24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Building a Multi-domain Neural Machine Translation Model using Knowledge Distillation

Idriss MGHABBAR[1] and **Pirashanth RATNAMOGAN** [2]

Abstract. Lack of specialized data makes building a multi-domain neural machine translation tool challenging. Although emerging literature dealing with low resource languages starts to show promising results, most state-of-the-art models used millions of sentences. Today, the majority of multi-domain adaptation techniques are based on complex and sophisticated architectures that are not adapted for real-world applications. So far, no scalable method is performing better than the simple yet effective mixed-finetuning, i.e finetuning a generic model with a mix of all specialized data and generic data. In this paper, we propose a new training pipeline where knowledge distillation and multiple specialized teachers allow us to efficiently finetune a model without adding new costs at inference time. Our experiments demonstrated that our training pipeline allows improving the performance of multi-domain translation over finetuning in configurations with 2, 3, and 4 domains by up to 2 points in BLEU.

1 Introduction

Since statistical machine translation, domain adaptation has always been a field of machine translation that has interested business and academic research. Indeed, model specialization methods now allow building models that perform better on specialized domains such as finance, marketing or science.

In this paper, we try to challenge the state of the art to solve a critical scalability problem for real-world applications: building an efficient and effective multi-domain neural machine translation model that will ensure high-quality translation in a large set of domains. Indeed, most of the companies have multiple services, such as legal, marketing or accounting, that require multiple specialized translation models. However deploying several single domain models in production is not scalable as it introduces storage costs and complexity.

Today, most of the solutions to tackle the multi-domain specialization problem suffers from a lack of effectiveness as they rely either on ensembling, reranking or prior classification of input sentences when translating. One of the key advantages of our work is that the original architecture is not altered and that there is no added complexity or parameter increase.

To do so, we leverage the well-known knowledge distillation to tackle the multi-domain neural machine translation problem. Knowledge distillation has been used as a compression technique where light models are learning to mimic cumbersome models [9]. Very recently researchers used this so-called teacher-student paradigm not only to compress models but also for training state-of-the-art sentiment analysis models [20] or multilingual neural machine translation system [24].

What we propose is to build a multi-domain student educated by multiple teachers trained to be experts in a specific domain using state-of-the-art single domain adaptation methods. Our method outperforms typical benchmarks in the domain while being as scalable and fast as a generic translation model.

Our main contribution can be separated in three parts:

- Extending multi-teacher knowledge distillation training framework to multi-domain translation task
- Using a pretrained generic student and dynamic data selection while finetuning the teachers
- Generalizing the approach for 2, 3 and 4 domains

2 Background

2.1 Neural Machine Translation

Neural Machine Translation is the field studying the use of neural networks to design automatic ways to perform translation from a language to another.

In our work, we will use the well known Seq2Seq architecture and especially the recent advances in sequence-to-sequence modeling, namely the Transformer that will be presented in Section 2.2.

The Seq2Seq architecture relies on an encoder-decoder mechanism. The very first models of such architecture [23, 2] where based on the following idea: the encoder receives an input sequence which it encodes into a hidden state vector exploited by the decoder to produce the output. In neural machine translation, a source sentence is encoded and then decoded to produce a valid translation.

2.1.1 Encoder

Given an input sequence $\mathbf{x} = (x_1, x_2, ..., x_{L_x})$ and an output sequence $\mathbf{y} = (y_1, y_2, ..., y_{L_y})$, the hidden states are computed as following :

$$h_t^{enc} = f^{enc}(W_{hh}^{enc} h_{t-1}^{enc} + W_{hx}^{enc} x_t + b_h^{enc}) \qquad (1)$$

where $t \in 1, ..., L_x$, and f^{enc} is the encoding function that depends on the chosen architecture.

[1] BNP Paribas, France, email: idriss.mg@gmail.com
[2] BNP Paribas, France, email: pirashanth.ratnamogan@bnpparibas.com

2.1.2 Decoder

On the decoder side, the initialization of the hidden states may be done using the encoder final hidden state, hence $h_0^{dec} = h_{L_x}^{enc}$, or randomly. Then the decoder's hidden states computation is as follows:
If teacher forcing [14] is enabled, i.e feeding the network with the ground truth:

$$h_t^{dec} = g^{dec}(W_{hh}^{dec} h_{t-1}^{dec} + W_{hx}^{dec} y_{t-1} + b_h^{dec}) \tag{2}$$

$$h_t^{dec} = g^{dec}(h_{t-1}^{dec}, y_{t-1}) \tag{3}$$

If teacher forcing is disabled, i.e feeding the network with the predicted value at the previous step:

$$h_t^{dec} = g^{dec}(W_{hh}^{dec} h_{t-1}^{dec} + W_{hx}^{dec} y_{t-1}^{pred} + b_h^{dec}) \tag{4}$$

Eventually the prediction is :

$$y_t^{pred} = z^{dec}(W_{hy}^{dec} h_t^{dec} + b_y^{dec}) \tag{5}$$

where t ∈ 1,...,L_y.

