From Basics to Breakthroughs: A Deep Dive into Knowledge Distillation in Neural Machine Translation

Joseph Attieh

Who am I?

Education

- Bachelor in Computer Engineering at Lebanese American University
- Double Master Degree in Computer Science, Communication Systems and Machine learning from Aalto University and KTH

Experience

- Interned at EPFL Lausanne, BMW Munich, Inmind.AI/UN-ESCWA Beirut, and Huawei Technologies Oy. Helsinki
- Worked for a year as a NLP Researcher at Huawei Technologies Oy., Helsinki

Currently

- PhD student at University in Helsinki working on Modularization and Knowledge Distillation for the **GreenNLP project**

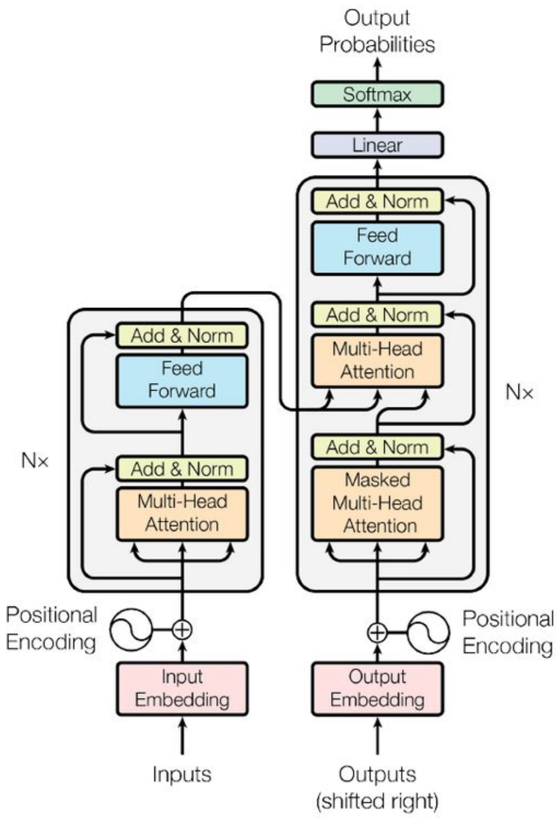
Neural Machine Translation (NMT)

- Having some pairs of source and target sentences (s_i, t_i) , we want the NMT model to learn a probability distribution $p_\theta(t|s)$
- The model predicts the most probable **target** sentence given **source**:

$$argmax_{t \in T}\ p_\theta(t|s;\theta)$$

Components of Basic NMT

- **Encoder-Decoder Architecture**
- **CE/NLL Loss** compares the model's predicted probability distribution with the true distribution (1-hot vector)



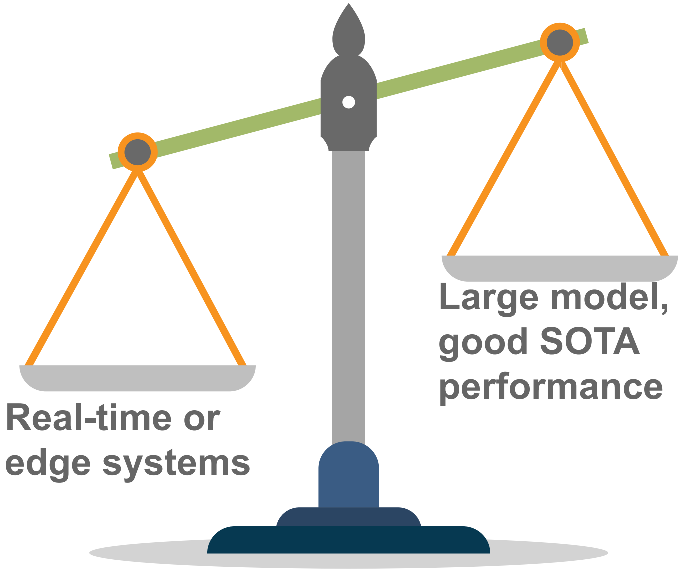
(Vaswani et al., 2017)

Challenges of NMT

- The best results are usually achieved with **Ensemble Models** or **Large Networks.**

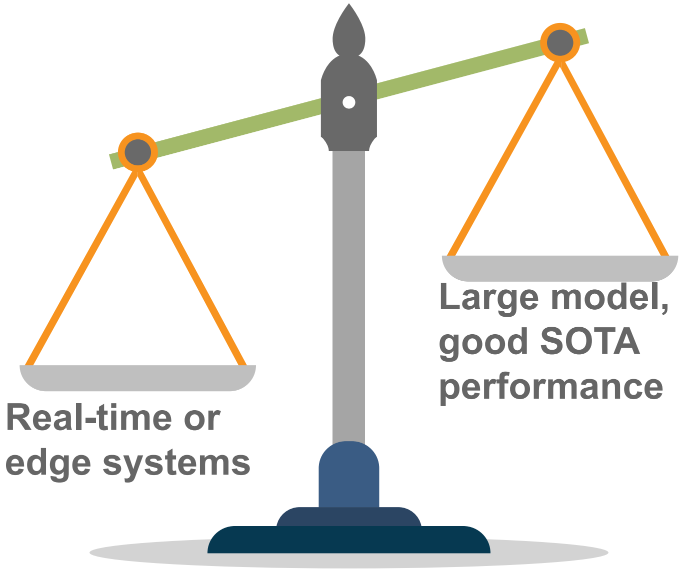
- Deploying large models on edge devices is challenging due to limited computational resources.

