Sequence-Level Knowledge Distillation for Class-Incremental End-to-End Spoken Language Understanding

Umberto Cappellazzo[1,3], Muqiao Yang[2], Daniele Falavigna[3], Alessio Brutti[3]

1 UNIVERSITY OF TRENTO

2 Carnegie Mellon University

3 FONDAZIONE BRUNO KESSLER

Motivation: the real world is *dynamic*

- The brittle *i.i.d. assumption* is far from what really happens in real scenarios.
- After training, the model must be able to cope with **shifts** in the data distribution or adapt to *new* objects/categories. <u>Retraining from scratch</u> the model is almost always unfeasible!
- However, DNNs fail to learn novel concepts sequentially because they overwrite the existing knowledge in favor of the new data.

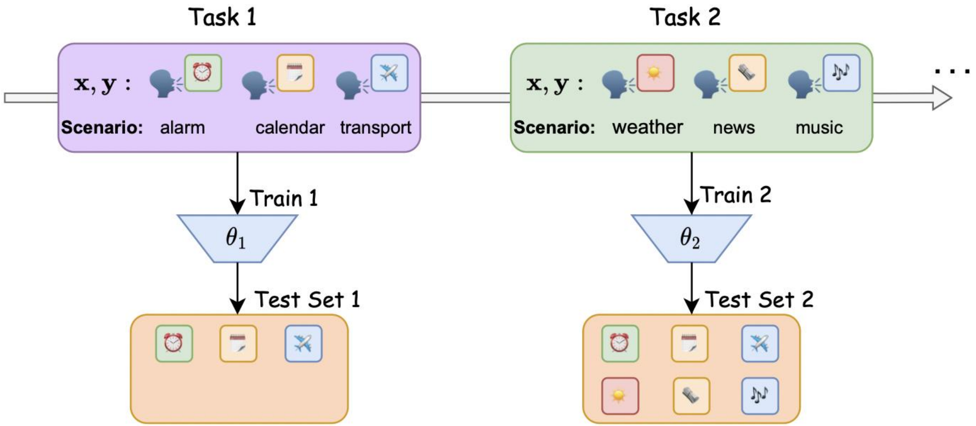
↓

"**Catastrophic forgetting**" of the past knowledge

Kirkpatrick et al., *Overcoming catastrophic forgetting in neural networks*, 2017.

Class-Incremental Learning

- **Class-Incremental Learning** (CIL) setup: non-overlapping classes arrive sequentially (i.e., "**tasks**"), and the model needs to learn to classify all the classes incrementally.
- Following the learning process for each task, the model's performance is assessed across all the classes (past + new).

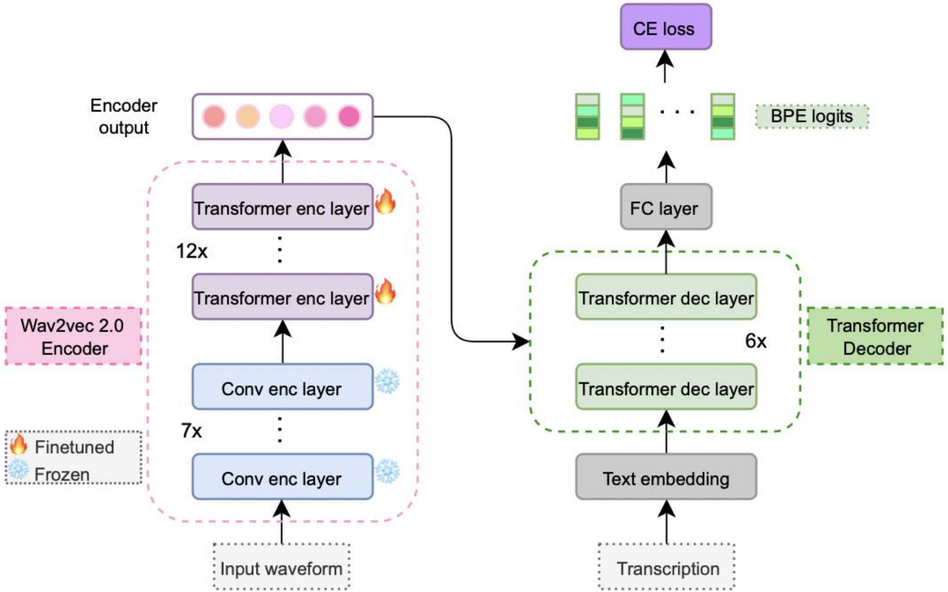


Zhou et al., *Deep Class-Incremental Learning: A Survey*, 2023.