2.2 Recurrent Neural Network and Attention Mechanism

2.2.1 Recurrent Neural Networks

Presented computations for encoder and decoder stands for the simplest and the most standard form of recurrent networks. Recurrent Neural Networks, are now a standard architecture that allows dealing with sequential data, as previous output are fed as input to the following computations. It has been improved in order to avoid vanishing and exploding gradient problem through the development of adapted architectures such as LSTM or GRU [10, 2].

2.2.2 Attention Mechanism

Later, encoding into a single vector a sentence was found limiting, especially for long sentence. [1, 16] introduces Attention based Sequence to Sequence networks allowing decoder to consider not only the encoder's last hidden state but all the encoder's hidden states. Attention Networks, are now at the basis of most state-of-the-art neural machine translation architectures. Attention mechanism, makes use of a context vector, that is a weighted average of the encoder's hidden states. Those attention weights are computed using a score that depends on current or previous hidden state and that is learned during the training step.

2.3 The Transformer Network

Until [27], Standard state-of-the-art Seq2Seq models were all based on recurrent neural networks or convolutional neural networks given the sequential nature of the data. However, handling sequences element by element sequentially is an obstacle for parallelization.

The novelty of the Transformer architecture is its replacement of sequential computations by attentions and positional encoding to keep track of the element's position in the sequence. This enabled to accelerate the training time and to reduce the complexity of computing dependency between elements independently of their position, while recurrent models were obliged to pass through all the intermediate elements.

Transformer is now the leading architecture that helped getting state-of-the-art results in most neural machine translation and natural language processing tasks.

2.4 Transfer learning

Transfer learning is a key topic in natural language processing as it led to state-of-art results [15, 5, 11].

It relies on the assumption that pre-training neural networks on a large set of data in a task $\mathcal{A}$ will help initialize a network trained on a second task $\mathcal{B}$ where data is scarce.

The insufficiency of in-domain data has led researchers to train well performing generic models before adapting them on the target domain using in-domain data.

2.5 Knowledge distillation

In machine learning, we accept the idea that the objective function is made to reflect the interest of the user as closely as possible. However, all the algorithms tend to minimize the cost function on the training data while the real interest is to generalize well on new data. It is clearly better to train models to generalize better but this requires knowing how to do so.

When we are distilling the knowledge from a large model to a small model or from a specialist to a generalist, we can train the student to generalize in the same way than the teacher. In general, the teacher is well suited to transfer this kind of information as it is a cumbersome model.

The objective of knowledge distillation is to fill the gap between the interest of the user, which is good generalization on unseen data, and the cost function used during training. One way to transfer the generalization ability of the cumbersome model is to use class probabilities as soft targets. Instead of trying to match the ground truth labels, we will perform optimization on the softened targets provided by the teacher.

In other terms, in classification, the negative log-likelihood cost function will be replaced by the Kullback-Leibler divergence between the teacher's distribution and the student distribution.

Traditionally in neural machine translation, the loss function for one sentence is:

$$l = -\sum_{j=1}^{sent\_length} \sum_{k=1}^{|V|} 1[y_j = k] log(p(y_j = k|x, y_{<j})) \tag{6}$$

where x is the source sentence and $y = (y_1, y_2, \ldots, y_{L_y})$ is the target sentence.

Replacing the cost function by the KL-divergence would result in:

$$l = -\sum_{j=1}^{sent\_length} \sum_{k=1}^{|V|} p^T(y_j = k|x, y_{<j}) \\ log(p^S(y_j = k|x, y_{<j})) \tag{7}$$

where p^T and p^S are the probability distributions of the teacher and the student respectively.

This is considered as the simplest form of distillation that can work without having true labels for the transfer set, which can be the teacher's training set. When the correct labels are known for this transfer set or a subset of it, we can incorporate this information to make use of the added information and train the model to produce the correct labels.

The combination method used in our case result in this cost function which is a weighted sum between the traditional negative log-likelihood and the Kullback-Leibler divergence.

$$l = - \sum_{j=1}^{sent\_length} \sum_{k=1}^{|V|} ((1-\lambda)1[y_j = k] + \lambda p^T(y_j = k|x, y_{<j})) \\ log(p^S(y_j = k|x, y_{<j})) \quad (8)$$

where λ is a hyper-parameter of the method quantifying the softening of the targets. If λ = 0, we get the cross-entropy loss and if λ = 1, we get the simplest form of distillation described above where the ground truth labels are not exploited.