- Assumption
  Time and cost of running inference a model is more important than the time and memory of training a model



Challenges of NMT

- The best results are usually achieved with **Ensemble Models** or **Large Networks.**
- Deploying large models on edge devices is challenging due to limited computational resources.
- <u>Assumption</u>
  Time and cost of running inference a model is more important than the time and memory of training a model
- Need to compress the large models
  - **Knowledge Distillation**



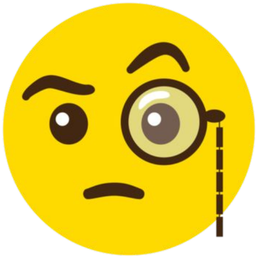
What This Presentation Is About and Is *Not* About

- **Goal:** Provide an overview of the key knowledge distillation methods for Machine Translation

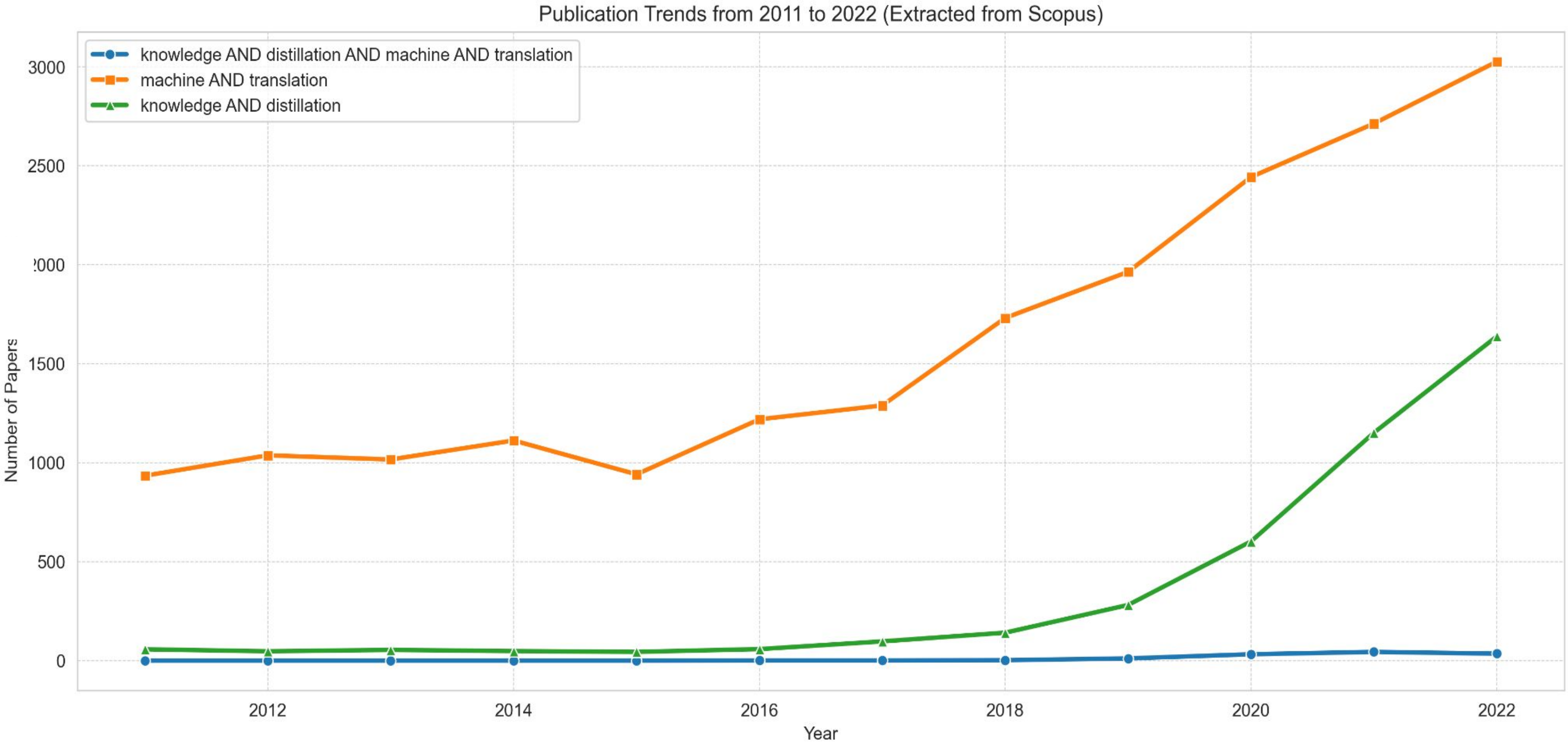
- **What this is not:** Exhaustive

  It's impossible to cover all related papers in one presentation

- **What we do cover:**
  Knowledge distillation explicitly applied on Autoregressive NMT models



Papers Trend: NMT↗, KD↗, KD for NMT🥴

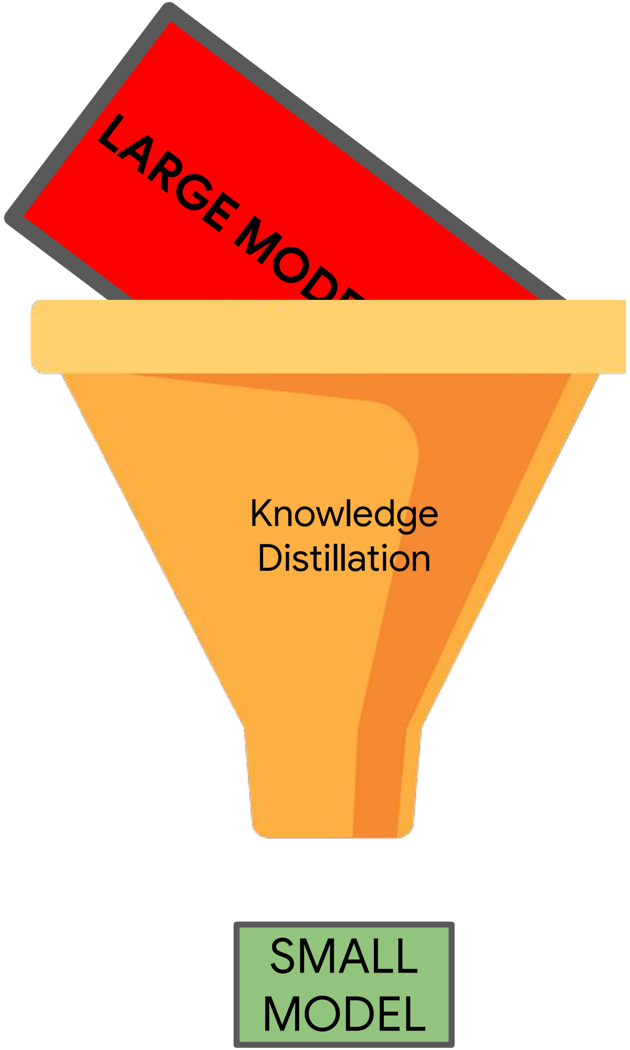


What is Knowledge Distillation?

