**The unreasonable effectiveness of distillation - Text Classification Part 1/3**

In recent years, increasingly large Transformer-based models such as BERT, have demonstrated remarkable state-of-the-art (SoTA) performance in many Natural Language Processing (NLP) tasks and have become the de-facto standard.

However there is no free lunch (indeed? Wait you read this blog series), those models are extremely inefficient and require massive computational resources and large amounts of data as basic requirements for training and deploying. This severely hinders the scalability and deployment of NLP-based systems across the industry.

I have always been fascinated by robustness and efficiency in NLP targeting production. So I decided to write a series of blogs that will provide some practical tips with code samples regarding how to deploy and adapt SoTA large transformer models.

In the first 3 blogs I will focus on model distillation for Text-Classification. Model distillation is a very powerful pruning technique and in many use-cases yields significant speed-up and memory size reduction. But still considered more for advanced users: relatively hard to implement, un-predictable performance, minimization of multi-loss errors, and lack of understanding of the inherent mechanism. I will try to show the opposite by providing a few real use-case examples with simple code snippets and distiiation’s efficacy intuition.

Ok so let’s start.

Suppose that you are a data scientist in one of the top leading enterprise companies and your task is to classify social media tweets and deliver SoTA models into production. You have also been told that your model must be very efficient since you will pay a high fee for any extra milion parameters. So most likely you will start by collecting sufficient data (a few hundred/thousands of labeled samples) and compare a few ML/DL models and Transformer models to target high accuracy with minimum cost (model size).

In order to demonstrate the accuracy achieved by different model types I choose an emotion classification dataset ([Emotions)](https://huggingface.co/datasets/emotion) that consists of Twitter posts labeled with any of six basic emotion categories: sadness, disgust, anger, joy, surprise, and fear. The data consists of 16K training samples and 2K test samples and available on HuggingFace data-set Hub. A code example for the following steps are available here.

**1st step: Set a baseline using a logistic regression model (TfIdf based)**

Accuracy = 86.1%

Our baseline result.

**2nd step: Deep Learning baseline**

Next, we will try a simple MLP Neural Network (NN). The model architecture is very basic and includes an input embedding size of 16, a word vocabulary size of 5000, and one hidden layer.

Accuracy = 86%

Number of parameters = 80K

Similar accuracy to Logistic regression but more efficient and dense. We are veterans in the NLP domain and by now already practiced with HuggingFace and their amazing “transformers” library so let’s try a few popular SoTA transformer models.

**3rd: Transformers models**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Accuracy** | **#Parameters** | **Implementation Source** |
| BERT | 92.4 | 110M | Our |
| Roberta | 93 | 110M | Our |
| T5 | 93.5 | 11B | [t5-base-finetuned-emotion](https://huggingface.co/mrm8488/t5-base-finetuned-emotion) |

Awsome, accuracy flies to the sky but … 110M parameters are far above our computational budget, and for sure IT hits the ceiling when they hear about the 11B model (:

So instead let’s try one of the more popular models like DistillBERT that HugginngFace released, half the size and double the speed vs. BERT base.

**4th step: DistillBERT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Accuracy** | **#Parameters** | **Implementation Source** |
| DistillBERT | 91.5 | 67M | Our |
| DistillRoberta | 92.3 | 67M | [Elvis Saravia](https://colab.research.google.com/drive/1nwCE6b9PXIKhv2hvbqf1oZKIGkXMTi1X#scrollTo=ZhHutCseBxjJ) |

Nice, we gained <1% loss vs. BERT model and a much smaller one.

So that’s it? Are we done? Are we ready to go with this model to production and pay the computing cost for the 67M parameters?

Can we do more? Can we do with less than 1M or 100K with minimal accuracy loss (<1% loss)?

The answer is “yes” (a spoiler: but in general the answer varys and depends on your dataset quality and task in-hand) with the help of distillation and additional data either augmented from the training set or sampled from your in-domain unlabeled data-set.

Recall BERT is a pre-trained language model trained for the Mask Language Model task and here we are interested only in emotion classification. Previous research shows that fine-tuned BERTs parameters are over-parameterized for domain specific tasks ([Kovaleva at al., 2019](https://arxiv.org/abs/1908.08593)).