Knowledge distillation as a compression technique. Knowledge distillation enables to transfer knowledge between a teacher and a student without any size constraint as only class probabilities are used in the cost function. The class probabilities compatibility depend only on the vocabulary which means that the teacher and the student must share the same vocabulary and not the size (number of parameters of both models).

Therefore, a cumbersome model having hundreds of millions of parameters can educate a simpler student having tens of millions of parameters and producing a similar prediction quality.

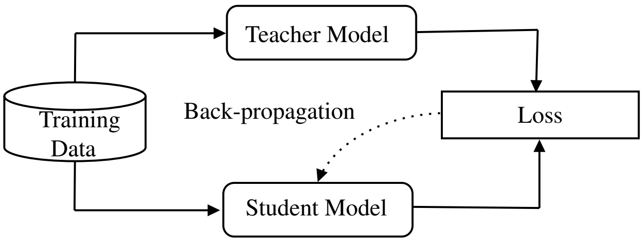

Figure 1: How compression works through distillation?

In [9], Geoffrey Hinton et al. investigated the effects of ensembling Deep Neural Network acoustic models that are used in Automatic Speech Recognition. They show it is possible to distill the knowledge of an ensemble of 10 models to a single model achieving similar Word Error Rate (WER) having ten times less parameters.

3 Related Work

Domain adaptation for machine translation has been largely explored.

In a survey summarizing the different developed methods [3], it is observed that most of the multi-domain adaptation techniques are based on new architectures with additional domain classifiers which reduces their scalability compared to the standard model, and only a few try to leverage data or the training objective.

The majority of the articles study adaptation **on a single domain**. Scalable methods are essentially based on finetuning using either in-domain data only or a mix of out-of-domain and in-domain data in order to reduce overfitting [22, 17, 7, 26]. Other methods try to leverage the loss function by introducing a weighting strategy [28].

While multiple articles tackle the **multi-domain adaption problem** with promising results, most of them rely either on ensembling [7], a priori domain clustering in order to add domain tags [25] or introducing a new domain specific gating vector [29]. So far, the best technique that does not add more complexity or prior classification, either using supervised or unsupervised methods, is based on fine-tuning on the concatenation of all in-domain data [21] [3].

Knowledge distillation is a well-known neural network compression technique, especially for machine translation [9, 12, 8]. Recently, [20] showed that relying on multiple experts teachers allowed to improve the performances of sentiment analysis on unseen domains. In [24], the authors applied the multiple teachers framework in knowledge distillation in order to build state-of-the-art multilingual machine translation system. This work applied to multilingual machine translation is the closest to the one that we propose: it uses a similar methodology to the one we explored but it fulfills a different goal. Highly motivated by those results, our goal was to apply the multiple teachers knowledge distillation framework in order to improve the performances on multi-domain neural machine translation. So far, to our knowledge, the only usage of knowledge distillation in the context of neural machine translation adaptation is limiting the performance degradation on the out-of-domain data [4].

4 Approach

Neural Machine Translation models are often trained on large open-source data coming from institutions such as the European parliament, the European commission, movies subtitles, Wikipedia pages, etc. These corpora offer us the necessary amount of data to train neural networks having hundreds of millions of parameters. However these models are often not well performing on specialized datasets and therefore transfer learning may be used.

In our approach, we use *Dynamic Data Selection*, a transfer learning technique, to train models on specialized datasets that will play a teacher role when building a multi-domain student through *a multi-task knowledge distillation strategy*.

4.1 Knowledge distillation as a multi-task technique

Initially, we presented the distillation strategy as a way to perform compression on neural networks. This has been proved to reduce the complexity while preserving the performance of the cumbersome network. In our work, we used knowledge distillation to produce a multi-domain student that may or may not be of the same size than the teachers.

This strategy enables us to use a single model to translate sentences from different domains instead of storing different expert models, filtering the data during inference and using the corresponding teacher. It is time and memory efficient and particularly convenient in a production environment.

The use of specialists that are trained on different domains has some resemblance to mixture of experts which learns how to assign each example to the most likely expert through probability computation. During training, two learning phases happen simultaneously, the experts are learning how to be accurate on the examples assigned to them and the gating network is learning how to assign examples to the corresponding experts.

The major difference between the two methods is parallelization. It is easier to train at the same time different teachers on their corresponding domain than to parallelize mixtures of experts.