What is Knowledge Distillation?

- Transferring the knowledge from a (set of) **large** model(s) to a **smaller** model w/o significant loss in performance.

- The small model is a **student** that learns from the large **teacher** model by imitating the teacher predictions.



How is KD performed for NMT models?

How is Knowledge Distillation performed for NMT models?

- Auto-regressive Negative Log-Likelihood (NLL) Loss:

$$L_{NLL} = -\sum_{j=1}^{|J|}\sum_{k=1}^{|V|} \mathbb{1}\{t_j = k\}\, log\, p_\theta(t_j = k|s, t_{<j})$$

compares the model's predicted probability distribution with the true distribution

* V is target vocabulary set

How is Knowledge Distillation performed for NMT models?

- Auto-regressive Negative Log-Likelihood (NLL) Loss

$$L_{NLL} = -\sum_{j=1}^{|J|}\sum_{k=1}^{|V|} \mathbb{1}\{t_j = k\} \, log \; p_\theta(t_j = k|s, t_{<j})$$

- Having access to a teacher distribution

$$L_{WORD-KD} = -\sum_{j=1}^{|J|}\sum_{k=1}^{|V|} q(t_j = k \,|s, t_{<j}) \; log \; p_\theta(t_j = k|s, t_{<j})$$

compares the student predicted probability distribution with the teacher's (~data distr)

Word-Level Knowledge Distillation (Kim & Rush, 2016)

- Auto-regressive Negative Log-Likelihood (NLL) Loss

$$L_{NLL} = -\sum_{j=1}^{|J|}\sum_{k=1}^{|V|} \mathbb{1}\{t_j = k\}\, log\; p_\theta(t_j = k|s,\, t_{<j})$$

- Having access to a teacher distribution

$$L_{WORD-KD} = -\sum_{j=1}^{|J|}\sum_{k=1}^{|V|} q(t_j = k\,|s,\, t_{<j})\; log\; p_\theta(t_j = k|s,\, t_{<j})$$

- Interpolate these two losses to take ground truth labels into account

$$L = (1-\alpha)L_{NLL} + \alpha\, L_{KD}$$

Sequence-Level Knowledge Distillation (Kim & Rush, 2016)

Sequence-Level Knowledge Distillation (Kim & Rush, 2016)

- Instead of minimizing word-level CE, minimize CE between sequence distributions

Sequence-Level Knowledge Distillation (Kim & Rush, 2016)

- Instead of minimizing word-level CE, minimize CE between sequence distributions

- The sequence-level NLL for NMT involves matching the 1-hot distribution over all complete sequences:

$$L_{NLL} = -\sum_{k=1}^{|V|} \mathbb{1}\{t = y\}\, log\, p_{\theta}(t \mid s)$$

Sequence-Level Knowledge Distillation (Kim & Rush, 2016)

- Instead of minimizing word-level CE, minimize CE between sequence distributions

- The sequence-level NLL for NMT involves matching the 1-hot distribution over all complete sequences:

$$L_{NLL} = -\sum_{k=1}^{|V|} \mathbb{1}\{t = y\}\ log\, p_\theta(t \mid s)$$

$$L_{SEQ-KD} = -\sum_{k=1}^{|V|} q(t|s)\ log\, p_\theta(t|s)$$

The teacher's sequence distribution over the sample space of all possible sequences

Sequence-Level Knowledge Distillation (Kim & Rush, 2016)

- Instead of minimizing word-level CE, minimize CE between sequence distributions

- The sequence-level NLL for NMT involves matching the 1-hot distribution over all complete sequences:

$$L_{NLL} = -\sum_{k=1}^{|V|} \mathbb{1}\{t = y\} \, log \, p_\theta(t \mid s)$$

$$L_{SEQ-KD} = -\sum_{k=1}^{|V|} q(t|s) \, log \, p_\theta(t|s)$$

The teacher's sequence distribution over the sample space of all possible sequences

- The authors approximate the teacher distribution by:
    - Replacing it by its mode
    - Replacing by the results of a beam search on the teacher

Sequence-Level Knowledge Distillation (Kim & Rush, 2016)

1) Run beam search over the training set with the teacher model
2) Train the student network with cross-entropy on this new dataset

Sequence-Level Knowledge Distillation (Kim & Rush, 2016)

1) Run beam search over the training set with the teacher model
2) Train the student network with cross-entropy on this new dataset

Sequence-Level Interpolation (Kim & Rush, 2016)

1) Run beam search over the training set with the teacher model with **K candidate translations**
2) Select a sequence which is close to the training target sequence in terms of similarity
3) Train the student network with cross-entropy on this new dataset

How is Knowledge Distillation performed for NMT models?

Model	$\text{BLEU}_{K=5}$	$\Delta_{K=5}$
English → German WMT 2014		
Teacher Baseline 4 × 1000 (Params: 221m)	19.5	–
Student Baseline 2 × 500 (Params: 84m)	17.6	–
Word-KD	17.7	+0.1
Seq-KD	19.0	+1.4
Student Baseline 2 × 300 (Params: 49m)	16.9	–
Word-KD	17.6	+0.7
Seq-KD	18.1	+1.2

What makes sequence-level knowledge distillation effective in compressing knowledge into the student model ?

Why does Sequence-Level KD works?

- First hypothesis (Kim and Rush, 2016)
  - Student models are smaller → more challenging to accurately fit the noisy training data
  - SLKD simplifies the noisy data

Why does Sequence-Level KD works?

- First hypothesis (Kim and Rush, 2016)
  - Student models are smaller → more challenging to accurately fit the noisy training data
  - SLKD simplifies the noisy data and frees more capacity
- Second hypothesis (Gordon et al., 2019)
  - SLKD performs regularization through data augmentation
  - SLKD does not restrict the model capacity

Why does Sequence-Level KD works?