CIL for joint ASR-SLU

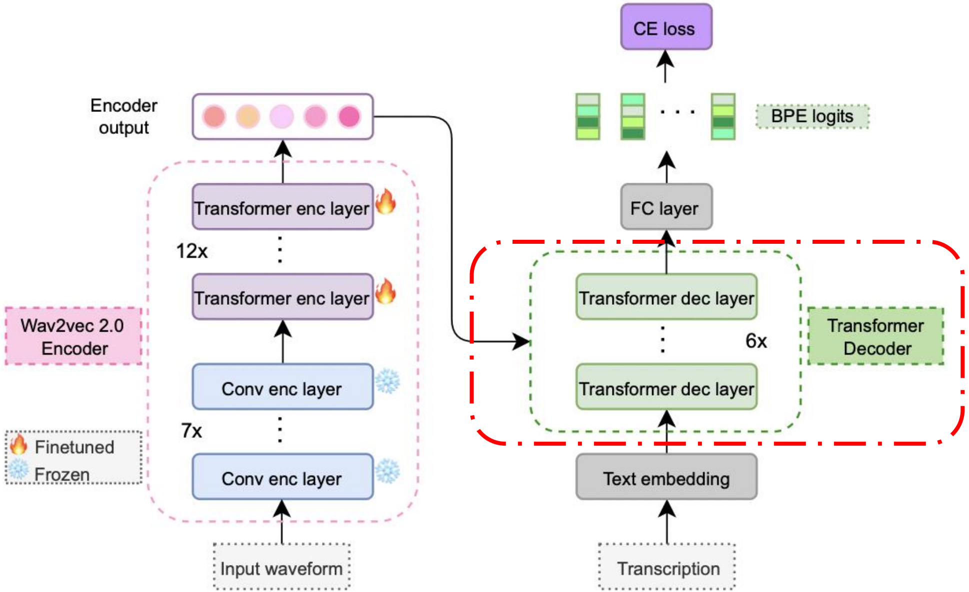
- **Spoken Language Understanding** (SLU): intent classification + entity classification (aka slot-filling).
- In our setting new intent classes emerge sequentially.
- Intent/entities predicted along with the real transcription using an ASR encoder-decoder model.

How can we mitigate forgetting for an enc-dec ASR system?



- Unlike standard enc + classifier pipeline used for CL, our system includes an ASR decoder.

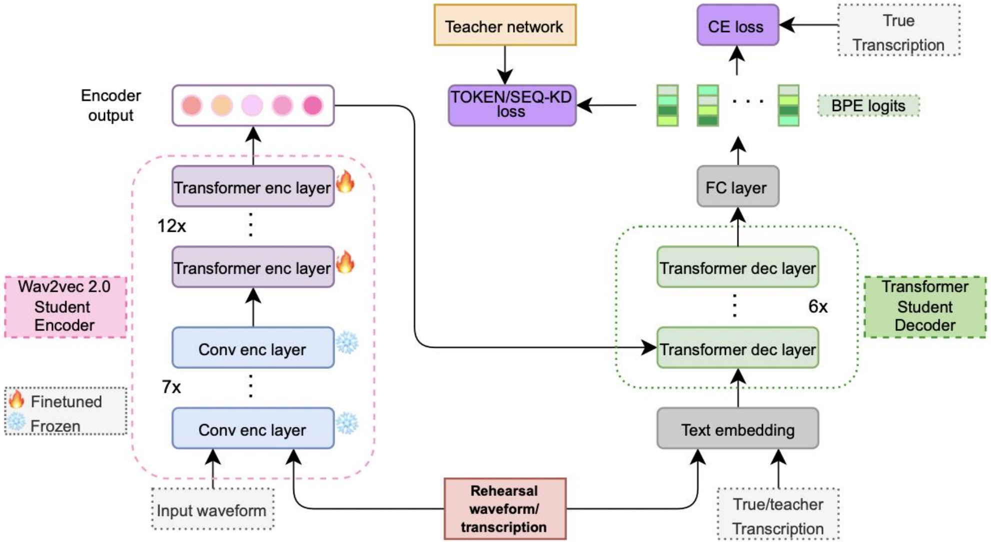
How can we mitigate forgetting for an enc-dec ASR system?



- Unlike standard enc + classifier pipeline used for CL, our system includes an ASR decoder.
- Main focus of our work: alleviate forgetting at the decoder side.

