence score C_g and shifted by a small scaling factor s (to set the initial λ to 0). I define lambda in the context of selective reliance as follows:

$$\lambda = \frac{(1 - C_b)}{max(C_g)} - s$$

4 Results

BLURB Model Scores	Test Acc
BioBert (large model)	84.14 %
PubMedBert (large model)	87.56%
DistillBert (small model)	74.44%
Baseline Model Scores	**Test Acc**
BioBert (large model)	68.12%
PubMedBert (large model)	60.04%
DistillBert (small model)	54.02%

Table 1: BLURB scores for the BioASQ task as reported by (Gu et al., 2021) (top). My baseline scores for the same BioASQ task (bottom).

The table above shows the baseline scores on the BioASQ task for various BERT models. These scores do not correspond exactly [1] with scores achieved in (Gu et al., 2021), but maintain the same general pattern of accuracy, with the smaller Bert model achieving the lowest accuracy, followed by the larger domain adapted models. Using the above results as a baseline, the experimental results from the knowledge distillation setting can be evaluated (see Table 2).

From Table 2, it is observed that applying Knowledge Distillation improves accuracy over the baseline score by roughly three percentage points. Using curriculum learning further improves over the KD experiment score, though the increase in accuracy is quite negligible. It is perhaps tenable that Curriculum Learning does not actively harm accuracy for this particular task. Regarding selective reliance, I was unable to run the experiments

[1] Most likely due to training environment differences; additionally, I was unable to find hyperparameter details for when these models were fine tuned on the BioASQ task

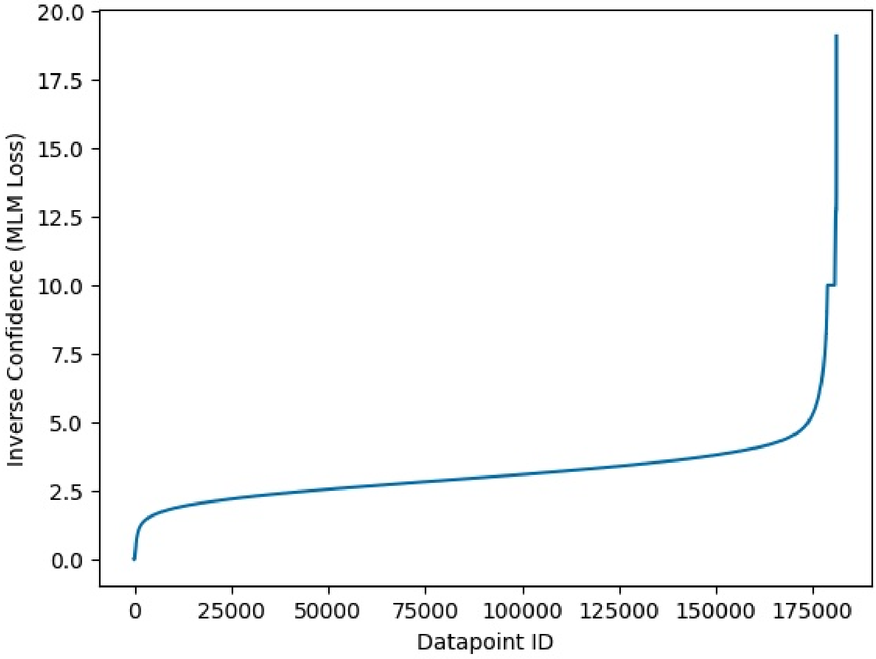

Figure 3: Sample Curriculum generated on a representative subset of 175K abstracts. Observe that a majority of abstracts are equally difficult by this metric, with two small populations of samples being deemed really easy (dip near ID 0, ≤2.5 loss) and really hard (spike near ID 175000, ≥ 5.0). The small plateau toward the tail of this graph is an artifact of NAN replacement strategy for "dirty" abstracts of inappropriate length or nonsensical text.

Experiment Model Scores	**Test Acc**
DistillBert-KD	57.03%
DistillBert-KD-CL	58.57%
DistillBert-KD-CL-SR	–

Table 2: Experiment Scores on the BioASQ Question Answering task. Here, the BioBert model is used as the teacher in Knowledge Distillation for convenience purposes, adapting previous open source code published by (Rohanian et al., 2022).

in due time, and leave them for future exploration.

5 Discussion

Ultimately, the limited set of results partially confirm my hypothesis that KD during MLM pretraining will improve the accuracy of a naive student model on a domain specific task, and that curriculum learning can be used to improve accuracy during KD. Still, the results are not that significant, especially for CL. There could be many reasons why KD did not significantly improve performance. Teacher and student models in this particular experiment may have too large of a size gap, one of the most fundamental problems in knowledge distillation. A proper hyperparameter sweep over various temperature and lambda values was not conducted, which would be necessary for a more rigorous analysis.

For more rigorous work to integrate CL into the KD framework, different types of curriculum generation methods could be tested. Instead of the student generating the curriculum prior to continual pretraining, a third (peer) model could be used to curate the curriculum, or the student could generate the curriculum during training using the snapshotting technique (Zhao et al., 2021). Reverse curriculums would also be helpful to generate and evaluate as sanity checks. Properties such as overall text length, sentence length, diction and word frequency could also be used as heuristics for linguistic complexity and aid in curriculum curation. A curriculum pacing function could be used throughout the training process (Hacohen and Weinshall, 2019).

