

How to read financial report

Structure, Strategy,

Pre-summary:

1. Tables in financial report:

- *balance sheet*
- *income statement*
- *cash flow statement*

Objective: introduce how to analyze the company's financial position from financial report

Contents:

1. Introduction of financial report
2. Structure of financial report
3. SOTA works in CV and NLP areas.
4. Sibling: Self-Supervised Learning
5. Business side prospect.

Weakly Supervised Learning

1. Introduction: What, When, Where, Why
2. Structure of financial report
3. Application and SOTA works
 - Appetizer: image classification task
 - Main-dish: speech recognition task
 - Dessert: what about self-supervised learning
4. Summary

Introduction: What, When, Where, Why

What: a periodic report about company's financial state

When:

- Q1 / Fiscal year (Q4): around end of April
- Q2 (half year): before end of August
- Q3: before end of September

Where: available at company's homepage

- * Google: XXX company financial report
- * Homepage -> investor relations -> financial report

Why: to understand the company's financial state and to **estimate the value** of stocks or bonds

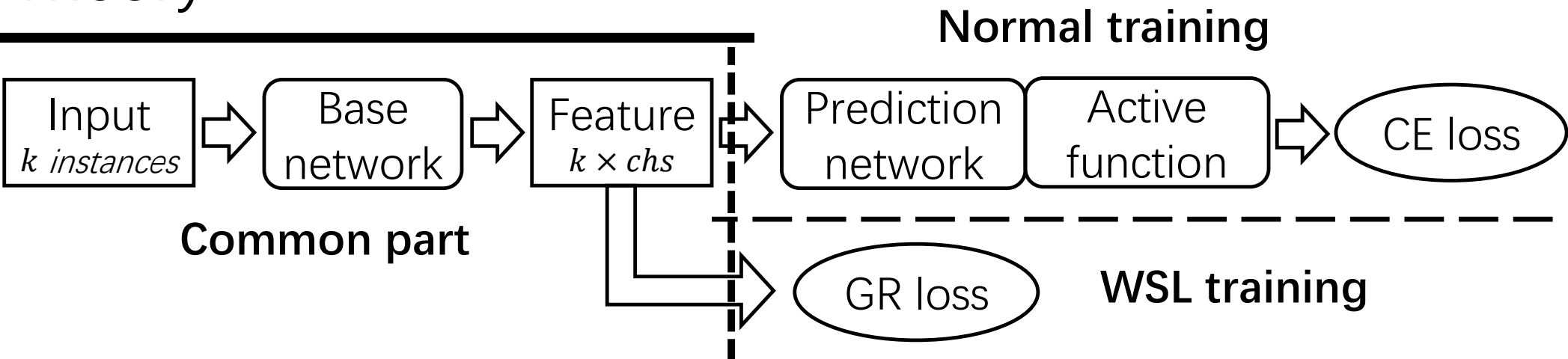
What to focus:

Common	For small companies	For big companies
Company's strategy: Year strategy, marketing, technic plan...	Industry research report: Market share, market size, competitor...	Global research report: Economics, exchange rate, geopolitics...
Financial states: Balance, income, cash flow	* Reliability of this financial report	

Weakly Supervised Learning

1. Introduction
2. Theory
3. Application and SOTA works
 - Appetizer: image classification task
 - Main-dish: speech recognition task
 - Dessert: what about self-supervised learning
4. Summary

Theory



Example of **G**raph **R**egularization loss:
(loss inside single group)

$$L_{reg} = \sum_{i,j} \tilde{s}_{i,j} \|h(\mathbf{x}_i) - h(\mathbf{x}_j)\|_2^2,$$

Accumulate
all pairs of
samples

Weight, calculated
by A,B, optional.