Proposed Approach

- We use a **rehearsal memory** to store a small fraction of samples from the previous tasks.
- We propose two **knowledge distillation** (KD) based approaches operating at the decoder side: 1) *Token-KD* 2) *Sequence-KD*.
- The two proposed KD losses are applied only to the rehearsal samples!



Kim et al., *Sequence-Level Knowledge Distillation*, 2016.

Preliminaries

ASR objective function:

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}*} p(\mathbf{y} | \mathbf{x}; \theta)$$

Preliminaries

ASR objective function: $$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{Y}*}{\operatorname{argmax}}\, p(\mathbf{y}|\mathbf{x};\theta)$$

Rehearsal data: $\mathcal{R}_t$

Current data: $\mathcal{D}_t$

Preliminaries

ASR objective function: $$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathcal{Y}^*}{\operatorname{argmax}}\, p(\mathbf{y}|\mathbf{x};\theta)$$

Rehearsal data: $\mathcal{R}_t$

Current data: $\mathcal{D}_t$

Student model distr. : $p(\mathbf{y}|\mathbf{x};\theta_t)$

Teacher model distr. : $p(\mathbf{y}|\mathbf{x};\theta_{t-1})$

Preliminaries

ASR objective function:

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}^*} p(\mathbf{y}|\mathbf{x};\theta)$$

Rehearsal data: $\mathcal{R}_t$

Current data: $\mathcal{D}_t$

Student model distr. : $p(\mathbf{y}|\mathbf{x};\theta_t)$

Teacher model distr. : $p(\mathbf{y}|\mathbf{x};\theta_{t-1})$

Standard CE loss at task *t*:

$$\mathcal{L}_{\text{CE}}^t = -\sum_{\mathbf{x} \in \mathcal{D}_t \cup \mathcal{R}_t} \log(p(\mathbf{y}|\mathbf{x};\theta_t))$$

Token-KD approach

Objective: we try to match the **student** predictions to that of the **teacher** for each token during the decoding process. We encourage the transfer of knowledge “*locally*” (for each position of the target sequence).

Token-KD approach

Objective: we try to match the **student** predictions to that of the **teacher** for each token during the decoding process. We encourage the transfer of knowledge "*locally*" (for each position of the target sequence).

$$\mathcal{L}^{t}_{\text{tok-KD}} = -\sum_{\mathbf{x}\in\mathcal{R}_t}\sum_{j=1}^{J} p(y_j|\mathbf{x},\mathbf{y}_{<j};\theta_{t-1})\boxed{\log(p(y_j|\mathbf{x},\mathbf{y}_{<j};\theta_t))}$$

Student token-level distribution

Token-KD approach

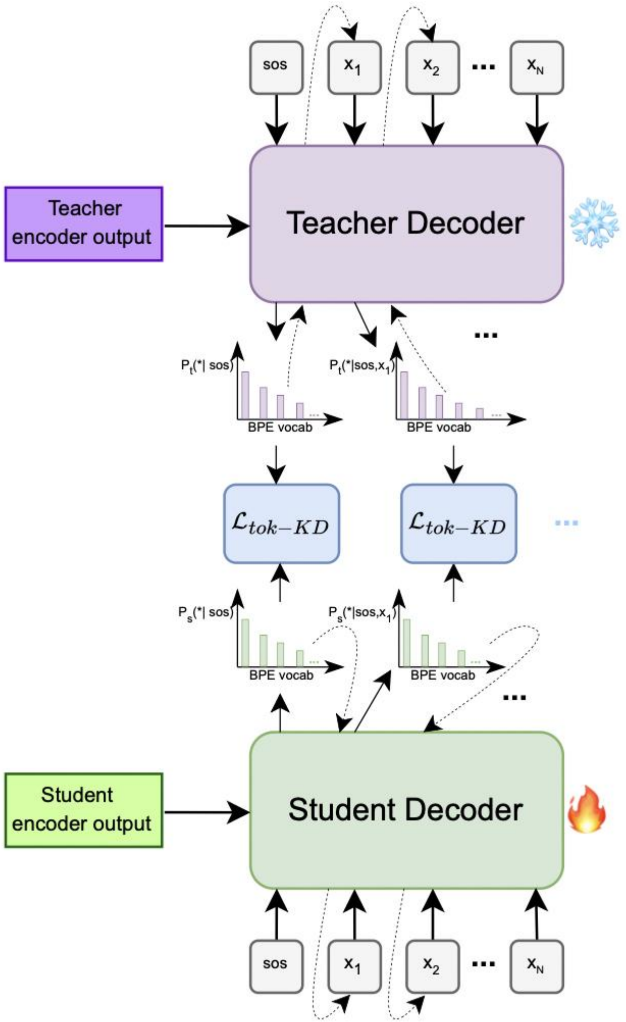
Objective: we try to match the **student** predictions to that of the **teacher** for each token during the decoding process. We encourage the transfer of knowledge "*locally*" (for each position of the target sequence).