- First hypothesis (Kim and Rush, 2016)
  - Student models are smaller → more challenging to accurately fit the noisy training data
  - SLKD simplifies the noisy data and frees more capacity
- Second hypothesis (Gordon et al., 2019)
  - SLKD performs regularization through data augmentation
  - SLKD does not restrict the model capacity
  - To confirm the hypothesis:

	SMALL Students				LARGE Students	
	w/ Dropout		No Dropout			
Dataset	BLEU	PPL_{Train}	BLEU	PPL_{Train}	BLEU	PPL_{Train}
baseline	26.79	4.86	25.37	4.24	31.75	4.99
kd	27.70	2.17	26.45	2.09	30.38	1.93
base+kd	27.74	3.53	27.84	3.02	32.52	3.33
base+kd+bt	27.87	3.41	**28.38**	2.93	**32.99**	3.29
base+best-2	**27.92**	3.12	28.03	2.64	32.59	2.73

Table 3: The tokenized test BLEU scores (Beam=5)[6] and BPE train perplexities for student models trained on concatenations of datasets. SMALL students are trained for 100 checkpoints, rather than the initial 30.

Why does Sequence-Level KD works?

- First hypothesis (Kim and Rush, 2016)
  - Student models are smaller → more challenging to accurately fit the noisy training data
  - SLKD simplifies the noisy data and frees more capacity
- Second hypothesis (Gordon et al., 2019)
  - SLKD performs regularization through data augmentation
  - SLKD does not restrict the model capacity
  - To confirm the hypothesis:

	SMALL Students				LARGE Students	
	w/ Dropout		No Dropout			
Dataset	BLEU	PPL_{Train}	BLEU	PPL_{Train}	BLEU	PPL_{Train}
baseline	26.79	4.86	25.37	4.24	31.75	4.99
kd	27.70	2.17	26.45	2.09	30.38	1.93
base+kd	27.74	3.53	27.84	3.02	32.52	3.33
base+kd+bt	27.87	3.41	**28.38**	2.93	**32.99**	3.29
base+best-2	**27.92**	3.12	28.03	2.64	32.59	2.73

Table 3: The tokenized test BLEU scores (Beam=5)[6] and BPE train perplexities for student models trained on concatenations of datasets. SMALL students are trained for 100 checkpoints, rather than the initial 30.

<u>Analysis</u>

1. Regularizing via dropout can help generalization at the cost of model capacity

2. Regularizing via SLKD helps the model generalize without restricting its capacity

Why does Sequence-Level KD works?

- First hypothesis (Kim and Rush, 2016)
  - Student models are smaller → more challenging to accurately fit the noisy training data
  - SLKD simplifies the noisy data and frees more capacity
- Second hypothesis (Gordon et al., 2019)
  - SLKD performs regularization through data augmentation
  - SLKD does not restrict the model capacity
- Third hypothesis (Zhang et al., 2023)
  - Almost all the knowledge of the teacher comes from the teacher's top-1 information
  - To confirm the hypothesis:

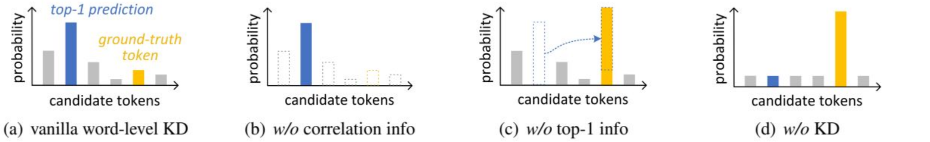


Figure 1: Removing different information from the original soft targets provided by the teacher during word-level KD. Note that the soft target in "*w/o* KD" is equivalent to the soft target of label smoothing.

Why does Sequence-Level KD works?

- Third hypothesis (Zhang et al., 2023)
  - Almost all the knowledge of the teacher comes from the teacher's top-1 information

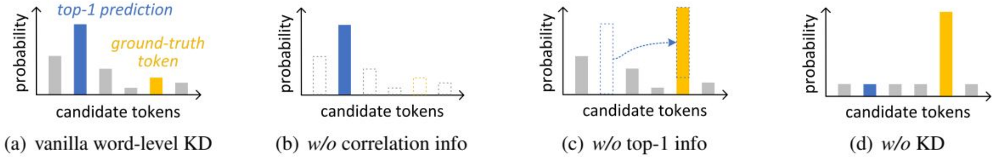


Figure 1: Removing different information from the original soft targets provided by the teacher during word-level KD. Note that the soft target in "*w/o* KD" is equivalent to the soft target of label smoothing.

Why does Sequence-Level KD works?

- Third hypothesis (Zhang et al., 2023)
  - Almost all the knowledge of the teacher comes from the teacher's top-1 information

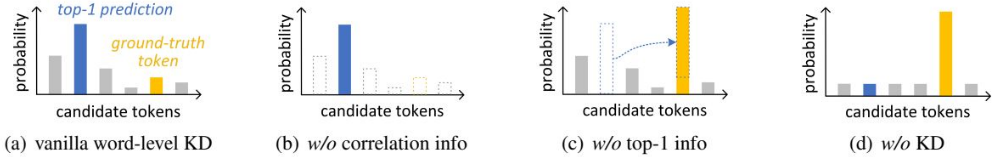


Figure 1: Removing different information from the original soft targets provided by the teacher during word-level KD. Note that the soft target in "*w/o* KD" is equivalent to the soft target of label smoothing.

Task	Model	TA	BLEU
En-De	(a) vanilla word-level KD	88.98	26.66
	(b) *w/o* correlation info	88.69	26.76
	(c) *w/o* top-1 info	87.49	26.43
	(d) *w/o* KD	87.22	26.37

Top-1 Agreement (TA) rate: overlap rate of the top-1 predictions between the student and the teacher on each position

What are the alternative KD techniques available for NMT models?