Instance A,B,
selected by
strategy

Distance function, usually L2

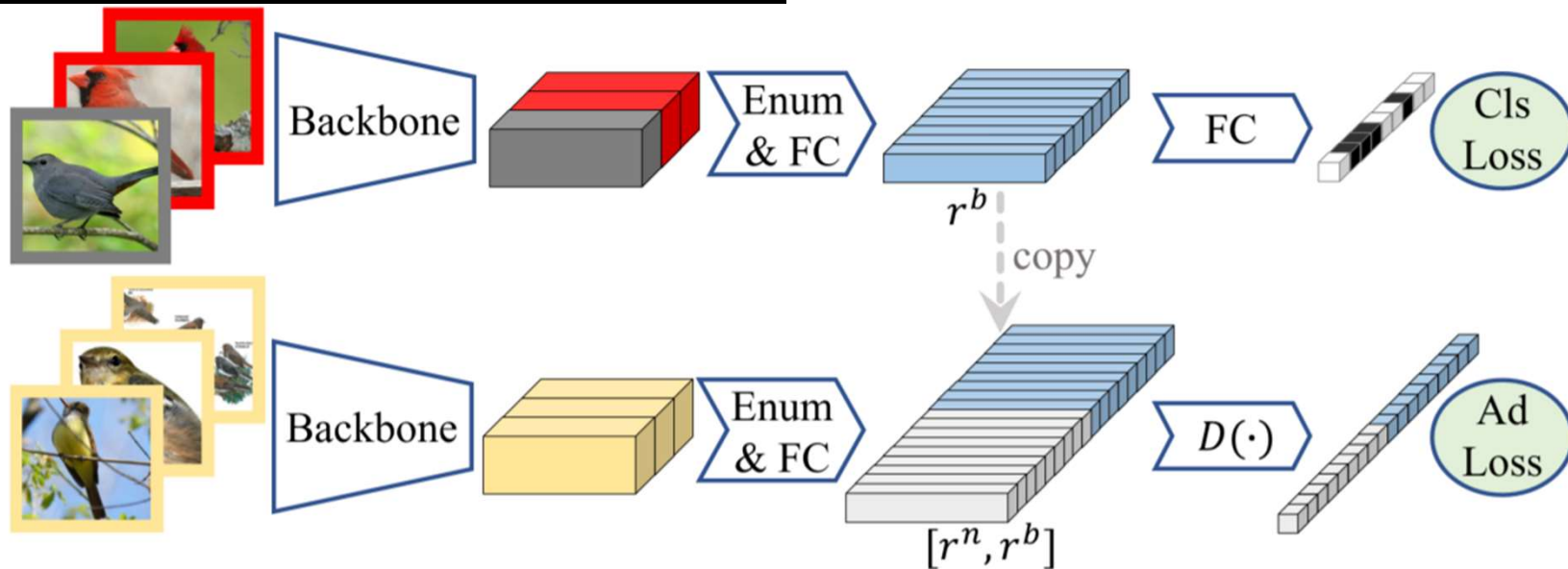
What people focus:

- How to choose instance pairs without label?
- How to add weight on these pairs?
- How to maximize distance between groups?

Weakly Supervised Learning

1. Introduction
2. Theory
3. Application and SOTA works
 - Appetizer: image classification task
 - Main-dish: speech recognition task
 - Dessert: what about self-supervised learning
4. Summary