$$\mathcal{L}^{t}_{\text{tok-KD}} = -\sum_{\mathbf{x}\in\mathcal{R}_t}\sum_{j=1}^{J} p(y_j|\mathbf{x},\mathbf{y}_{<j};\theta_{t-1})\boxed{\log(p(y_j|\mathbf{x},\mathbf{y}_{<j};\theta_t))}$$

Student token-level distribution

Note: the distributions are over the BPE tokens.

Token-KD approach



Token-KD limitations

- Token distributions are optimal “locally”, not globally. Initial errors can be propagated forward.
- Ideally, we would like the student to mimic the teacher’s behaviour at the **sequence-level**!
- The sequence distribution conveys more fine-grained and stable information.

Sequence-KD

Sequence-KD approach

$$\mathcal{L}^{t}_{\text{seq-KD}} = - \sum_{\mathbf{x} \in \mathcal{R}_t} \sum_{\mathbf{y} \in \mathcal{Y}^*} p(\mathbf{y}|\mathbf{x}; \theta_{t-1}) \boxed{\log(p(\mathbf{y}|\mathbf{x}; \theta_t))}$$

Student sequence-level distribution

Sequence-KD approach

$$\mathcal{L}^{t}_{\text{seq-KD}} = -\sum_{\mathbf{x}\in\mathcal{R}_t}\sum_{\boxed{\mathbf{y}\in\mathcal{Y}^*}} p(\mathbf{y}|\mathbf{x};\theta_{t-1})\boxed{\log(p(\mathbf{y}|\mathbf{x};\theta_t))}$$

Student sequence-level distribution

Note: we are summing over all possible sequences → intractable!

Sequence-KD approach

$$\mathcal{L}^t_{\text{seq-KD}} = -\sum_{\mathbf{x}\in\mathcal{R}_t} \sum_{\boxed{\mathbf{y}\in\mathcal{Y}^*}} p(\mathbf{y}|\mathbf{x};\theta_{t-1}) \boxed{\log(p(\mathbf{y}|\mathbf{x};\theta_t))}$$

Student sequence-level distribution

Note: we are summing over all possible sequences → intractable!

Approximation: we just use the sequence generated by using beam search with the teacher network!

Sequence-KD approach

$$\mathcal{L}^{t}_{\text{seq-KD}} = -\sum_{\mathbf{x}\in\mathcal{R}_t}\sum_{\boxed{\mathbf{y}\in\mathcal{Y}^*}} p(\mathbf{y}|\mathbf{x};\theta_{t-1})\boxed{\log(p(\mathbf{y}|\mathbf{x};\theta_t))}$$

Student sequence-level distribution

Note: we are summing over all possible sequences → intractable!

$$\mathcal{L}^{t}_{\text{seq-KD}} \approx -\sum_{\mathbf{x}\in\mathcal{R}_t}\sum_{\mathbf{y}\in\mathcal{Y}^*} \mathbb{1}\{\mathbf{y}=\tilde{\mathbf{y}}\}\log(p(\mathbf{y}|\mathbf{x};\theta_t))$$

Sequence-KD approach

$$\mathcal{L}^t_{\text{seq-KD}} = - \sum_{\mathbf{x} \in \mathcal{R}_t} \sum_{\boxed{\mathbf{y} \in \mathcal{Y}^*}} p(\mathbf{y}|\mathbf{x}; \theta_{t-1}) \boxed{\log(p(\mathbf{y}|\mathbf{x}; \theta_t))}$$

Student sequence-level distribution

Note: we are summing over all possible sequences → intractable!

$$\mathcal{L}^t_{\text{seq-KD}} \approx - \sum_{\mathbf{x} \in \mathcal{R}_t} \sum_{\mathbf{y} \in \mathcal{Y}^*} \mathbb{1}\{\mathbf{y} = \tilde{\mathbf{y}}\} \log(p(\mathbf{y}|\mathbf{x}; \theta_t))$$

$$= - \sum_{\mathbf{x} \in \mathcal{R}_t} \log(p(\tilde{\mathbf{y}}|\mathbf{x}; \theta_t))$$

Sequence-KD in practice

- At the end of previous task ***t-1***, we run **beam search** decoding over the rehearsal and we store the resulting pseudo-transcripts.
- During current task ***t***, we compute the seq-KD loss using the previous equation and we minimize it (together with the CE loss).