Knowledge distillation (KD) to very simpler architectures ([Tang et al., 2019](https://arxiv.org/pdf/1903.12136.pdf); [Wasserblat et al., 2020](https://www.aclweb.org/anthology/2020.sustainlp-1.5.pdf)) has shown promising results for reducing the model size and computational load while preserving much of the original model’s performance. A typical model distillation setup includes two stages. In the first stage, a large, cumbersome and accurate teacher neural network is trained for a specific downstream task. In the second stage, shown in Figure 1, a smaller and simpler student model that is more practical for deployment in environments with limited resources, is trained to mimic the behavior of the teacher model.

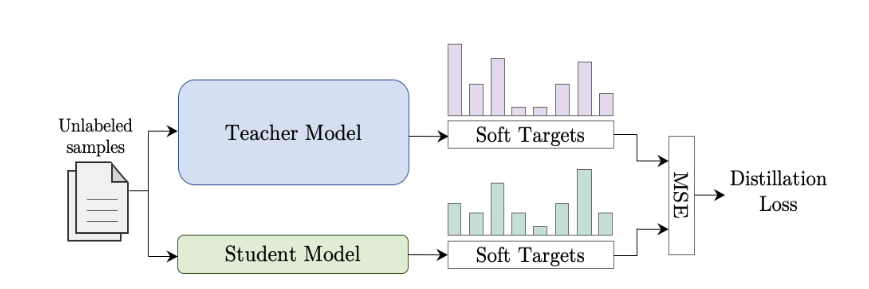


Figure 1: Student model training process.

Code disclaimer: In order to make KD super easy, we only use the distillation loss which is calculated for each training batch by calculating Kullback–Leibler (KL) distance between the target predictions that are produced by the student and teacher models. We didn't notice any performance loss vs. deploying MSE between soft targets (logits), nor adding temperature as in the distillation original paper ([Hinton et al., 2015](https://arxiv.org/abs/1503.02531)). The KL loss allows us to use the same code for standard training (labeled data) or distillation (pseudo labeled data).

**5th step: Distill Roberta to a simpler student**

In our case let’s distill Roberta’s knowledge into our simple MLP NN.

Following are the results:

|  |  |  |
| --- | --- | --- |
| **Model** | **Accuracy** | **#Parameters** |
| MLP | 86 | 80K |
| MLP\_D | 91.8 | 80K |

Wow!!! Surprisingly not bad at all, on-par accuracy with DistilBERT.

The following figure summarizes the results that we achieved so far for Emotion in terms of (model acc./BERT acc.)% Vs. model size.

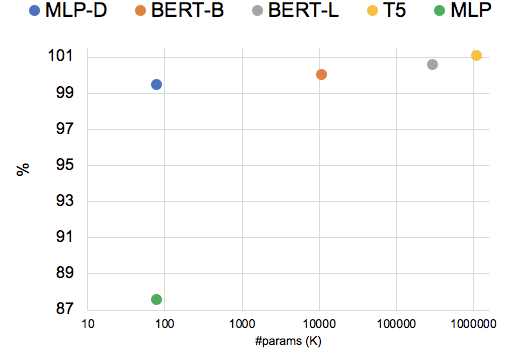


Figure: Unreasonable example for the effectiveness of distillation: Accuracy of a distilled MLP student (80K parameters) is on-par with BERT (the teacher with 110M params), BERT Large (330M params) and T5 (11B params) on Emotions dataset.

Ok, We were able to distill Roberta’s knowledge into our tiny model with almost no loss.

We benefit from high transformer model performance with very little cost to pay (and for our IT).

In this stage you probably have many questions, here are few examples:

* Does this “trick” hold for any classification sub-task (or any other NLP’s tasks) specifically on our own data-set?
* If not, when does it work?
* How to choose the best student for my data-set?
* What is the intuition behind this mechanism?

I’ll try to answer those questions and give a bit of intuition in the following blogs.