1. Selective Knowledge Distillation for NMT (Wang et al., 2021)

1. Selective Knowledge Distillation for NMT (Wang et al., 2021)

- Not all knowledge from the teacher model is beneficial during KD

1. Selective Knowledge Distillation for NMT (Wang et al., 2021)

- Not all knowledge from the teacher model is beneficial during KD
- **Word CE** measures how the student model agrees with the golden label

1. Selective Knowledge Distillation for NMT (Wang et al., 2021)

- Not all knowledge from the teacher model is beneficial during KD
- **Word CE** measures how the student model agrees with the golden label
- Words with large CE are more **difficult** to learn

1. Selective Knowledge Distillation for NMT (Wang et al., 2021)

- Not all knowledge from the teacher model is beneficial during KD
- **Word CE** measures how the student model agrees with the golden label
- Words with large CE are more **difficult** to learn

- Two Strategies proposed:
  1. Batch-Level Selection Strategy
  2. Global-Level Selection Strategy

1. Selective Knowledge Distillation for NMT (Wang et al., 2021)

A. Batch-Level Selection Strategy

- Choose **top r% words with higher CE** within current mini-batch and distill them
- Hard samples get extra supervision

$$\mathcal{L}_{kd} = \begin{cases} -\sum_{k=1}^{|V|} q(y_k) \cdot \log p(y_k), & y \in \mathcal{S}_{Hard} \\ 0 & , y \in \mathcal{S}_{Easy} \end{cases}$$

where we simplify the notation of p and q for clarity.

1. Selective Knowledge Distillation for NMT (Wang et al., 2021)

A. Batch-Level Selection Strategy

- Choose top r% words with higher CE within current mini-batch and distill them
- Hard samples get extra supervision

$$\mathcal{L}_{kd} = \begin{cases} -\sum_{k=1}^{|V|} q(y_k) \cdot \log p(y_k), & y \in \mathcal{S}_{Hard} \\ 0 & , y \in \mathcal{S}_{Easy} \end{cases}$$

where we simplify the notation of p and q for clarity.

B. Global-Level Selection Strategy

Approximate optimal global CE distribution using a queue

Algorithm 1 Global-level Selection

Input: B: mini-batch, $\mathcal{Q}$: FIFO global queue, $\mathcal{T}$: teacher model, $\mathcal{S}$: student model

```
for each word_i in B do
    Compute L_ce of word_i by Equation 1
    Compute L_kd of word_i by Equation 2
    Push L_ce to Q
    if L_ce in top_r%(Q) then
        Loss_i ← L_ce + α · L_kd
    else
        Loss_i ← L_ce
    Loss ← Loss + Loss_i
Update S with respect to Loss
```

1. Selective Knowledge Distillation for NMT (Wang et al., 2021)

Transformer	27.29	ref
Word-KD	28.14	+0.85
Seq-KD	28.15	+0.86
Batch-level Selection	28.42*	+1.13
Global-level Selection	**28.57*†**	**+1.28**

Table 2: BLEU scores (%) on WMT'14 English-German (En-De) task. Δ shows the improvement compared to Transformer (Base). '*': significantly ($p < 0.01$) better than Transformer (Base). '†': significantly ($p < 0.05$) better than the Word/Seq-KD models.

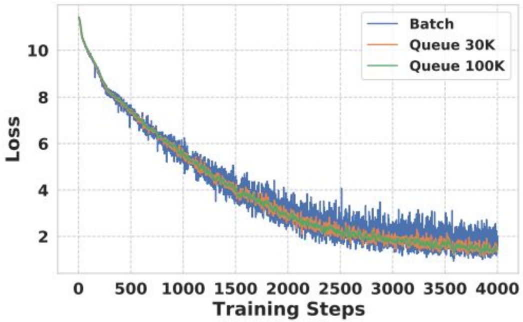


Figure 6: Partition point for $\mathcal{S}_{Hard}$ and $\mathcal{S}_{Easy}$, with respect to different strategies. Batch-level selection clearly suffers from large fluctuations and high variance.

2. Nearest Neighbor Knowledge Distillation for NMT (Yang et al., 2022)

2. Nearest Neighbor Knowledge Distillation for NMT (Yang et al., 2022)

- Overcorrection Phenomenon (Zhang et al., 2019)

2. Nearest Neighbor Knowledge Distillation for NMT (Yang et al., 2022)

- Overcorrection Phenomenon (Zhang et al., 2019)
  - The standard NMT models are typically trained with CE loss, which requires a strict **word-by-word matching** between the model prediction and the ground-truth

2. Nearest Neighbor Knowledge Distillation for NMT (Yang et al., 2022)

- Overcorrection Phenomenon (Zhang et al., 2019)
  - The standard NMT models are typically trained with CE loss, which requires a strict **word-by-word matching** between the model prediction and the ground-truth
  - Even when the model predicts a word that is reasonable but deviates from the ground-truth, the CE loss will treat it as an error and punish the model

2. Nearest Neighbor Knowledge Distillation for NMT (Yang et al., 2022)

- Overcorrection Phenomenon (Zhang et al., 2019)
  - The standard NMT models are typically trained with CE loss, which requires a strict **word-by-word matching** between the model prediction and the ground-truth
  - Even when the model predicts a word that is reasonable but deviates from the ground-truth, the CE loss will treat it as an error and punish the model
- Two proposed solutions based on KNN:
  A. **kNN-MT**(Khandelwal et al., 2021)
  B. **kNN-KD** (Yang et al., 2022)

2. Nearest Neighbor Knowledge Distillation for NMT (Yang et al., 2022)

A. KNN-MT

- <u>Training Step:</u> The context representations and target tokens are stored into a large datastore

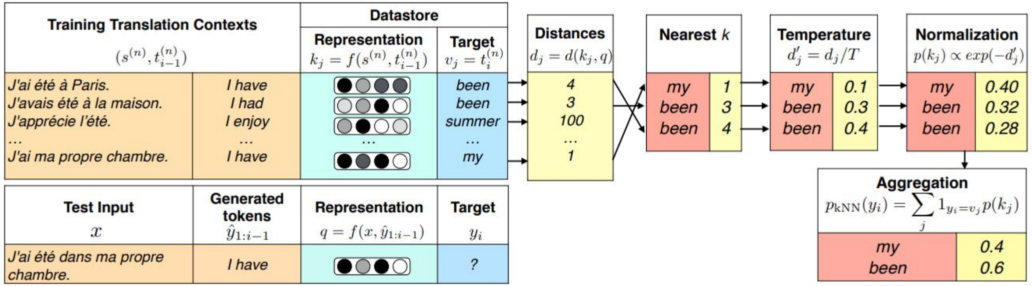


Figure 1: An illustration of how the kNN distribution is computed. The datastore, which is constructed offline, consists of representations of training set translation contexts and corresponding target tokens for every example in the parallel data. During generation, the query representation, conditioned on the test input as well as previously generated tokens, is used to retrieve the k nearest neighbors from the datastore, along with the corresponding target tokens. The distance from the query is used to compute a distribution over the retrieved targets after applying a softmax temperature. This distribution is the final kNN distribution.