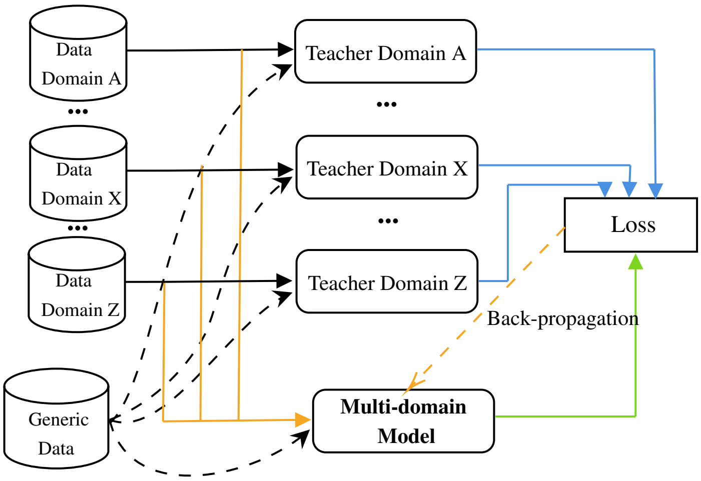

Figure 2: Training a multi-domain student using multiple single domain teachers

4.2 Dynamic Data Selection

Here we present a transfer learning technique called *Dynamic data selection* developed by Marlies van der Wees et al. in [26] used to build the teachers during the distillation process.

The idea is to select sentences from generic data that are similar to the in-domain corpus and at the same time dissimilar to the generic corpus. The metric used for the ranking is based on the cross-entropy.

Let I be the in-domain data, G the generic data. We assume that G contains a subset G_I that has the same distribution than I.

The objective of the method is to select sentences s from G that are most probably belonging to G_I, i.e. maximizing $P(G_I|s, G)$.

$$P(G_I|s, G) = \frac{P(s|I)P(G_I|G)}{P(s|G)} \tag{9}$$

This quantity, log-transformed, is close to : $H_I(s) - H_G(s)$.

The entropy $H_A(b)$ is defined as: $H_A(b) = E_{x \sim P}(-logQ(b))$, P being the language model computed on the corpus A, Q the estimated language model and b the sentence of interest.

In order to perform data selection on a bi-text corpus, a ranking metric is introduced:

$$\begin{aligned} CED_s = {} & [H_I(s_{source}) - H_G(s_{source})] \\ & + [H_I(s_{target}) - H_G(s_{target})] \end{aligned} \tag{10}$$

For example, when calculating $H_I(s_{source})$, we build a language model on the in-domain corpus then we use the **score** function to rank sentences. The output is the log-transformed product of each word's probability.

During gradual fine-tuning, the selection size n is a function of the epoch i :

$$n(i) = \alpha|G|\beta^{\frac{i-1}{\nu}} \tag{11}$$

where α is the relative start size i.e. the fraction of the out-of-domain data for the first selection, G is the size of the latter, $0 \leq \beta \leq 1$ is the retention rate i.e. the fraction to be kept at each selection, i the epoch, ν the number of epochs where the selected subset doesn't change.

This finetuning approach has been proven to provide state-of-the-art result in the context of domain specialization (especially when the set of in-domain parallel sentences is small) while significantly reducing the training time. This method have been used in order to finetune our expert teachers on single domains.

4.3 Distillation process

The training that we propose for distillation process is described in **Algorithm 1**.

Algorithm 1 Multi-domain student training pipeline

```
Input: Generic dataset $\mathcal{G}$, $n$ different specialiazied datasets $(\mathcal{D}_i)_{i \in \{1..n\}}$
Output: Multi-domain student model $S_{multi}$ trained using knowledge distillation
Initialization: Transformer model $M_{gen}$ initialized with random weights.
Empty array $\mathcal{P}_T$ of size (T,L,$|V|$) where T is the sentence length, L the training set size and $|V|$ the vocabulary size.
1. Train generic model:
  while $M_{gen}$ not converging do
    Train $M_{gen}$ with batches from $\mathcal{G}$
  end while
2. Finetune teachers:
  for $i$ in $\{1..n\}$ do
    Initialize $\mathcal{T}_i$ = $M_{gen}$
    Finetune $\mathcal{T}_i$ using dynamic data selection on specialized dataset $\mathcal{D}_i$ and generic data $\mathcal{G}$.
  end for
3. Extend $\mathcal{P}_T$:
  for $i$ in $\{1..n\}$ do
    for $d$ in $\mathcal{D}_i$ do
      Compute $\mathcal{P}_{T_i}(d_{src})$, the probability distribution resulting from $\mathcal{T}_i$'s translation of $d_{src}$.
      Append $\mathcal{P}_{T_i}(d_{src})$ to $\mathcal{P}_T$
    end for
  end for
4. Train student model:
Let $\mathcal{G}_{sub}$ be a randomly selected subset from the generic dataset.
  while $S_{multi}$ not converging do
    Train $S_{multi}$ with batches from $(\cup_{i \in \{1..n\}} \mathcal{D}_i) \cap \mathcal{G}_{sub}$ using $\mathcal{P}_T$ in the cost function.
  end while
```