**The unreasonable effectiveness of distillation - Text-Classification Part 2/3**

In the first blog, we achieved a surprisingly good performance by distilling large (teacher) models into tiny (student) models that are on-par with the transformer mammoth models. In this blog, we intend to further explore it and verify whether these results can also be achieved for other text-classification’s data-sets and sub-tasks.

For that matter we choose SST-2 and CoLA which are popular single-sentence classification datasets and are part of the widely used General Language Understanding Evaluation ([GLUE](https://arxiv.org/abs/1804.07461)) benchmark.

**SST-2 Dataset**

SST-2 The Stanford Sentiment Treebank 2 comprises single sentences extracted from movie reviews and binary (positive/negative) sentiment classification labels.

For SST-2, we use the dataset version provided [here](https://github.com/clairett/pytorch-sentiment-classification/blob/master/data/SST2/train.tsv), the data consists of 6920 training samples and 1800 test samples..

Following are the results of few transformer models, MLP and the distillation of the RoBERTa model into the MLP.

|  |  |  |
| --- | --- | --- |
| **Model** | **Accuracy** | **#Parameters** |
| RoBERTa (teacher) | 93 | 110M |
| BERT-base | 91.4 | 110M |
| [DistillBERT](https://arxiv.org/pdf/1910.01108.pdf) | 90.4 | 66M |
| MLP | 79 | 80K |
| MLP-D | 86 | 80K |

Oops, we notice a drop of 6% with MLP-D compared to BERT-base.

So, this time the tiny model fails to learn the full capacity of knowledge that allows it to decode the classification task as well as the teacher model.

Instead, let's try to replace our student model with a much deeper architecture but still significantly smaller vs. BERT: A Bi-LSTM with 1.7M parameters (x80).

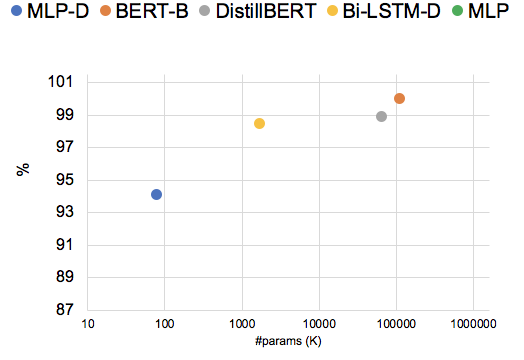
Following are the results of the Bi-LSTM student distilled model.

|  |  |  |
| --- | --- | --- |
| **Model** | **Accuracy** | **#Parameters** |
| Bi-LSTM-D | 90 | 1.7M |

Not bad, we only have a 2% accuracy drop and we are on-par with DistillBERT.

So, for SST-2, a simple Bi-LSTM student whould be considered as sufficient for production purposes.

The following figure summarizes the results that we achieved so far for SST-2 in terms of (model acc./BERT acc.)% Vs. model size.



**The CoLA Dataset** and Task

[CoLA](https://nyu-mll.github.io/CoLA/) The Corpus of Linguistic Acceptability consists of English acceptability judgments drawn from books and journal articles on linguistic theory. Each sentence is associated with a label that indicates whether it is a grammatical English sentence or not.

Following are the results for the teacher model (BERT) and the two student distilled models :

|  |  |  |
| --- | --- | --- |
| **Model** | **MCC** | **#Parameters** |
| BERT | 52.8 | 110M |
| MLP+D | 10 | 80K |
| Bi-LSTM+D | 24 | 1.7M |

Whoops, in contrast to the SST-2 dataset even the Bi-LSTM doesn’t hold the capacity to match BERT on the CoLA task. Both the Bi-LSTM and the MLP+Attn.models are far from being comparable to BERT. In the case of SST-2 task, Bi-LSTM which is a larger and deeper model than MLP closed the gap with the teacher model (BERT).

Let’s try a model with a larger and deeper model than Bi-LSTM for the CoLA task, but one that is still more efficient compared to the teacher model.

We chose DistilBERT, a very popular model by HuggingFace based on self-distill of logits, and embeddings.