Appetizer: Weak-shot image classification



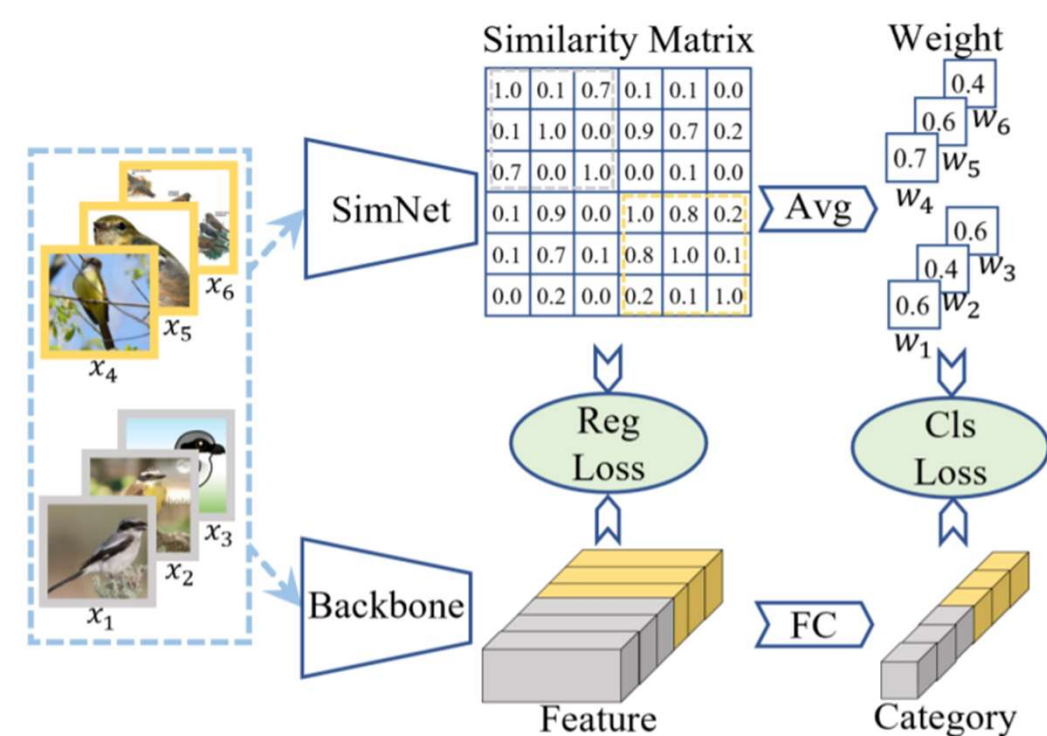
Novelty:

- Use *transferred similarity + denoise strategy* to tackle web training data.
- Apply adversarial loss to similarity net.

- Network (derived from *ref.1.1*):
- Pair-enumerate: $(k, d) \rightarrow (k, k, 2d) \rightarrow (k^2, 2d)$
- Fc: $(k^2, 2d) \rightarrow (k^2)$; Similar prediction: Y/N
-
- Feature constraint: cluster unlabeled data based
- on feature from labeled data.

Weak-shot Fine-grained Classification via Similarity Transfer, NIPS 2021

Appetizer: Weak-shot image classification



Training steps:

1. Train network on labeled data.
 - Prediction target: binary classification of similar or not.
 - CE loss.
 - Strong constraint: batch size k s.t. k^2 items pre-step.
2. Cluster unlabeled data via pre-trained network.
 - Select k_1 samples from labeled data.
 - Select k_2 samples from unlabeled data.
 - Calculate feature f_1, f_2 .
 - Weighted L2 loss by similarity.

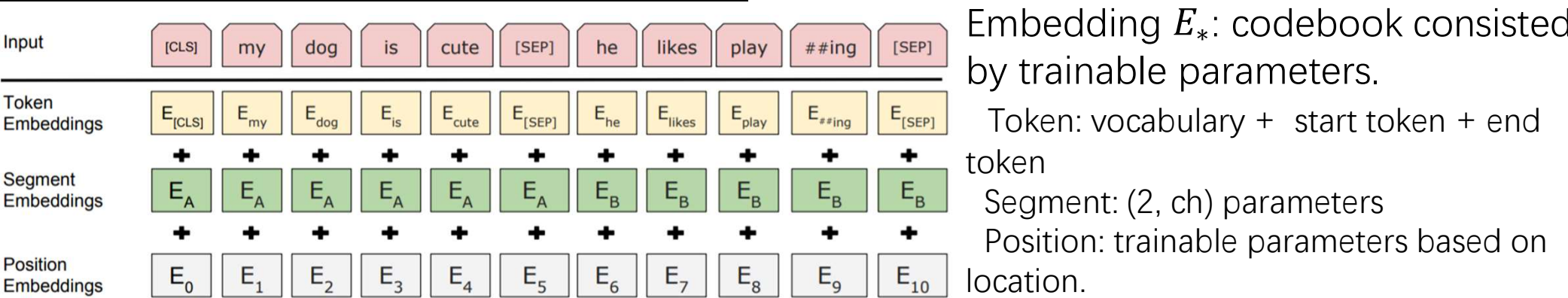
$$L_{reg} = \sum_{i,j} \tilde{s}_{i,j} \|h(\mathbf{x}_i) - h(\mathbf{x}_j)\|_2^2, \quad w_{c,i} = \frac{1}{N_c^n} \sum_{j=1}^{N_c^n} \frac{s_{c,i,j} + s_{c,j,i}}{2}.$$