2. Nearest Neighbor Knowledge Distillation for NMT (Yang et al., 2022)

A. KNN-MT

- <u>Inference</u>:
  - k possible target tokens are retrieved by conducting nearest search from the datastore every decoding step
  - KNN-MT interpolates a base NMT model's probability with the KNN model

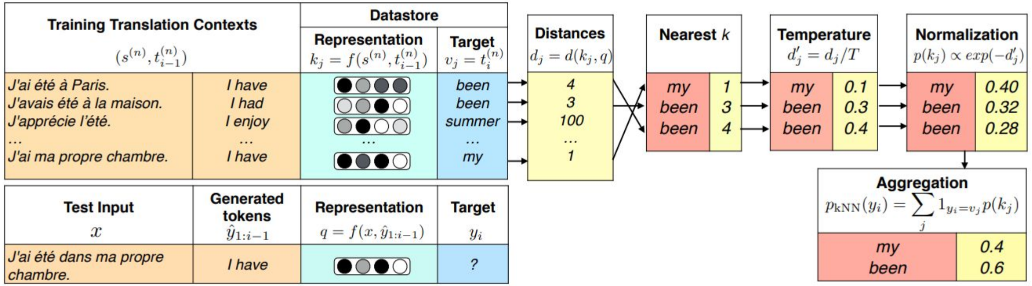


Figure 1: An illustration of how the kNN distribution is computed. The datastore, which is constructed offline, consists of representations of training set translation contexts and corresponding target tokens for every example in the parallel data. During generation, the query representation, conditioned on the test input as well as previously generated tokens, is used to retrieve the k nearest neighbors from the datastore, along with the corresponding target tokens. The distance from the query is used to compute a distribution over the retrieved targets after applying a softmax temperature. This distribution is the final kNN distribution.

2. Nearest Neighbor Knowledge Distillation for NMT (Yang et al., 2022)

A. KNN-MT

- <u>Problem:</u> Each decoding step of each beam requires a kNN search over the whole datastore

→ Hard to be deployed in real-world applications

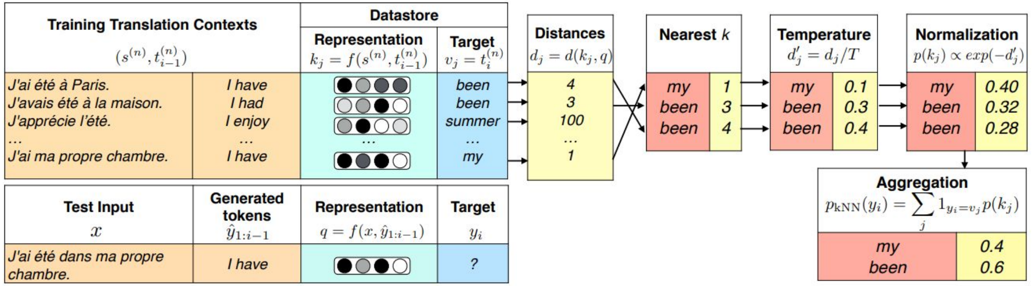


Figure 1: An illustration of how the kNN distribution is computed. The datastore, which is constructed offline, consists of representations of training set translation contexts and corresponding target tokens for every example in the parallel data. During generation, the query representation, conditioned on the test input as well as previously generated tokens, is used to retrieve the k nearest neighbors from the datastore, along with the corresponding target tokens. The distance from the query is used to compute a distribution over the retrieved targets after applying a softmax temperature. This distribution is the final kNN distribution.

2. Nearest Neighbor Knowledge Distillation for NMT (Yang et al., 2022)

B. KNN-KD

- Use the KNN-MT model as a teacher and train a base NMT model by approximating the distribution of KNN and using classical NMT-KD

2. Nearest Neighbor Knowledge Distillation for NMT (Yang et al., 2022)

B. KNN-KD

- Use the KNN-MT model as a teacher and train a base NMT model by approximating the distribution of KNN and using classical NMT-KD

Models	De-En		
	BLEU	upd/s	token/s
Transformer	34.11	2.02(1.00×)	3148.10(1.00×)
Word-KD	34.26	1.77(0.88×)	3291.28(1.06×)
Seq-KD	34.60	2.14(1.06×)	3409.86(1.08×)
Selective-KD	34.38	1.72(0.85×)	3365.68(1.07×)
*k*NN-MT	36.17	-	920.72(0.29×)
*k*NN-KD	**36.30**	2.14(1.06×)	3321.24(1.05×)

3. Annealing Knowledge Distillation (Jafari et al., 2021)

3. Annealing Knowledge Distillation (Jafari et al., 2021)

Capacity gap problem

3. Annealing Knowledge Distillation (Jafari et al., 2021)

Capacity gap problem

Annealing-KD

- **Stage I:** gradually training the student to mimic the teacher using the Annealing-KD loss