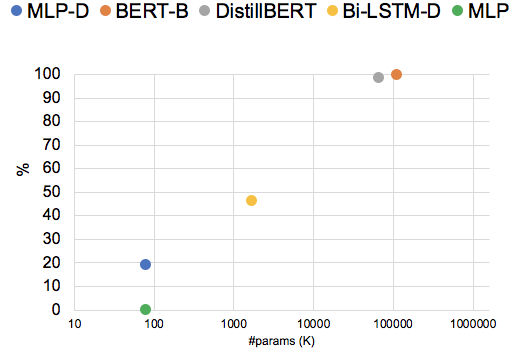
Following are the results:

|  |  |  |
| --- | --- | --- |
| **Model** | **MCC** | **#Parameters** |
| DistilBERT | 51.3 | 66M |

Great, now we only have 1.5% accuracy drop and we are on-par with BERT base.

So, for the CoLA task, a self-distilled Bert student will be considered as sufficient for production purposes.

The following figure summarizes the results that we have so far with CoLA.



**Note**: At this point it may be trivial to think that Bi-LSTM/MLP didnt hold the full capacity of BERT since they have simpler architecture but remember that KD dependent on the availability of un-labeled in-domain data. In the case of CoLA we didnt have sufficient in-domain data so we had to generate data using data different augmentation techniques. ColA’s grammatical acceptence task is very challenging so not sure we were able to generate high quality data that represent the full task’s data distribution. This is very important research direction, we have noticed significant improvement (2-3 points) when distill with generative model (e.g. GPT-2) vs. Easy Data Augmentatoion (EDA) which based on semantic replacement of random key-words.

**Some Intuition**

In general we showed that it is feasible to distil BERT using very efficient models while preserving comparable results. However, the success of the distillation (student size vs. accuracy) depends on the dataset and task at hand.

What is the reason for such variance in performance?

In order to answer this question we need first to look into the different datasets.

Here are few data instances taken from the 3 dataset:

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset** | **Instance examples** | **Category** | **Task** |
| Emotion | im updating my blog because i feel shitty | Sadness | Emotion |
|  | i just feel greedy and lame making one | Anger |  |
|  | im feeling more comfortable with derby | Joy |  |
| SST-2 | works because , for the most part , it avoids the stupid cliches and formulaic potholes that befall its brethren | POS | Sentiment |
|  | but the characters tend to be cliches whose lives are never fully explored | NEG |  |
| [CoLA](https://arxiv.org/pdf/1805.12471.pdf) | many evidence was provided | unacceptable | Grammatical acceptability |
|  | The jeweller inscribed the ring with the name | acceptable |  |
|  | The gardener planted roses in the garden | acceptable |  |
|  | The more books I ask to whom he will give, the more he reads | unacceptable |  |

A successful classification of the Emotion task seems to be heavily dependent on lexical cues - salient emotional words that represent the emotional category regardless of structure and syntax. On the other hand the CoLA task is inherently dependent on the syntax structure and less on lexical cues. Whereas, successful classification of theSST-2 task seems to be dependent on a mix of lexical and syntactic clues.

So in order to successfully distill the full teacher knowledge which is required for a given task, the student architecture must be capable to absorb its teacher capacity that is related to the task. In the case of CoLA and partially SST-2, the architecture of the MLP model does not have the capacity to learn the full syntactic information stored in the teacher. This is clear since the MLP’s architecture is basically based on Bag Of Word (BOW) embedding implementation. Classification of the Emotion’s task requires the learning of mostly semantic lexical knowledge therefore MLP is sufficient and Bi-LSTM and DistillBERT are over-qualified,

.

To summarize: Classification tasks that require capturing of general lexical semantics cues can be successfully distilled by very simple and efficient models; however, classification tasks that require the detection of linguistic structure and contextual relations are more challenging for distillation using simple student models

In the next blog, we will suggest a simple metric to estimate the success of the distillation.

We also showcase a simple dynamic architecture that utilizes this metric to achieve optimal tradeoff between size and accuracy for a given data set/task. Then we propose a few future research directions.

**The unreasonable effectiveness of distillation - Word Order Sensetivity and Dynamic Data Aware Transformer Part 3/3**

In the last two blogs we saw useful examples of text classification distillation methods.