Weak-shot Fine-grained Classification via Similarity Transfer, NIPS 2021

Weakly Supervised Learning

1. Introduction
2. Theory
3. Application and SOTA works
 - Appetizer: image classification task
 - Main-dish: speech recognition task
 - Dessert: what about self-supervised learning
4. Summary

Main-dish: W2v-BERT



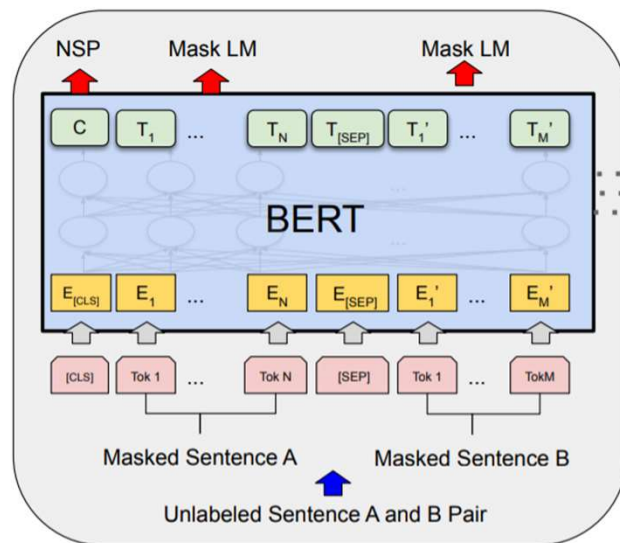
Input: 2 sentences A,B from articles.

Task: randomly mask multi-words (15%) in A and B, predict the words by context (CE loss).

Task: randomly select continuous and discontinuous (50%/50%) sentences A and B, determine whether B is the next sentence of A (CE loss).

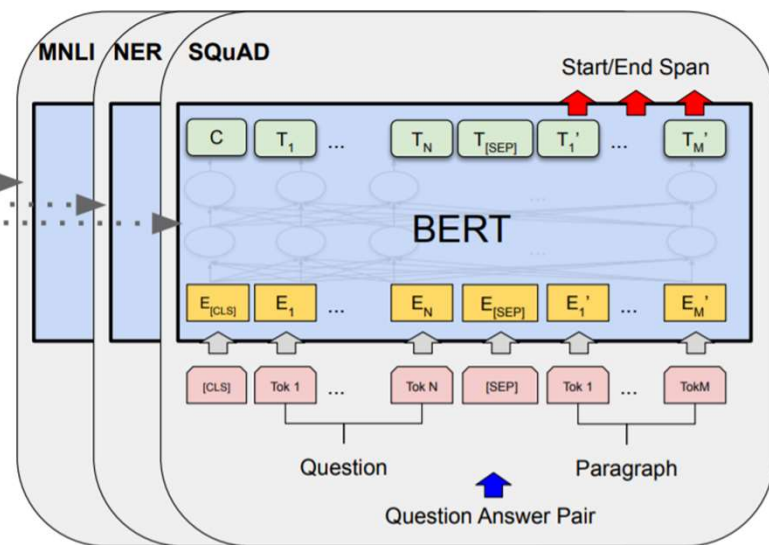
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, NACCL 2019, Google AI

Main-dish: W2v-BERT



Pre-training

Bidirectional encoder network: transformer with unfixed sequence.