- **Stage II:** fine-tuning the student with hard labels using the CE loss

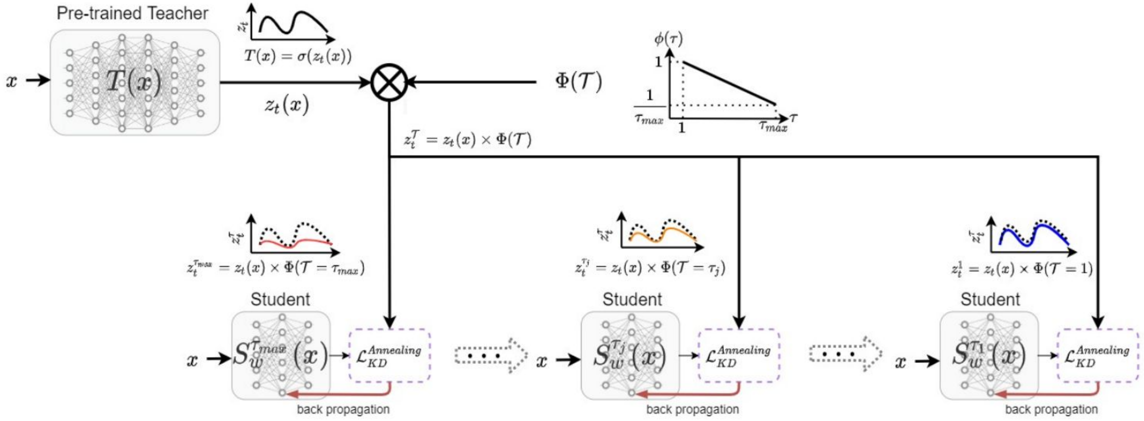


Figure 1: Illustrating the Stage I of the Annealing-KD technique. Given a pre-trained teacher network, we can derive the annealed output of the teacher at different temperature using the annealing function $\Phi(\mathcal{T})$. We start training of the student from $\mathcal{T} = \tau_{max}$ and go to $\mathcal{T} = 1$.

4. Top-1 Information Enhanced Knowledge Distillation (Zhang et al., 2023)

- SLKD works because we distill Top-1 Information from the teacher (third hypothesis)
- The classic KD methods lack specialized learning of the most **important top-1 information**

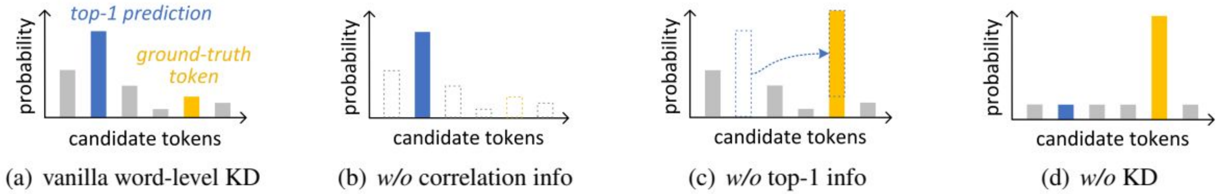


Figure 1: Removing different information from the original soft targets provided by the teacher during word-level KD. Note that the soft target in "*w/o* KD" is equivalent to the soft target of label smoothing.

4. Top-1 Information Enhanced Knowledge Distillation (Zhang et al., 2023)

- TIEKD combines
  - Hierarchical Ranking Loss
  - Iterative Knowledge Distillation

4. Top-1 Information Enhanced Knowledge Distillation (Zhang et al., 2023)

- TIEKD combines
  - **Hierarchical Ranking Loss**
  - Iterative Knowledge Distillation

<u>Hierarchical Ranking Loss</u>

- Boosts the learning of the top-1 information from the teacher
- The student model can be enforced to rank the top-1 predictions of the teacher to its own top-1 places

4. Top-1 Information Enhanced Knowledge Distillation (Zhang et al., 2023)

- TIEKD combines
  - Hierarchical Ranking Loss
  - **Iterative Knowledge Distillation**

Hierarchical Ranking Loss

- Boosts the learning of the top-1 information from the teacher
- The student model can be enforced to rank the top-1 predictions of the teacher to its own top-1 places

Algorithm 1 Iterative Knowledge Distillation

Input: source and target data in current mini-batch $(\mathbf{x}, \mathbf{y})$; student model $\mathcal{S}$; teacher model $\mathcal{T}$; iteration times N;

Initialize $\mathbf{y}^0 = \mathbf{y}$; $\mathcal{L}_{kd} = 0$;
Compute $\mathcal{L}_{ce}$ based on Eq.(1)
for i in $1, 2, ..., N$ **do**
  $p^i = \mathcal{S}(\mathbf{x}; \mathbf{y}^{i-1})$ ▷ *probability distributions from the student model*
  $q^i = \mathcal{T}(\mathbf{x}; \mathbf{y}^{i-1})$ ▷ *probability distributions from the teacher model*
  Compute $\mathcal{L}^i_{kd}(p^i, q^i)$ based on Eq.(7)
  $\mathcal{L}_{kd} \leftarrow \mathcal{L}_{kd} + \mathcal{L}^i_{kd}$
  $\mathbf{y}^i = \arg\max(p^i)$ ▷ *student predictions as inputs in the next iteration*
end for
$\mathcal{L}_{word\text{-}kd} \leftarrow (1-\alpha)\mathcal{L}_{ce} + \frac{\alpha}{N}\mathcal{L}_{kd}$