We presented some intuition why distillation works and how to choose the smallest student model that can match the capacity of its teacher model.

In this blog, I would like to suggest a metric for estimating the complexity level of your dataset/task and exploit it to optimize the distillation performance. We’ll then propose a naive switch transformer architecture in order to maximize the simplicity of our dataset/task to the efficiency in terms of student model size. We will end the series of blogs with a few thoughts on the transformer structure and future research directions.

**Word order sensitivity**

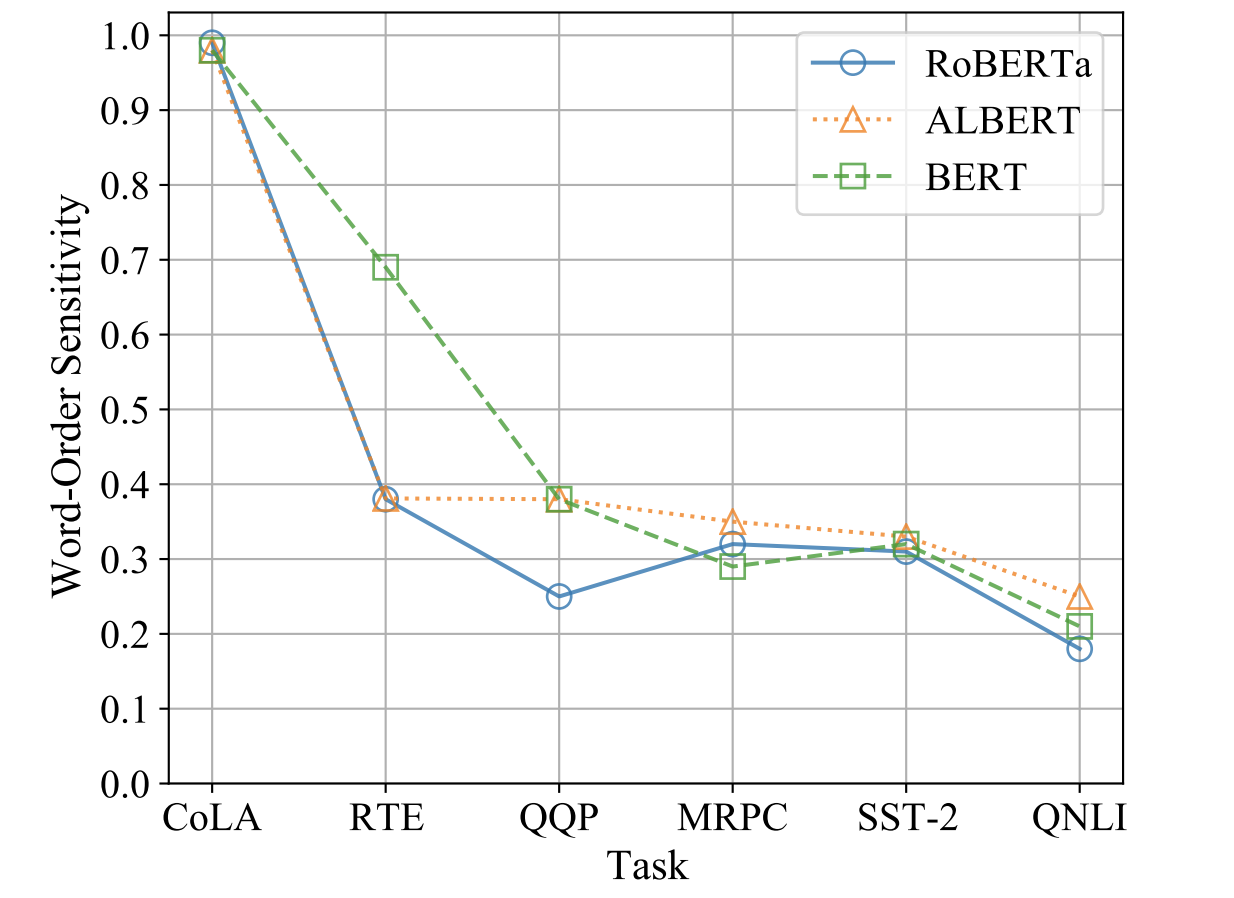
As we noted “simple” instances/examples means that prediction is mostly based on lexical semantic cues and seems to be rather agnostic to syntax or word order. [Thang et al., 2020](https://arxiv.org/pdf/2012.15180.pdf) showed a surprising phenomenon: between 75% and 90% of the correct predictions of Transformer-based classifiers, trained on GLUE tasks, remained unchanged when the input words were randomly shuffled. The authors further suggested a simple metric to measure dataset sensitivity to word order:

