



# **Learning small sentence classifiers from BERT using teacher-student knowledge distillation**

*Sam Sučík*

**MInf Project (Part 2) Report**

Master of Informatics  
School of Informatics  
University of Edinburgh

2020



# Abstract

## **Acknowledgements**

Thanks to Steve and Vova, to Ralph Tang and to Slávka.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Motivation . . . . .	7
1.2	Aims . . . . .	8
1.3	Contributions . . . . .	8
<b>2</b>	<b>Background</b>	<b>9</b>
2.1	Pre-Transformer sequence encoders . . . . .	9
2.2	Transformer models and BERT . . . . .	9
2.3	Teacher-student knowledge distillation . . . . .	10
2.3.1	Brief history of knowledge distillation . . . . .	10
2.3.2	Knowledge distillation in NLP . . . . .	11
2.4	Understanding the models . . . . .	13
<b>3</b>	<b>Datasets</b>	<b>15</b>
<b>4</b>	<b>Implementation</b>	<b>17</b>
<b>5</b>	<b>Student tuning</b>	<b>19</b>
<b>6</b>	<b>Analysing the students</b>	<b>21</b>
<b>7</b>	<b>Overall discussion and conclusions</b>	<b>23</b>
<b>8</b>	<b>Future work</b>	<b>25</b>
<b>9</b>	<b>Notes</b>	<b>27</b>
9.1	Knowledge distillation – assorted . . . . .	27
9.2	Datasets . . . . .	27
9.3	Plans for MVP . . . . .	28
	<b>Bibliography</b>	<b>29</b>



# Chapter 1

## Introduction

### 1.1 Motivation

- After the deep learning hype started, NLP went through an era of LSTMs. Since 2017, the area has been becoming dominated by Transformer models pre-trained on large unlabelled corpora.
- As newer and bigger Transformer-based models were proposed in 2018 and 2019, improving on the SOTA, it was becoming clearer that their big size and low speed was rendering them difficult to use (both train and deploy) in practice outside of research labs.
- Recently, we've seen various early attempts at making Transformers – in particular BERT ([Devlin et al., 2018](#)) – smaller by removing attentional heads ([Michel et al., 2019](#)), quantisation and pruning ([Cheong and Daniel, 2019](#); [Sucik, 2019](#)). In terms of actually down-sizing and accelerating the models, knowledge transfer using teacher-student knowledge distillation has led to the most attractive results ([Mukherjee and Awadallah, 2019](#); [Tang et al., 2019](#); [Jiao et al., 2019](#); [Sanh et al., 2019](#)).
- However, these studies focus only on using knowledge distillation as a tool. Important questions about the nature of this technique and how it interacts with properties of the teacher and student models remain generally unexplored.
- In line with the increasing demand for explainable AI, it is desirable to better understand how knowledge distillation works, in this case when distilling NLP knowledge from Transformer models. Such understanding can also help researchers improve the ways knowledge distillation is used or modified for various use cases.

## 1.2 Aims

- Explore the effectiveness of knowledge distillation in very different NLP tasks. To cover a broad variety of tasks, I use datasets ranging from binary sentiment classification to 57-way intent classification to linguistic acceptability.
- Explore how distilling knowledge from a Transformer varies with different student architectures. I limit myself to using the extremely popular BERT model (Devlin et al., 2018) as the teacher architecture. As students, I use two different architectures: a BiLSTM, building on the successful work of Ralph Tang (Tang et al., 2019; Tang and Lin, 2019), and a down-scaled BERT architecture.
- Explore how successfully can different types of NLP knowledge and capabilities be distilled. Since NLP tasks are often possible for humans to reason about, I analyse the models' behaviour (e.g. the mistakes they make) to learn more about knowledge distillation. I also probe the models for different linguistic capabilities, building on previous successful probing studies (Conneau et al., 2018; Tenney et al., 2019).

## 1.3 Contributions

My actual findings. To be added later.