4. Top-1 Information Enhanced Knowledge Distillation (Zhang et al., 2023)

Methods	WMT'14 En-De		WMT'14 En-Fr		WMT'16 En-Ro	
	BLEU	COMET	BLEU	COMET	BLEU	COMET
Student (*Transformer$_{base}$*)	27.42$_{\pm0.01}$	48.11$_{\pm1.04}$	40.97$_{\pm0.14}$	62.19$_{\pm0.11}$	33.59$_{\pm0.15}$	50.96$_{\pm0.43}$
+ Word-KD (Kim and Rush, 2016)	28.03$_{\pm0.10}$	51.59$_{\pm0.23}$	41.10$_{\pm0.11}$	63.81$_{\pm0.14}$	33.77$_{\pm0.01}$	53.15$_{\pm0.26}$
+ Seq-KD (Kim and Rush, 2016)	28.22$_{\pm0.02}$	51.23$_{\pm0.15}$	41.44$_{\pm0.02}$	63.12$_{\pm0.14}$	33.69$_{\pm0.02}$	50.63$_{\pm0.11}$
+ Annealing KD (Jafari et al., 2021)	27.91$_{\pm0.10}$	51.58$_{\pm0.03}$	41.20$_{\pm0.13}$	63.59$_{\pm0.09}$	33.67$_{\pm0.09}$	52.22$_{\pm1.02}$
+ Selective-KD (Wang et al., 2021)	28.24$_{\pm0.21}$	52.15$_{\pm0.42}$	41.25$_{\pm0.04}$	64.24$_{\pm0.01}$	33.74$_{\pm0.02}$	53.05$_{\pm0.28}$
+ TIE-KD (ours)	**28.46**$^{*}_{\pm0.01}$	**52.63**$^{*}_{\pm0.09}$	**41.57**$^{*}_{\pm0.08}$	**65.06**$^{*}_{\pm0.44}$	**34.70**$^{*}_{\pm0.07}$	**55.76**$^{*}_{\pm0.21}$
Teacher (*Transformer$_{big}$*)	*28.81*	*53.20*	*42.98*	*69.58*	*34.70*	*57.04*

Table 6: BLEU scores (%) and COMET (Rei et al., 2020) scores (%) on three translation tasks. Results with † are taken from the original papers. Others are our re-implementation results using the released code with the same setting in Sec.5.2 for a fair comparison. We report average results over 3 runs with random initialization. Results with $*$ are statistically (Koehn, 2004) better than the vanilla Word-KD with $p < 0.01$.

Notes on Knowledge Distillation

Notes on Knowledge Distillation

- Knowledge distillation and data augmentation (Aji & Heafield, 2020)
    1. Training data for students does not have to be the same as the teacher as long as the domain agrees

Notes on Knowledge Distillation

- Knowledge distillation and data augmentation (Aji & Heafield, 2020)
  1. Training data for students does not have to be the same as the teacher as long as the domain agrees
  2. Generally, more training data often leads to better performance. In KD, generating and mixing synthetic data is more important.

Notes on Knowledge Distillation

- Knowledge distillation and data augmentation (Aji & Heafield, 2020)
    1. Training data for students does not have to be the same as the teacher as long as the domain agrees
    2. Generally, more training data often leads to better performance. In KD, generating and mixing synthetic data is more important.
    3. Augmenting the dataset with forward translated source text and forward translated back-translated text improve BLEU depending on the test set's original language.
        - Forward translating source originated text worked well if the test set was also originated from the source language.
        - In contrast, forward translating back translation data worked well if the test set was originated from the target language.

Other KD methods for NMT:

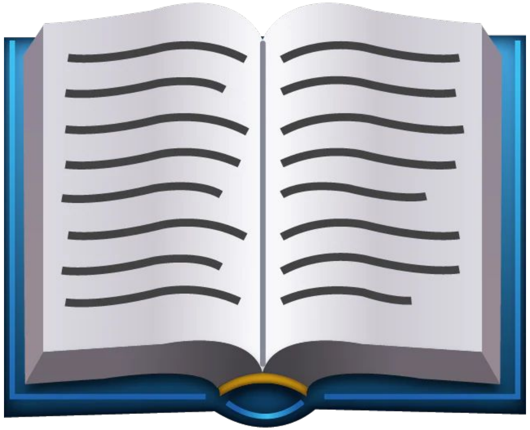
- Distill, Adapt, Distill: Training Small, In-Domain Models for Neural Machine Translation
- Target-Oriented Knowledge Distillation with Language-Family-Based Grouping for Multilingual NMT
- Continual Knowledge Distillation for Neural Machine Translation
- Collective Wisdom: Improving Low-resource Neural Machine Translation using Adaptive Knowledge Distillation
- Combining Sequence Distillation and Transfer Learning for Efficient Low-Resource Neural Machine Translation Models
- Life-long Learning for Multilingual Neural Machine Translation with Knowledge Distillation

Other generic KD methods

- A Study on Knowledge Distillation from Weak Teacher for Scaling Up Pre-trained Language Models
- ReAugKD: Retrieval-Augmented Knowledge Distillation For Pre-trained Language Models
- AD-KD: Attribution-Driven Knowledge Distillation for Language Model Compression
- Robustness Challenges in Model Distillation and Pruning for Natural Language Understanding
- BERT Learns to Teach: Knowledge Distillation with Meta Learning
- Tailoring Instructions to Student's Learning Levels Boosts Knowledge Distillation
- Parameter-Efficient and Student-Friendly Knowledge Distillation

Do you want to learn more about Knowledge Distillation?

- Join our reading group on Knowledge Distillation
  - Organized jointly with **Ona De Gibert**
  - Every **Tuesday at 11:00 AM starting November 7**
- Join our Slack channel: #reading-group



References

Aji, A. F., & Heafield, K. (2020). Fully Synthetic Data Improves Neural Machine Translation with Knowledge Distillation.

Gumma, V., Dabre, R., & Kumar, P. (2023). An Empirical Study of Leveraging Knowledge Distillation for Compressing Multilingual Neural Machine Translation Models. Proceedings of the 24th Annual Conference of the European Association for Machine Translation.

Jafari, A., Rezagholizadeh, M., Sharma, P., & Ghodsi, A. (2021). Annealing Knowledge Distillation. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 2493–2504, Online. Association for Computational Linguistics.

Kim, Y., & Rush, A. M. (2016). Sequence-Level Knowledge Distillation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.

Wang, F., Yan, J., Meng, F., & Zhou, J. (2021). Selective Knowledge Distillation for Neural Machine Translation. 6456-6466. 10.18653/v1/2021.acl-long.504.

Wu, Y., Passban, P., & Rezagholizadeh, M. (2020). Why Skip If You Can Combine: A Simple Knowledge Distillation Technique for Intermediate Layers. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics.

Yang, Z., Sun, R., & Wan, X. (2022). Nearest Neighbor Knowledge Distillation for Neural Machine Translation. Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics.

Zhang, S., Liang, Y., Wang, S., Chen, Y., Han, W., Liu, J., & Xu, J. (2023). Towards Understanding and Improving Knowledge Distillation for Neural Machine Translation. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics.

Thanks for listening!