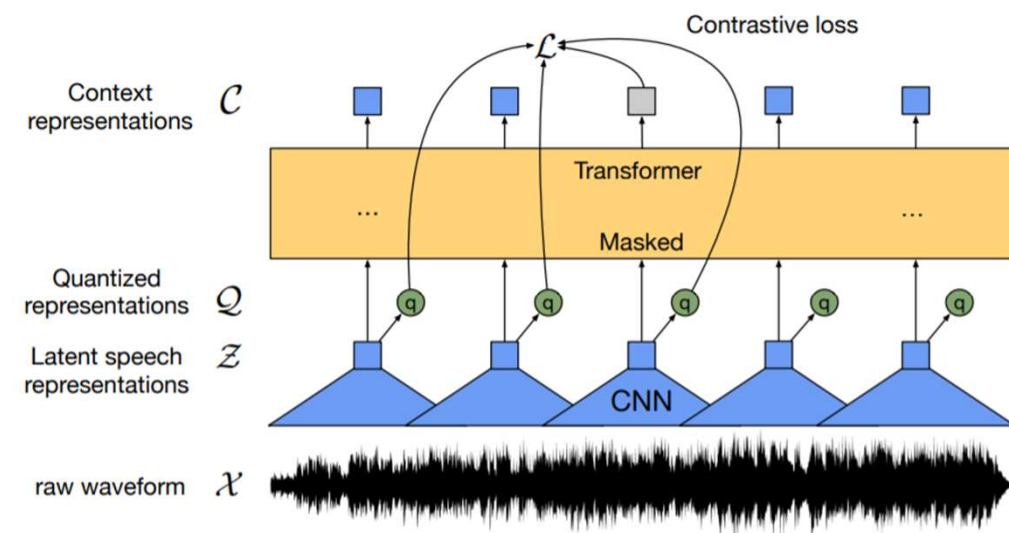


Fine-Tuning

Normal supervised training for task with low amount (e.g. 1/1000 of unlabeled data), labeled, specific data (so called downstream task).

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, NACCL 2019, Google AI

Main-dish: W2v-BERT



Auxiliary task: identify the true quantized representation from masked region (*ref2.1.1*).

G codebooks $\in \mathbb{R}^{V \times d}$, product quantization:

$\mathbf{z} \in \mathbb{R}^{ch} \xrightarrow{f(\mathbf{z})} \mathbf{I} \in \mathbb{R}^{G \times V} \xrightarrow{\text{Gum.Smax}(\mathbf{I})} \mathbf{p}_{g,v}$

select G rows e_1, \dots, e_G by $\mathbf{p}_{g,v}$ via *argmax*

concatenate into $\mathbf{e} \in \mathbb{R}^{G \times d} \xrightarrow{f(\mathbf{z})} \mathbf{q} \in \mathbb{R}^{ch}$

Gumbel softmax:
$$p_{g,v} = \frac{\exp(l_{g,v} + n_v)/\tau}{\sum_{k=1}^V \exp(l_{g,k} + n_k)/\tau},$$

τ : non-negative temperature

$n = -\log(-\log(u))$

u : uniform sample from (0,1)

Input: speech voice (1d sequence, float)

Task 1: identify the true quantized latent speech representation

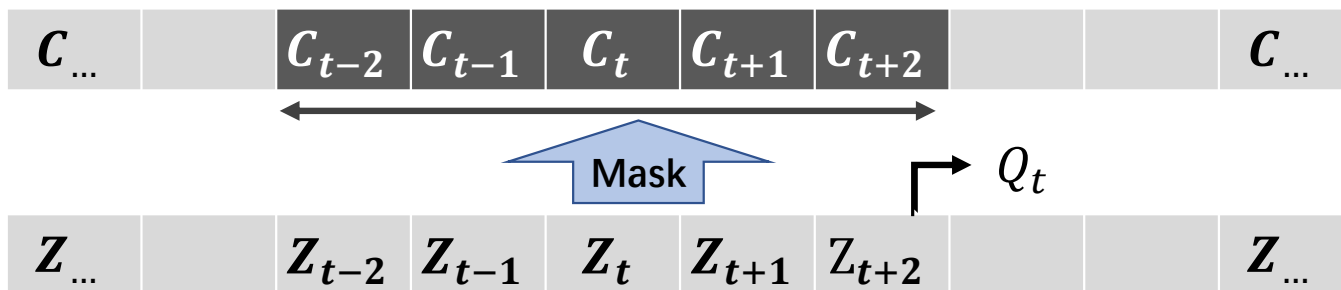
Task 2: keep vectors in codebooks used as equal as possible.

Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations, NIPS 2020, Facebook AI

Main-dish: W2v-BERT

Random mask:

replace inputs of transformer by shared, trained feature vector



Contrastive Loss (*ref2.1.2*):

$$\mathcal{L}_m = -\log \frac{\exp(\text{sim}(\mathbf{c}_t, \mathbf{q}_t)/\kappa)}{\sum_{\tilde{\mathbf{q}} \sim \mathbf{Q}_t} \exp(\text{sim}(\mathbf{c}_t, \tilde{\mathbf{q}})/\kappa)}$$

\mathbf{c} : context representation

\mathbf{q} : quantized representation

sim : cosine similarity

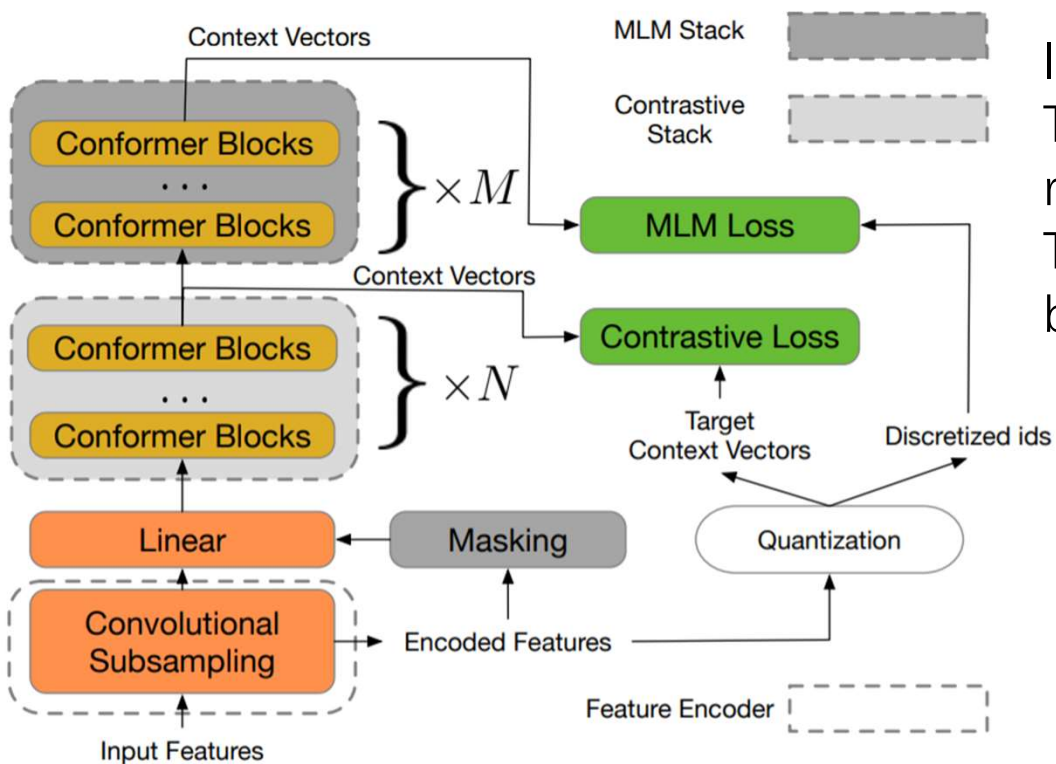
$$\text{sim}(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T \mathbf{b} / \|\mathbf{a}\| \|\mathbf{b}\|$$

$$\mathcal{L}_d = \frac{1}{GV} \sum_{g=1}^G -H(\bar{p}_g) = \frac{1}{GV} \sum_{g=1}^G \sum_{v=1}^V \bar{p}_{g,v} \log \bar{p}_{g,v}$$

Diversity Loss: encourage the equal use of the V entries in each of the G codebooks by maximizing the entropy of the averaged softmax distribution I

Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations, NIPS 2020, Facebook AI

Main-dish: W2v-BERT



Combine W2v and BERT to conduct WSL on unlabeled voice data.

Input: unlabeled voice data

Task: identify the true quantized latent speech representation (contrastive loss)

Task: randomly mask words and predict them by context (**M**asked **L**anguage **M**odeling).