# Chapter 2

## Background

### 2.1 Pre-Transformer sequence encoders

NLP is all about sequences of variable lengths: sentences, sentence pairs, documents, speech segments...

NLP tasks are typically about making simple predictions about sequences: classifying sentences based on their intent or language, scoring a document's level of formality, predicting whether two sentences form a coherent question-answer pair or not, predicting the next word of a sentence...

Machine learning predictors are typically designed to work with fixed-size representations of inputs. Therefore, ever since the resurgence of neural networks around 2010, neural NLP has been using various models for encoding variable-length sequences into common fixed-dimensional representations.

RNN- and later LSTM-based encoder architectures were dominating the area for a long time as they were naturally suited for processing sequences of any length.

A major breakthrough came when [Kalchbrenner and Blunsom \(2013\)](#) and [Sutskever et al. \(2014\)](#) developed the encoder-decoder architecture for machine translation and other sequence-to-sequence tasks such as paraphrasing or parsing.

[Bahdanau et al. \(2014\)](#) improved things by introducing attention, enabling the recurrent encoders to learn to selectively attend or ignore parts of the input sequence.

### 2.2 Transformer models and BERT

[Vaswani et al. \(2017\)](#) introduced Transformer. Main idea: process tokens in parallel, not sequentially, with sequentiality represented by positional markers (embeddings). Self-attention is used to pool from the context of the entire sequence, leading to evolving rich contextualised representations of each token in the higher layers.

[Radford et al. \(2018\)](#) introduced the idea of generative LM pre-training and fine-tuning. This concept helps train much better models even for low-resource tasks with small datasets, by leveraging general language knowledge acquired by the model in the pre-training phase. Publishing pre-trained model instances makes the power of NLP much more accessible to anyone and has become a popular thing to do.

[Devlin et al. \(2018\)](#) made improved the concept by pre-training the model bi-directionally (leading to language modelling based on left *and* right context). They also changed the pre-training to 2 tasks trained at the same time: masked language modelling to learn to understand words, and next sentence prediction to learn to reason about entire sentences (as the actual NLP tasks often require such reasoning). This is how BERT was born, which then became extremely popular in the community, attracting a lot of work on improving it, analysing its capabilities, extending it to other languages, and even applying it to multi-modal tasks such as video captioning.

Following the success of BERT, further and often bigger Transformer models started emerging:

- GPT-2 ([Radford et al., 2019](#)), a bigger and improved version of GPT
- XLM ([Lample and Conneau, 2019](#)) with added introduced cross-lingual pre-training
- Transformer XL ([Dai et al., 2019](#)) with better handling of much longer contexts
- and many others

Although the open-sourced powerful pre-trained models were a huge step towards more accessible NLP, the model size meant they couldn't be applied easily outside of research: They were memory-hungry and slow. This inspired another wave of research: compressing the huge, well-performing Transformers (very often BERT) to make them faster and resource-efficient. I will focus on the compression method that so far looks the most effective: knowledge transfer from huge models into smaller ones using teacher-student knowledge distillation.

## 2.3 Teacher-student knowledge distillation

### 2.3.1 Brief history of knowledge distillation

The concept of knowledge distillation was introduced by ([Bucila et al., 2006](#)) as a way of knowledge transfer from huge models (or ensembles of models) into small ones, with the aim of having smaller (and hence faster) yet well-performing models. The main idea is to train a big neural classifier model also called the *teacher* and then let a smaller neural classifier model (called the *student*) learn from it – by learning to mimic the teacher's behaviour. Hence also the term *teacher-student knowledge distillation*. The mimicking was originally realised by labelling a dataset with the trained teacher model and training the student model on these labels (retrospectively referred to as

*hard labels*). The dataset used for training the student model is often called the *transfer dataset*.