4.4 Top-K distillation

In our experiments, the student model does not match the full distribution of the teacher model but only the top-K output distribution in order to reduce memory cost. Actually, it is expensive in terms of RAM memory to load the full distributions of the different teacher models knowing that we may have a large number of teachers.

4.5 Word-Level knowledge distillation

In Neural Machine Translation, two kind of distillation processes have been explored so far. The first one, word-level is based on training the student to mimick teacher's local word distributions (using the ground truth target sequence) while the second one, sequence-level is based on training the student to mimick the teacher's output after beam search (without using the ground truth target sequence).

While [12] have proved sequence level knowledge distillation efficiency for neural network compression. [24] proved that sequence level knowledge distillation when used in the multi-teacher framework results in significantly lower performances than word-level distillation.

Hence, all our experiments are based on word-level knowledge distillation only.

4.6 Label smoothing impact on distillation

In [18], the authors investigate the consequences of using label smoothing, when training teachers, on the student performance. They've shown that using label smoothing does not necessarily lead to better distillation.

They assume that this shortcoming is linked to the information erasure caused by label smoothing. A visualization technique applied to the penultimate layer representations of image classifiers trained on image datasets such as CIFAR10 enabled to show that using label smoothing results in a loss of information concerning similarities between examples of different classes.

5 Experiments and Results

5.1 Experimental setup

5.1.1 *Datasets and preprocessing*

In order to assess our method's efficiency we focused on the English to French multi-domain translation. First we start by building the initial **generic** model used for the transfer learning tasks. The initial generic model was trained using WMT14 data preprocessed using a standard procedure: 40k operations based BPE joint-tokenization (source and target sentences are sharing the same BPE tokenization), filtering sentences longer than 250 tokens and sentences with a ratio between source sentence and target sentence length higher than 1.5 [6].

Domain **specific** data will focus on 4 domains: Medical, Legal, Software documentation, and religion. Indeed, the experiments will be conducted on open-source data: EMEA[3] [Medical] (European Medicines Agency documents), JRC[4] [Legal] (a collection of legislative text of the European Union), GNOME [5] [Software] and BIBLE [6] [Religion] corpora . Specialized data is processed in the same way as the generic one. All the datasets are splitted in a training and a testing part which is the first 2000 sentences of the corpus.

5.1.2 *Hyperparameters and settings*

During the experiment we trained a Transformer Base network implemented in OpenNMT-py framework [13], using Adam optimizer ($\beta_1 = 0.9, \beta_2 = 0.98$), label smoothing and dropout equal to 0.1, with noam decay and an initial learning rate equal to 2.

The generic model was trained during 130k steps on 4 V100 GPUs, with a token-based batch size of 4096 and gradient accumulation during 2 steps (equivalent to 8 V100 training). Following common postprocessing we averaged model checkpoints during steps 110k, 120k, and 130k in order to get the final model.

In order to build the teachers used during knowledge distillation, this pretrained model was finetuned for 10k steps on each specialized dataset. Checkpoints were saved every 1000 steps and are used to get the best model according to the criterion described in the Results section.

This process generated four expert teachers for Medical, Legal, Software and Religion datasets.

5.2 Results

5.2.1 *Distillation strategies*

In our first experiment, we compare the performance according to three different strategies : no distillation ($\lambda = 0$), pure distillation ($\lambda = 1$), and flexible distillation ($0 < \lambda < 1$). You can notice that the first strategy corresponds to finetuning.

For the flexible distillation, we use greedy search to find the best value for the parameter λ combining the negative log-likelihood and the Kulback-Leibler divergence. $\lambda = 0.7$ was found to yield the best results.

The performances are evaluated using the BLEU score implemented in *sacrebleu* [19].