WOS (Word Order Sensitivity) = (100-p)/50

where p is the accuracy of a task-trained model evaluated on a dev-s set

(See Thang’s Sec 3.1 and 2.3.2)

Below is a figure taken from [Thang et al.](https://arxiv.org/pdf/2012.15180.pdf), that shows the WOS scores plotted for various GLUE tasks followed by a table that presents our measure of BERT WOS score for Emotion, SST-2 and CoLA datasets (1-gram shuffling):



|  |  |
| --- | --- |
| **Dataset** | **WOS** |
| Emotion | 0.14 |
| SST-2 | 0.34 |
| CoLA | 0.99 |

Our intuition, discussed in the previous blog, was right! The CoLA dataset with average WOS score of 0.99 which means that it consists of a vast majority of “hard” samples where the Emotion dataset has the lowest WOS score which means that it consists of a vast majority of “simple” samples.

The SST-2 WOS score is 0.34 which means that it tends to have more “simple” instances than “hard” ones. Those results are quite consistent with the distillation performances (student model size vs. accuracy): the Emotion dataset was successfully (almost without loss) distilled to a tiny MLP model, SST-2 to Bi-LSTM model and CoLA to DistillBERT model.

**IMDB’s example**

Lets apply our new metric to the popular IMDB dataset and try to predict the distillation results.

IMDB The Internet Movie Database ([IMDB](https://huggingface.co/datasets/imdb)) comprises single sentences extracted from informal movie reviews for binary (positive/negative) sentiment classification.

The training data consists, for this example, a subset of 1K randomly sampled samples from the 25K training samples and 25K test samples.

We have calculated the WOS score of IMDB to be 0.28. Since WOS is relatively low (<0.3) we anticipate that an MLP or Bi-LSTM models should be suitable for absorbing the capacity of its teacher model (Roberta in this case).

Here are the results of the distillation of the IMDB dataset/task :

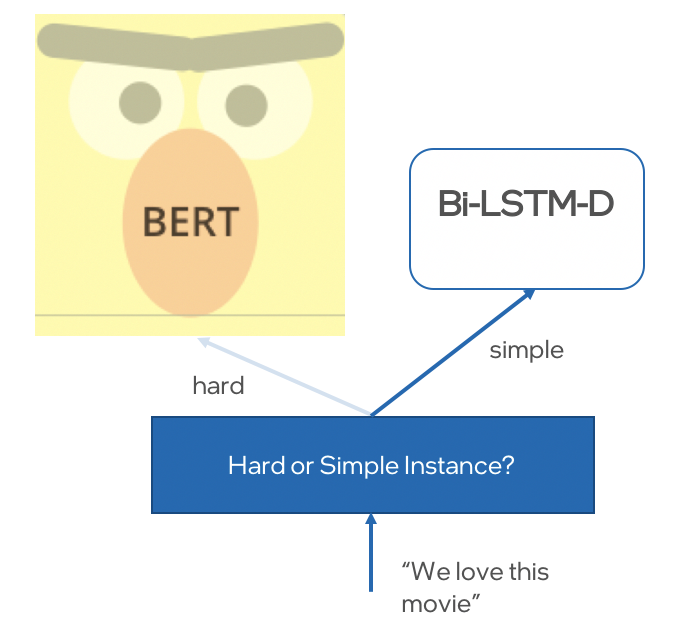
|  |  |  |
| --- | --- | --- |
| **Model** | **Accuracy** | **#Model Parameters** |
| BERT | 88.6 | 110M |
| DistilBERT | 87.7 | 110M |
| RoBERTa (teacher) | 91.6 | 110M |
| MLP | 80 | 80K |
| MLP+D | 90 | 80K |
| Bi-LSTM+D | 91 | 0.7M |

Indeed, as we predicted, a small MLP/Bi-LSTM model is capable of absorbing RoBERTa’s knowledge for the IMDB dataset/task and even outperforms BERT performance.