Later, [Ba and Caruana \(2014\)](#) introduced the idea of learning from the *soft labels*, i.e. learning to predict the teacher’s logits. The idea is that the teacher’s knowledge can be transferred to the student in a richer way if the entire logit distribution is utilised (compare with hard labels, when only the information about the maximum element of the logit distribution is utilised). When first proposed, the student’s loss function was then mean squared distance between the student’s outputs and teacher’s logits.

[Hinton et al. \(2015\)](#) proposed a more general approach, addressing the issue of over-confident teachers with very sharp logit distributions, and formulated a new loss for student models: the cross-entropy loss with temperature. The temperature softens the teacher’s sharp distribution, allowing the student to learn more rich information from it. This approach is what is today typically referred to as knowledge distillation.

Since 2015, further variants have been proposed, enhancing knowledge distillation in different ways, for example:

- [Papamakarios \(2015, p. 13\)](#) points out that mimicking teacher outputs (e.g. with cross-entropy loss) can be extended to mimicking the derivatives of the loss w.r.t. inputs (i.e. including in the loss function also this term:  $\frac{\partial \mathbf{o}_{student}}{\partial \mathbf{x}} - \frac{\partial \mathbf{o}_{teacher}}{\partial \mathbf{x}}$ ).
- [Romero et al. \(2015\)](#) introduced the idea of learning to match not just the teacher’s outputs, but also its intermediate input representations. [Huang and Wang \(2017\)](#) achieved this by learning to align the distributions of neuron selectivity patterns between the teacher’s and student’s hidden layers. Unlike standard knowledge distillation, this approach no longer works just for classifier models with a softmax output layer (see the standard loss proposed by [Hinton et al. \(2015\)](#)).
- [Sau and Balasubramanian \(2016\)](#) show that learning can be more effective when noise is added to the teacher logits.
- [Mirzadeh et al. \(2019\)](#) show that a knowledge distillation performs poorly from large teachers into very small students and alleviate this limitation by multi-stage distillation: first distilling knowledge into an intermediate-size “teacher assistant” and learning the student from the assistant model.

### 2.3.2 Knowledge distillation in NLP

Practically all the so far mentioned research in knowledge distillation was done in the domain of image processing. This comes as no surprise: It was image processing that was benefitting the most from the resurgence of deep learning. Ever since the AlexNet ([Krizhevsky et al., 2012](#)), bigger and bigger models were proposed, simultaneously driving the research in model compression so as to make the models usable in practice.

In natural language processing, research on knowledge distillation was rare for a long time. One notable work was the adaptation of distillation for sequence-to-sequence machine translation models – whose outputs are no longer simple classification scores

– by [Kim and Rush \(2016\)](#). Another pioneering study compressed a recurrent neural language model for use on mobile devices ([Yu et al., 2018](#)).

However, the real need for model compression started very recently when huge Transformer models were introduced. This follows on from the main limitation of the otherwise very accessible pre-trained models: being huge and slow.

When distilling knowledge from big, pre-trained Transformer models, the main decision is whether to distil before or after fine-tuning on a concrete downstream task. Each option has its pros and cons. In the first scenario, the steps are: 1) distilling into a small student, 2) fine-tuning the student on any downstream task. One advantage is that the distillation is done only once and fine-tuning the student can be fast (due to the student’s size). Since the distillation can be done on the same data that the teacher was pre-trained on – large corpora of unlabelled data, one does not have to worry about not having enough data. One possible downside is that the amount of general language knowledge contained in the pre-trained teacher will be too much to contain within a small student, hence requiring students that are themselves considerably large (and slow). [Sanh et al. \(2019\)](#) took this approach and while their student model is very successfully fine-tuned to a wide range of tasks, it is smaller than the teacher (BERT base) only by 40%, still having 66M parameters.