Table 1: Performance of the teachers, trained without label smoothing, on their corresponding test set

Teacher	**Domain test set**	**WMT14 test**
Teacher Legal	58.8	35.2
Teacher Medical	58.7	33.4
Teacher Software	37.6	30.5
Teacher Religion	27.8	25.4

Table 2: BLEU scores corresponding to four configurations with 2 to 4 domains using three different distillation strategies

Configurations	λ	**Legal**	**Medical**	**Software**	**Religion**	**WMT14**
Initial Generic Model	-	51.0	33.3	24.6	11.5	38
Legal Medical	0	57.4	50.0	-	-	34.4
	0.7	**58.4**	53.0			**35.2**
	1	58	**54.1**			33.7
Legal - Medical Software	0	57.0	50.0	**40.5**	-	35.1
	0.7	**57.9**	52.7	40.2		**35.4**
	1	57.6	**53.1**	39.3		33.6
Legal - Medical Software - Religion	0	56.4	48.8	37.5	24.4	**34.9**
	0.7	**56.7**	51.4	37.7	27.3	34.0
	1	**56.7**	**52.1**	**38.7**	**27.9**	29.4

The scores shown in Table 2 correspond to the best step for each configuration. The ranking criterion is the average BLEU score over the specialized datasets available in the configuration.

To conduct our experiments, we decided to consider datasets of different difficulties where the most difficult to translate would be in the last configuration. You can see in Table 1 that Legal and Medical, studied in the first configuration, are the easiest ones to translate

[3] http://opus.nlpl.eu/EMEA.php
[4] http://opus.nlpl.eu/JRC-Acquis.php
[5] http://opus.nlpl.eu/GNOME.php
[6] http://opus.nlpl.eu/bible-uedin.php

(BLEU score of 58.8 and 58.7 respectively) whereas the Software dataset, studied in the configuration that follows, is more challenging (BLEU score of 37.6) and eventually the Religion dataset introduced in the last configuration corresponds to the lowest performance (BLEU score of 27.8).

The first experiment showed that the pure distillation strategy ($\lambda = 1$) yields the best results on the specialized datasets. However, we see that the BLEU score on the generic dataset WMT14 decreased substantially. In fact, in the last configuration when using $\lambda = 1$, a decrease of 4.6 points in BLEU score is observed on the WMT14 test set compared to the model using $\lambda = 0.7$.

As we want to avoid overfitting on the specialized datasets, we choose the model using $\lambda = 0.7$ to conduct the next experiment.

5.2.2 *Impact of label smoothing*

In the following experiment we compare three models: the student using teachers trained with hard targets, the student using teachers trained with soft targets and, finetuned model (literature optimal benchmark [21]) and the ensembled model.

The results are summarized similarly to the previous experiment where for each configuration the BLEU scores are computed on the corresponding test sets using the following models: Hard student (**HS**), Soft student (**SS**), Ensembling (**Ensemble**), and Finetuned (**F**). Ensembling relies on ensembling all the involved single domain teachers.

Moreover, we show in **Figure 3** the evolution of the BLEU score on the different test sets for the last configuration where 4 teachers are built.

Table 3: Performance of the teachers, trained with label smoothing, on their corresponding test set.

Teacher	Domain test set	WMT14 test
Soft teacher Legal	58.8	35.9
Soft teacher Medical	53.5	35.7
Soft teacher Software	39.3	29.1
Soft teacher Religion	27.4	26.8

If we compare the results shown in Table 3 with those in Table 1, we notice that teachers trained without label smoothing tend to perform better than the ones trained with label smoothing on their specific domain but the opposite is observed on the generic domain. This confirms the assumption that label smoothing results in better generalization.

Before introducing a metric that will help us evaluate the label smoothing impact, we analyze the results on each dataset.

- On **WMT14** all methods result in a decrease of the BLEU score compared to the initial generic model with **38** BLEU (39.9 with multibleu) as the training goal is to optimize the performance on in-domain test sets. The decrease amount is approximately similar for all three specialization methods. Ensembling, is generally performing best on this set, but with the a lot more additional complexity and a significant decrease on in-domain data.
- On **Legal**, the generic model already provides a quite high BLEU score : 51. Indeed, the Legal dataset that consists of a collection of legislative documents is quite similar to the training data of the initial generic model. Even if original BLEU score was high, domain adaptation allows to get even better scores. The HS model outperforms the other models on this domain in all configurations.

Table 4: BLEU scores corresponding to four configurations with 2 to 4 domains using the four different strategies: knowledge distillation with hard teachers, knowledge distillation with soft teachers, mixed finetuning and ensembling.