Contrastive stack: transformer-based encoder

MLM stack: transformer-based decoder

W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training, 2021, MIT & Google Brain

Main-dish: W2v-BERT

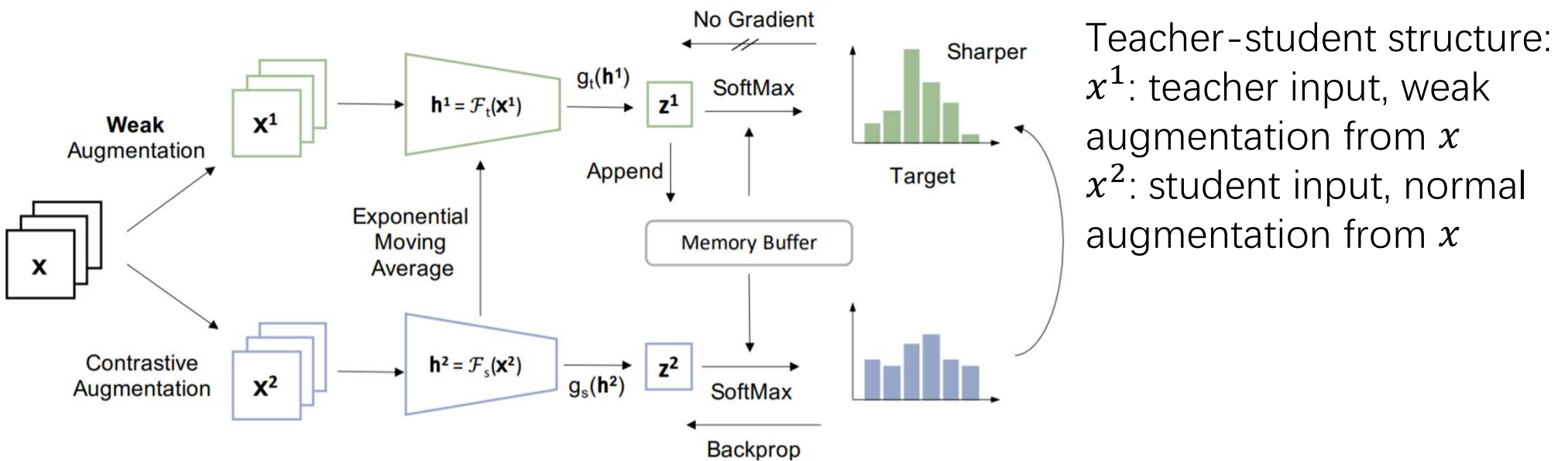
Method	Unlabeled Data (hrs)	AM Size (B)	LM Size (B)	No LM				With LM			
				dev	dev-other	test	test-other	dev	dev-other	test	test-other
Trained from Scratch											
Conformer L [21]*	N/A	0.1	0.1	1.9	4.4	2.1	4.3	—	—	1.9	3.9
Self-training Only											
Conformer L with NST [21]	60k	0.1	0.1	1.6	3.3	1.7	3.5	1.6	3.1	1.7	3.3
Pre-training Only											
wav2vec 2.0 [22]	60k	0.3	> 0.4 [†]	2.1	4.5	2.2	4.5	1.6	3.0	1.8	3.3
HuBERT Large [25]	60k	0.3	—	—	—	—	—	1.5	3.0	1.9	3.3
HuBERT X-Large [25]	60k	1.0	—	—	—	—	—	1.5	2.5	1.8	2.9
w2v-Conformer XL [21]	60k	0.6	0.1	1.7	3.5	1.7	3.5	1.6	3.2	1.5	3.2
w2v-Conformer XXL [21]	60k	1.0	0.1	1.6	3.2	1.6	3.3	1.5	3.0	1.5	3.1
w2v-BERT XL (Ours)	60k	0.6	0.1	1.5	2.9	1.5	2.9	1.4	2.8	1.5	2.8
w2v-BERT XXL (Ours)	60k	1.0	0.1	1.5	2.7	1.5	2.8	1.4	2.6	1.5	2.7
Pre-training + Self-training											
wav2vec 2.0 [22]	60k	0.3	> 0.4	1.3	3.1	1.7	3.5	1.1	2.7	1.5	3.1
w2v-Conformer XXL [21]	60k	1.0	0.1	1.3	2.7	1.5	2.8	1.3	2.6	1.4	2.7
w2v-Conformer XXL+ [21]	60k	1.1	0.1	1.3	2.7	1.5	2.7	1.3	2.6	1.4	2.6
w2v-BERT XL (Ours)	60k	0.6	0.1	1.3	2.6	1.4	2.7	1.3	2.6	1.4	2.6
w2v-BERT XXL (Ours)	60k	1.0	0.1	1.4	2.4	1.4	2.5	1.3	2.4	1.4	2.5