In the second scenario, the steps are: 1) fine-tuning the pre-trained teacher on a downstream task, 2) distilling the teacher into a small student. In this case, the downstream task-specific language knowledge is likely to be much smaller than the teacher’s overall pre-trained language knowledge, meaning that a much smaller student can still contain all important knowledge from the fine-tuned teacher. However, with this approach, the teacher fine-tuning and distillation has to be done separately every downstream task, which can be resource-intensive. Even more importantly, the distillation procedure can suffer from lack of data, since the distillation is now being done only with the downstream task dataset, which will often be small and has to be augmented. Various ways of augmenting small datasets to increase the amount of transfer data have been proposed, with mixed success. [Mukherjee and Awadallah \(2019\)](#) use additional unlabelled in-domain sentences with labels generated by the teacher – an approach that only works if such in-domain sentences are available. [Tang et al. \(2019\)](#) propose augmentation based on simple, rule-based perturbation of existing sentences from the downstream task data. Finally, [Jiao et al. \(2019\)](#) and [Tang and Lin \(2019\)](#) use big Transformer models generatively to create new sentences. In the first case, BERT is repeatedly used to choose a suitable replacement for an existing word in a given sentence, eventually leading to a new sentence with many words changed for different ones. In the second case, a GPT-2 model is fine-tuned on the downstream task with a language model objective, and a new sentence is generated by repeatedly predicting the next token, conditioned on the sequence generated so far.

In this work, I adopt the approach of [Tang and Lin \(2019\)](#). However, while they use only bidirectional LSTM students, I also experiment with a smaller version of BERT, similarly to [Jiao et al. \(2019\)](#). For detailed description of the system see [Chapter 4](#).

## **2.4 Understanding the models**



# **Chapter 3**

## **Datasets**





# **Chapter 4**

## **Implementation**



# **Chapter 5**

## **Student tuning**



## **Chapter 6**

### **Analysing the students**



## **Chapter 7**

### **Overall discussion and conclusions**





## **Chapter 8**

### **Future work**



# Chapter 9

## Notes

### 9.1 Knowledge distillation – assorted

? use KD of DNNs into decision forests for interpretability.

### 9.2 Datasets

Unsuitable:

1. [ATIS](#): too easy (see [here](#)). Rasa version [here](#).
2. [AskUbuntu](#), [Chatbot](#) and [Web Applications](#) (all from TU Munich): too small (max. 206 datapoints). Rasa version [here](#).
3. [SNIPS](#): too easy (see [here](#)). Rasa version [here](#).

Suitable:

1. [FB's Multilingual Task Oriented Dataset](#): F1 0.99 by supervised embeddings (very easy, but perhaps usable). Rasa version [here](#).
2. [CoLA](#) ([Warstadt et al., 2018](#)), needs processing into Rasa format.
3. [SST](#) ([Socher et al., 2013](#)), needs processing into Rasa format. 5-way classification may be too hard (accuracy ~50%), 2-way much easier.
4. [TREC question-type classification](#) ([Voorhees and Tice, 2000](#)), needs processing into Rasa format. 6-way classification (abbreviation, entity, description, human, location, numeric), also has another (more fine-grained) level of categories.

Maybe suitable:

1. [Microsoft Dialogue Challenge](#) ([Li et al., 2018](#)) needs processing into Rasa format. Also, no results using Rasa.

2. [TOP \(Gupta et al., 2018\)](#) needs processing into Rasa format. Also, no results using Rasa. Intents are hierarchical, would need to take only the top-most intent.
3. [SciCite \(Cohan et al., 2019\)](#) needs processing into Rasa format. Also, no results using Rasa.

Probing/evaluation: Started by Shi et al. (2016) and Adi et al. (2017)?

1. [Google's edge probing \(?\)](#) for evaluating span representations on 9 tasks closely following classical NLP pipeline (data not freely accessible!)
2. [FB's probing \(?\)](#) used to evaluate entire sentence embeddings (10 tasks from sentence length to semantics)
3. [FB's SentEval \(Conneau and Kiela, 2018\)](#) is meant for evaluating trained sentence encoders (i.e. not meant as downstream tasks that encoders should be fitted to), but it curates interesting existing datasets.

### 9.3 Plans for MVP

Let's distill BERT into a smaller BERT.