MLM loss as a proxy for difficulty could be criticised. Depending on which words are masked, an example text may arbitrarily achieve a higher or lower loss, independent of the overall complexity of the passage - other metrics to determine confidence could be explored. Lastly, It must be noted that there is also mixed evidence in support for the efficacy of curriculum learning in the context of language modeling, though this is a contested topic. (Campos, 2021).

Distant future work could include analysis on multi-teacher settings, and looking at other evaluation metrics like model convergence or model size.

6 Conclusion

In conclusion, KD improves model accuracy over baseline models, and curriculum learning seems to only marginally improve model accuracy in the context of knowledge distillation for large language models. Selective reliance can still be explored, though it may first be wise to conduct a more rigorous analysis of curriculum learning in the KD for LM setting before moving forward with more niche techniques.

7 Acknowledgments

Thank you to Professor Linzen and Guy Davidson for all their guidance towards this project. Thank you to especially to my peers, many of whom left actionable and useful feedback that led to changes in my experiments (for example, I dropped teacher based curriculum generation based on relevant peer feedback

A Appendix A: Data

A.1 BioASQ for Question Answering

The BioASQ (Tsatsaronis et al., 2015) is an annually updated expert annotated corpus comprised of various biomedical domain question answering tasks. The task selected for this experiment is 7b, which is a binary yes/no QA task. BioASQ is specifically chosen from amongst the BLURB tasks because it shows one of the widest performance gaps between small models trained on this task compared to larger expert models (Gu et al., 2021).

For all experiments, a 1500/100/166 train / validation / test split was used after undersampling the full dataset to account for class balance.

A.2 Pubmed Abstracts for Continued Pretraining

To conduct continual pretraining, a selection 1.81M abstracts from PubMed were downloaded from *ywchoi/pubmed-abstract-1* off HuggingFace. Naturally, this took a very long time to run.

B Appendix B: Loss Functions & Evaluation Metrics

B.1 MLM Loss

MLM loss is used both for continual pretraining, and as a way to generate confidence values for each span of text.

B.2 Accuracy

In keeping with its usage in the BLURB BioASQ task, accuracy is also used as the evaluation metric of choice for the BioASQ fine-tuning task. Care is taken to balance for class labels for accuracy to remain a useful metric.

C Appendix C: Experiment Parameters & Model Specifications

Optimal hyperparameters were found using randomized grid search sweeps using the Weights & Biases library. Included are hyperparameters for continued pretraining on PubMed abstracts and continued fine tuning on BioASQ.

Knowledge Distillation Hyperparameters for continued pre-training:

Model	**Hyperparameters**

DistillBert-KD	temperature=2.0,
	lambda = 0.5

DistillBert-KD-CL	temperature=2.0,
	lambda = 0.5

DistillBert-KD-CL-SR	–

Hyperparameters for fine-tuning on BioASQ:

Model	**Hyperparameters**

BioBert	batch-size = 8
	epochs = 2
	lr = 4.727e-06
	weight-decay = 0.1

DistillBert	batch-size = 8
	epochs = 2
	lr = 5.537e-05
	weight-decay = 0.05

DistillBert-KD-CL	batch-size = 8
	epochs = 2
	lr = 9.001e-05
	weight-decay = 0.1

DistillBert-KD-CL-SR	–

Model Specifications:

Model	Specifications

DistillBert	vocab-size = 28996
	parameters = 65M

Bio-Bert	vocab-size = 28996
	parameters = 108M

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 41–48, New York, NY, USA. Association for Computing Machinery.

Daniel Campos. 2021. Curriculum learning for language modeling.

Jang Hyun Cho and Bharath Hariharan. 2019. On the efficacy of knowledge distillation. *CoRR*, abs/1910.01348.

Shangchen Du, Shan You, Xiaojie Li, Jianlong Wu, Fei Wang, Chen Qian, and Changshui Zhang. 2020. Agree to disagree: Adaptive ensemble knowledge distillation in gradient space. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare*, 3(1):1–23.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *CoRR*, abs/2004.10964.

Guy Hacohen and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. *CoRR*, abs/1904.03626.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *CoRR*, abs/1901.08746.

Omid Rohanian, Mohammadmahdi Nouriborji, Samaneh Kouchaki, and David A. Clifton. 2022. On the effectiveness of compact biomedical transformers.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, Yannis Almirantis, John Pavlopoulos, Nicolas Baskiotis, Patrick Gallinari, Thierry Artieres, Axel Ngonga, Norman Heino, Eric Gaussier, Liliana Barrio-Alvers, Michael Schroeder, Ion Androutsopoulos, and Georgios Paliouras. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC Bioinformatics*, 16:138.

Sergey Zagoruyko and Nikos Komodakis. 2016. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer.

Haoran Zhao, Xin Sun, Junyu Dong, Zihe Dong, and Qiong Li. 2021. Knowledge distillation via instance-level sequence learning.