Configurations	Model	Legal	Medical	Software	Religion	WMT14
Initial Generic Model	-	51.0	33.3	24.6	11.5	38
Legal Medical	HS	**58.4**	**53.0**	-	-	35.2
	SS	58.0	52.0			35.2
	F	57.4	50.0			34.4
	Ensemble	57.1	41.1			**36.9**
Legal - Medical Software	HS	**57.9**	**52.7**	40.2	-	35.4
	SS	57.7	51.4	**40.6**		35.4
	F	57.0	50.0	40.5		35.1
	Ensemble	54.6	36.6	28.6		**35.9**
Legal - Medical Software - Religion	HS	**56.7**	**51.4**	37.7	**27.3**	34.0
	SS	56.5	50.2	**38.8**	26.8	34.0
	F	56.4	48.8	37.5	24.4	34.9
	Ensemble	54.1	34.5	25.4	15.8	**35.8**

- On **Medical**, domain adaptation allows to get significant improvement over the initial generic model (up to 20 points in BLEU) as this medical corpus is different from WMT14 data on which the initial generic model was trained. The HS model yields the highest BLEU score in all configurations.
- On **Software**, the improvement is also significant over the initial generic model. The GNOME dataset, added in the configuration with 3 domains, contains short and specific sentences. As it is less specific than Legal and Medical, all three methods perform similarly.
- On **Religion**, the initial generic model results in a low BLEU score. Indeed, the BIBLE dataset contains sentences written in old English/French. On this complex specialization set, the knowledge distillation strategies outperform the mixed finetuning method (gain of 2.65 BLEU in average).

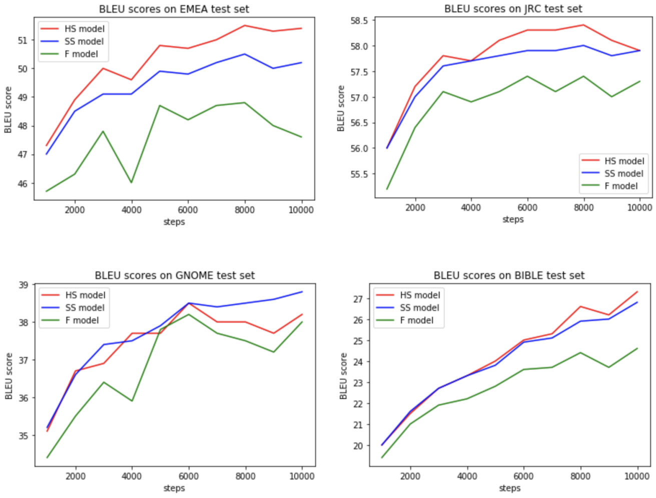

Figure 3: Evolution of the BLEU score for the configuration: 4 teachers

We now define the metric Δ as the average gain over the finetuned model in terms of the BLEU score.

Let Δ_{HS} and Δ_{SS} correspond respectively to the Δ metric computed using the HS model and the SS model. They are defined as following:

$$\begin{cases} \Delta_{HS} = avg_{datasets}(BLEU_{HS} - BLEU_F) \\ \Delta_{SS} = avg_{datasets}(BLEU_{SS} - BLEU_F) \end{cases} \tag{12}$$

We compute both metrics using the results presented in Table 4 :

Table 5: Average BLEU gap between knowledge distillation based and finetuning based trained multi-domain models

Configuration	Δ_{HS}	Δ_{SS}
2 teachers	2.0	1.3
3 teachers	1.1	0.73
4 teachers	1.55	1.45

Conclusion.

Our experiments showed that knowledge distillation outperforms finetuning and ensembling in all the tested configurations and confirmed the impact of label smoothing on the distillation process as the student trained using hard teachers performs better than the one that was trained using soft teachers. It also showed that our intuitions about the increasing nature of Δ_{HS} and Δ_{SS} aren't true. To our opinion, the distillation strategy is sensitive to the choice of λ, as it sets up the combination between the negative log-likelihood and the Kullback-Leibler divergence, therefore using the same value of λ for all the teachers may have limited the potential of this strategy.

6 Conclusions and Future Directions

In our work, we showed that knowledge distillation enables to gather the expertise of multiple teachers in one student reducing memory cost and inference time.

Indeed, we have presented an approach to building a single multi-domain model model incorporating the cognition of multiple single-domain experts with knowledge distillation. We also introduced a state-of-the art method to finetune models on small specialized datasets. Experimental results on English-French translations tasks on Medical, Legal, Software and Religion specialized datasets demonstrate the capability of our approach to outperform finetuning methods while being as scalable and effective as a generic translation model.

In future work, we plan to explore building a multilingual multi-domain model by distilling multiple unilingual multi-domain models trained using our approach.

This approach may easily be applied to other domains such as speech-to-text and computer vision as the knowledge transfer is only made through probability distributions.

7 Acknowledgements

The authors would also like to thank Mr. **Pierre Bertrand**, Mr. **Bruce Delattre**, and Mr. **Geoffrey Scoutheeten** of BNP Paribas for their valuable comments and suggestions. The work is supported by the Data & AI Lab of BNP Paribas.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014. cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.