**The Switch Architecture**

[Scwartz et al., 2020](https://arxiv.org/pdf/2004.07453.pdf) proposed a transformer early exit based on the confidence of the prediction in each layer, or in other words the transformer dynamically expands/shrinks its size during inference based on the complexity of each inference sample. As we showed so far, a dataset with the majority of simple instances can be distilled by a very efficient model. So, continuing along this line we suggest to apply a “simple\hard” predictor of each input instance during inference time, and decide which student model to use based on this prediction. We refer to it as “the switch architecture”: for the “simple” instances use the student model and for the “hard” ones use a teacher model (no distillation). Since the student’s model architecture is considerably more efficient in relation to the teacher model (e.g. MLP/Bi-LSTM vs. BERT) for a dataset that consists of **a majority of simple instances** the average speed-up boost will be high with the switch architecture

Below a diagram that presents an abstract view of the switch architecture and followed by simple/hard instance examples from SST-2 dataset:



|  |  |  |
| --- | --- | --- |
| **Dataset** | **Simple Instance examples** | **Hard Instance examples** |
| SST-2 | extremely confusing | the issue of faith is not explored very deeply |
|  | the rock is aptly named | works because , for the most part , it avoids the stupid cliches and formulaic potholes that befall its brethren |
|  | sturdy entertaining period drama both caine and fraser have their moments | what it lacks in originality it makes up for in intelligence and b grade stylishness |
|  | moving and invigorating film | succeeds in providing a disquiet world the long dreaded completion of the police academy series |
|  | an intelligent | but the characters tend to be cliches whose lives are never fully explored |
|  | good movie | the story and characters are nowhere near gripping enough |

Following are some examples of the potential speed-up gains that can be achieved for the SST-2 dataset/task with different student models:

|  |  |  |
| --- | --- | --- |
| **Model** | **\*Speed-Up vs. BERT-base** | **Accuracy** |
| **RoBERTa (teacher)** | x1 | 93.54 |
| **BERT-base** | x1 | 91.4 |
| **DistillBERT** | x2 | 90.4 |
| **Bi-LSTM** | \*\*x40 | 81 |
| **Bi-LSTM-D** | x40 | 90 |
| Switch Bi-LSTM-D+BERT  5% hard samples | x20 | 91.4 |
| Switch Bi-LSTM-D+BERT  20% hard samples | x5 | 92.5 |

\*In our code example the instance classifier (hard/simple) implementation is far from being efficient, we need to run BERT twice on each test data instance. I’ll leave it for future research and be happy to get ideas to improve it. In the table, the computing overhead for the classification has been neglected.

\*\* Bi-LSTM relative speed-up as measured [here](https://www.aclweb.org/anthology/2020.sustainlp-1.5.pdf).

**Data aware transformer and future research**

BERT holds extensive knowledge on language structure including semantic/syntactic cues. It well known that after fine-tuning BERT utilizes partial and sufficient knowledge to solve a given task on a specific domain. What maybe innovative contribution to BERTology, based on the switch arch results above, it seems that during inference BERT “retrive” sufficient knowledge to decode a specific instance, for simple examples shallow information (aka semantic) for hard examples higher level information (aka sysntactic or even world knolwedge). As a result data aware training allows to distill this knowledge into simpler architecture with limited capacity.

We encourage to continue and explore data aware optimization techniques in order to

dynamically adapt transformer size and speed in production. In addition, it would be important to understand when a data instance is “simple” or “hard” for a given task and exploit this prediction for an early exit or efficient switch arch. For challenging tasks (majority of the data are “hard” instances) like CoLA, SQuAD, OpenQA would be interesting to explore how external knowledge retrieval could transform “hard” instances to “simple” ones to gain maximum efficiency.