Code: pytorch-transformers (re-using code for DistilBERT), no Rasa for now.

Dataset: CoLA (because pytorch-transformers offers easy feeding of GLUE datasets).

# Bibliography

- Ba, J. and Caruana, R. (2014). Do deep nets really need to be deep? In *NIPS*.
- Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate.
- Bucila, C., Caruana, R., and Niculescu-Mizil, A. (2006). Model compression. In *KDD '06*.
- Cheong, R. and Daniel, R. (2019). transformers.zip: Compressing transformers with pruning and quantization.
- Cohan, A., Ammar, W., van Zuylen, M., and Cady, F. (2019). Structural scaffolds for citation intent classification in scientific publications. In *NAACL-HLT*.
- Conneau, A. and Kiela, D. (2018). Senteval: An evaluation toolkit for universal sentence representations. *arXiv*, abs/1803.05449.
- Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. (2018). What you can cram into a single  $\$ \& ! \# *$  vector: Probing sentence embeddings for linguistic properties. In *ACL*.
- Dai, Z., Yang, Z., Yang, Y., Carbonell, J. G., Le, Q. V., and Salakhutdinov, R. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. In *ACL*.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding.
- Gupta, S., Shah, R., Mohit, M., Kumar, A., and Lewis, M. (2018). Semantic parsing for task oriented dialog using hierarchical representations. In *EMNLP*.
- Hinton, G. E., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.
- Huang, Z. and Wang, N. (2017). Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv*, abs/1707.01219.
- Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., and Liu, Q. (2019). TinyBERT: Distilling BERT for natural language understanding.
- Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.

- Kim, Y. and Rush, A. M. (2016). Sequence-level knowledge distillation. *arXiv*, abs/1606.07947.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *NIPS*.
- Lample, G. and Conneau, A. (2019). Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.
- Li, X., Panda, S., Liu, J., and Gao, J. (2018). Microsoft dialogue challenge: Building end-to-end task-completion dialogue systems. *arXiv*, abs/1807.11125.
- Michel, P., Levy, O., and Neubig, G. (2019). Are sixteen heads really better than one?
- Mirzadeh, S., Farajtabar, M., Li, A., and Ghasemzadeh, H. (2019). Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *CoRR*, abs/1902.03393.
- Mukherjee, S. and Awadallah, A. H. (2019). Distilling transformers into simple neural networks with unlabeled transfer data.
- Papamakarios, G. (2015). Distilling model knowledge (MSc thesis). *arXiv*, abs/1510.02437v1.
- Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training. URL [https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language\\_understanding\\_paper.pdf](https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language_understanding_paper.pdf).
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.
- Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. (2015). Fitnets: Hints for thin deep nets. In *In Proceedings of ICLR*.
- Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.
- Sau, B. B. and Balasubramanian, V. N. (2016). Deep model compression: Distilling knowledge from noisy teachers. *arXiv*, abs/1610.09650.
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.
- Sucik, S. (2019). Pruning BERT to accelerate inference. <https://blog.rasa.com/pruning-bert-to-accelerate-inference/>.
- Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.
- Tang, R. and Lin, J. (2019). Natural language generation for effective knowledge distillation.

- Tang, R., Lu, Y., Liu, L., Mou, L., Vechtomova, O., and Lin, J. (2019). Distilling task-specific knowledge from BERT into simple neural networks. *arXiv*, abs/1903.12136.
- Tenney, I., Das, D., and Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. In *ACL*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- Voorhees, E. M. and Tice, D. M. (2000). Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '00*, pages 200–207, New York, NY, USA. ACM.
- Warstadt, A., Singh, A., and Bowman, S. R. (2018). Neural network acceptability judgments. *arXiv*, abs/1805.12471.
- Yu, S., Kulkarni, N., Lee, H., and Kim, J. (2018). On-device neural language model based word prediction. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 128–131, Santa Fe, New Mexico. Association for Computational Linguistics.