[2] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, 'Learning phrase representations using RNN encoder–decoder for statistical machine translation', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, Doha, Qatar, (October 2014). Association for Computational Linguistics.

[3] Chenhui Chu and Rui Wang, 'A survey of domain adaptation for neural machine translation', *CoRR*, **abs/1806.00258**, (2018).

[4] Praveen Dakwale and Christof Monz, 'Fine-tuning for neural machine translation with limited degradation across in-and out-of-domain data', *Proceedings of the XVI Machine Translation Summit*, 117, (2017).

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 'BERT: pre-training of deep bidirectional transformers for language understanding', *CoRR*, **abs/1810.04805**, (2018).

[6] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier, 'Understanding back-translation at scale', *CoRR*, **abs/1808.09381**, (2018).

[7] Markus Freitag and Yaser Al-Onaizan, 'Fast domain adaptation for neural machine translation', *CoRR*, **abs/1612.06897**, (2016).

[8] Markus Freitag, Yaser Al-Onaizan, and Baskaran Sankaran, 'Ensemble distillation for neural machine translation', *CoRR*, **abs/1702.01802**, (2017).

[9] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean, 'Distilling the knowledge in a neural network', in *NIPS Deep Learning and Representation Learning Workshop*, (2015).

[10] Sepp Hochreiter and Jürgen Schmidhuber, 'Long short-term memory', *Neural computation*, **9**(8), 1735–1780, (1997).

[11] Jeremy Howard and Sebastian Ruder, 'Fine-tuned language models for text classification', *CoRR*, **abs/1801.06146**, (2018).

[12] Yoon Kim and Alexander M. Rush, 'Sequence-level knowledge distillation', in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, (November 2016). Association for Computational Linguistics.

[13] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush, 'Opennmt: Open-source toolkit for neural machine translation', *CoRR*, **abs/1701.02810**, (2017).

[14] Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio, 'Professor forcing: A new algorithm for training recurrent networks', in *Advances in Neural Information Processing Systems 29*, eds., D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, 4601–4609, Curran Associates, Inc., (2016).

[15] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, 'Roberta: A robustly optimized BERT pretraining approach', *CoRR*, **abs/1907.11692**, (2019).

[16] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning, 'Effective approaches to attention-based neural machine translation', *CoRR*, **abs/1508.04025**, (2015).

[17] Thang Luong, Hieu Pham, and Christopher D. Manning, 'Effective approaches to attention-based neural machine translation', in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, Lisbon, Portugal, (September 2015). Association for Computational Linguistics.

[18] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton, 'When does label smoothing help?', *CoRR*, **abs/1906.02629**, (2019).

[19] Matt Post, 'A call for clarity in reporting BLEU scores', in *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Belgium, Brussels, (October 2018). Association for Computational Linguistics.

[20] Sebastian Ruder, Parsa Ghaffari, and John G. Breslin, 'Knowledge adaptation: Teaching to adapt', *CoRR*, **abs/1702.02052**, (2017).

[21] Hassan Sajjad, Nadir Durrani, Fahim Dalvi, Yonatan Belinkov, and Stephan Vogel, 'Neural machine translation training in a multi-domain scenario', *CoRR*, **abs/1708.08712**, (2017).

[22] Christophe Servan, Josep Maria Crego, and Jean Senellart, 'Domain specialization: a post-training domain adaptation for neural machine translation', *CoRR*, **abs/1612.06141**, (2016).

[23] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, 'Sequence to sequence learning with neural networks', *CoRR*, **abs/1409.3215**, (2014).

[24] Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu, 'Multilingual neural machine translation with knowledge distillation', in *International Conference on Learning Representations*, (2019).

[25] Sander Tars and Mark Fishel, 'Multi-domain neural machine translation', *CoRR*, **abs/1805.02282**, (2018).

[26] Marlies van der Wees, Arianna Bisazza, and Christof Monz, 'Dynamic data selection for neural machine translation', in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1400–1410, Copenhagen, Denmark, (September 2017). Association for Computational Linguistics.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin, 'Attention is all you need', in *Advances in Neural Information Processing Systems 30*, eds., I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 5998–6008, Curran Associates, Inc., (2017).

[28] Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita, 'Instance weighting for neural machine translation domain adaptation', in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1482–1488, Copenhagen, Denmark, (September 2017). Association for Computational Linguistics.

[29] Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao, 'Multi-domain neural machine translation with word-level domain context discrimination', in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 447–457, Brussels, Belgium, (October-November 2018). Association for Computational Linguistics.