Main Results

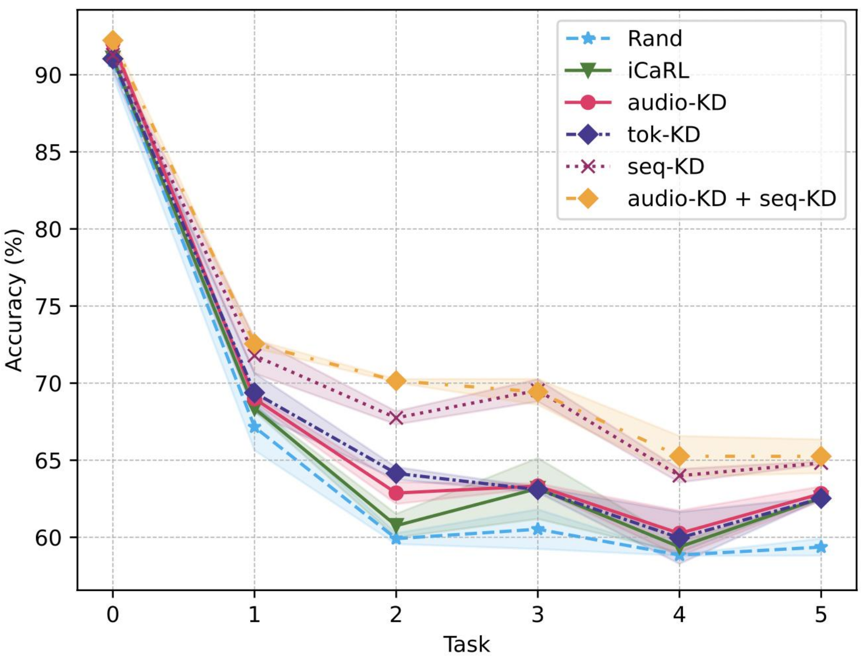
Method	SLURP-3				SLURP-6			
	Avg Acc	Last Acc	Avg WER	SLU F1	Avg Acc	Last Acc	Avg WER	SLU F1
Offline	85.84	-	20.46	70.59	85.84	-	20.46	70.59
Fine-tuning	46.27	18.36	35.82	49.25	33.56	12.42	46.26	37.88
Rehe-5% rand	79.79	74.82	25.79	65.85	77.12	73.11	28.87	63.22
Rehe-1% rand	71.30	61.47	29.13	60.05	66.11	59.37	34.77	55.33
Rehe-1% iCaRL [1]	71.49	61.66	28.62	60.23	67.55	62.55	33.82	56.09
+ audio-KD [2]	72.14	63.03	28.68	61.08	68.40	62.83	**32.04**	58.15
+ token-KD	71.79	61.54	28.82	**61.88**	68.36	62.53	32.47	58.20
+ seq-KD	**76.12**	**68.94**	**28.56**	61.50	**71.56**	**64.82**	32.50	**58.29**

[1] Rebuffi et al., *iCaRL: Incremental Classifier and Representation Learning*, 2016.
[2] Cappellazzo et al., *An Investigation of the Combination of Rehearsal and Knowledge Distillation in Continual Learning for Spoken Language Understanding*, 2023.

Ablation Study: Combining Multiple Losses

Combination	Avg Acc	Last Acc	Avg WER	SLU F1
audio + token	68.13	61.50	32.46	57.30
audio + seq	**72.48**	65.25	**31.37**	**60.00**
seq + token	72.07	63.46	33.08	58.25
audio + seq + token	71.83	**65.45**	32.55	58.48

Accuracy Trend Task by Task



Conclusion and Future Work

- We proposed two losses that operate at the decoder side to attenuate forgetting
- The *seq-KD* provides interesting gain for the evaluating metrics
- Its combination with the *audio-KD* results in the best results
- The proposed losses are applied only to the rehearsal data, which is a tiny fraction of the entire dataset (1%) → limited additional compute time.
- **What’s next?** Better approx of the seq-KD method is possible: multiple hypotheses with corresponding probs can be used! However, storage requirements scale linearly with the # of hypotheses → a tradeoff is needed!

Thank you for your attention!
Questions?

CIL for joint ASR-SLU

- **Spoken Language Understanding** (SLU): intent classification + entity classification (aka slot-filling).
- In our setting new intent classes emerge sequentially.
- Intent/entities predicted along with the real transcription using an ASR encoder-decoder model.

email_sendemail _SEP person _FILL charlotte _SEP reply email to charlotte