W2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training, 2021, MIT & Google Brain

Weakly Supervised Learning

1. Introduction
2. Theory
3. Application and SOTA works
 - Appetizer: image classification task
 - Main-dish: speech recognition task
 - Dessert: what about self-supervised learning
4. Summary

Dessert: SSL, strategy purely for pre-training



Data augmentation based **Self-Supervised Learning**: distribution of predicted classes of instances between two augmentations should be similar.
Calculate the similarity between *single* z_i^1 and *all* z_j^2 .

ReSSL: Relational Self-Supervised Learning with Weak Augmentation, NIPS 2021

Dessert: SSL, strategy purely for pre-training

Preliminaries:

Noise Contrastive Estimation

$$\mathcal{L}_{NCE} = -\log \frac{\exp(\text{sim}(\mathbf{z}^1, \mathbf{z}^2)/\tau)}{\exp(\text{sim}(\mathbf{z}_i^1, \mathbf{z}_i^2)/\tau) + \sum_{k=1}^N \exp(\text{sim}(\mathbf{z}_i^1, \mathbf{z}_k)/\tau)}$$

Reference (1) points to \mathbf{z}^1 in the numerator.
 Non-zero τ points to τ in the numerator.
 Target term (1) points to $\text{sim}(\mathbf{z}_i^1, \mathbf{z}_i^2)/\tau$ in the denominator.
 Local2global term ($1 \times N$) points to $\sum_{k=1}^N \exp(\text{sim}(\mathbf{z}_i^1, \mathbf{z}_k)/\tau)$ in the denominator.

sim: cosine similarity

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v} / \|\mathbf{u}\| \|\mathbf{v}\|$$

Weak aug. and normal aug. vs. original image:

$$\mathbf{p}_i^1 = \frac{\exp(\text{sim}(\mathbf{z}^1, \mathbf{z}_i)/\tau_t)}{\sum_{k=1}^K \exp(\text{sim}(\mathbf{z}^1, \mathbf{z}_k)/\tau_t)}, \quad \mathbf{p}_i^2 = \frac{\exp(\text{sim}(\mathbf{z}^2, \mathbf{z}_i)/\tau_s)}{\sum_{k=1}^K \exp(\text{sim}(\mathbf{z}^2, \mathbf{z}_k)/\tau_s)}$$

1-1 similarity (top arrow)
 1-all similarity (bottom arrow)

$\mathbf{z}^{1,2}$: 1st image from weak and normal aug.

\mathbf{z}_i : i -th original image in batch.

Minimize Kullback–Leibler divergence

$$\mathcal{L}_{relation} = D_{KL}(\mathbf{p}^1 || \mathbf{p}^2) = H(\mathbf{p}^1, \mathbf{p}^2) - H(\mathbf{p}^1)$$

Because $H(\mathbf{p}^1)$ is target distribution so only regress $H(\mathbf{p}^1, \mathbf{p}^2)$ hence $L = H_{CE}(\mathbf{p}^1, \mathbf{p}^2)$

ReSSL: Relational Self-Supervised Learning with Weak Augmentation, NIPS 2021

Weakly Supervised Learning

1. Introduction
2. Theory
3. Application and SOTA works
 - Appetizer: image classification task
 - Main-dish: speech recognition task
 - Dessert: what about self-supervised learning
4. Summary

WSL: a way to utilize public, unlabeled data

Methodology:

- Clustering on feature space: reduce feature distance, increase similarity, etc.
- Data augmentation: add uncertainty on certain data.

From view of business:

- Convert pre-trained network as asset: one network to multiple downstream tasks.
- Reduce labeling cost.
- Reduce data storage cost: unlabeled data can be removed after producing pre-trained network.

From view of R&D:

- Additional supervision is conducive to regress network more quickly, precisely.
- Graph based supervision positively contribute to class robustness.
- Flexible, dynamic network structure to deliver: FC layer is no longer the only choice.
- Easily used as novelty in research.