User behavior modeling

With Multi-Modal Attention-based Hierarchical Graph Neural Network (MM-AHGNN)

References:

1. *A Survey on Knowledge Graphs: Representation, Acquisition and Applications, IEEE Transactions on Neural Networks and Learning Systems 2021*
2. *Object Interaction Recommendation with Multi-Modal Attention-based Hierarchical Graph Neural Network, IEEE International Conference on Big Data 2021*

Objective:

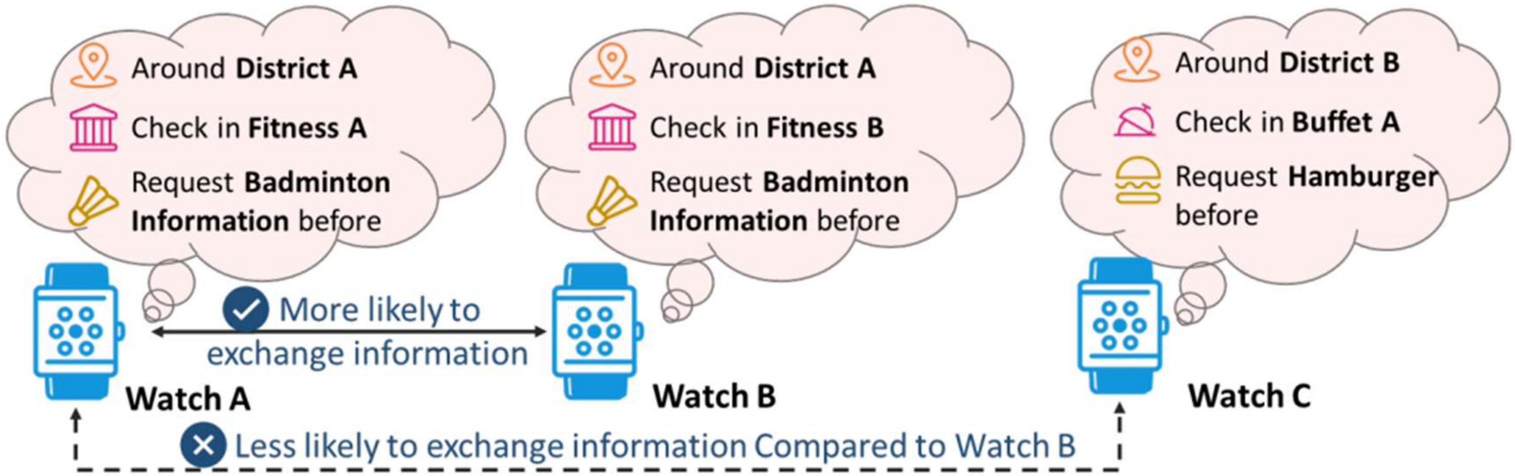
コンテンツ:

1. 領域紹介及び背景知識.
2. ビジネス側のメリット.
3. 従来手法及び問題点.
4. MM-AHGNN.

User behavior modeling

1. What is user behavior?
2. Knowledge Graphとは？どのような従来手法がある？
3. MM-AHGNN
   - GNN: 図理論で動態数のデータでのモデリング技術
   - MM-AHGNN: GNN をKnowledge Graph領域に応用
   - 性能分析
4. まとめ

行動歴史モデリングとは？



実体間の行動から実体関係を推論する手法

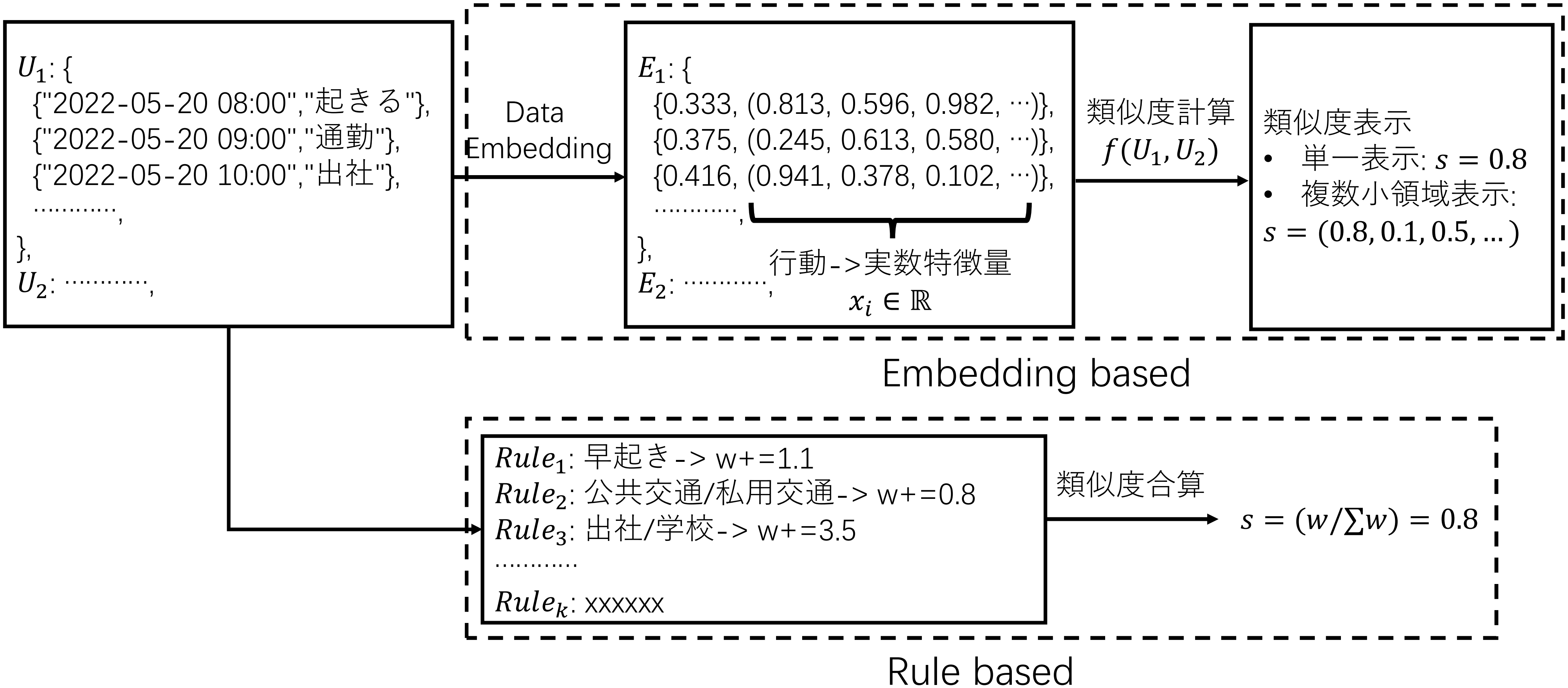
例:スマートウォッチユーザーの行動歴史
- ユーザー: {行動1,行動2, …}
- ユーザーA, Bの行動パターンが近いよって関連性が高い
- ユーザーA, Cの行動パターンが遠いよって関連性が低い

関連性の応用例**:**
- 興味推論: Aの興味≈Bの興味
- 行動予測: A,B {行動1,行動2}; A{行動3} -> B{行動3}
- ユーザーグルーピング:ユーザーを種類別分類できる

手順の全体像は？

実体間の行動から実体関係を推論する手法

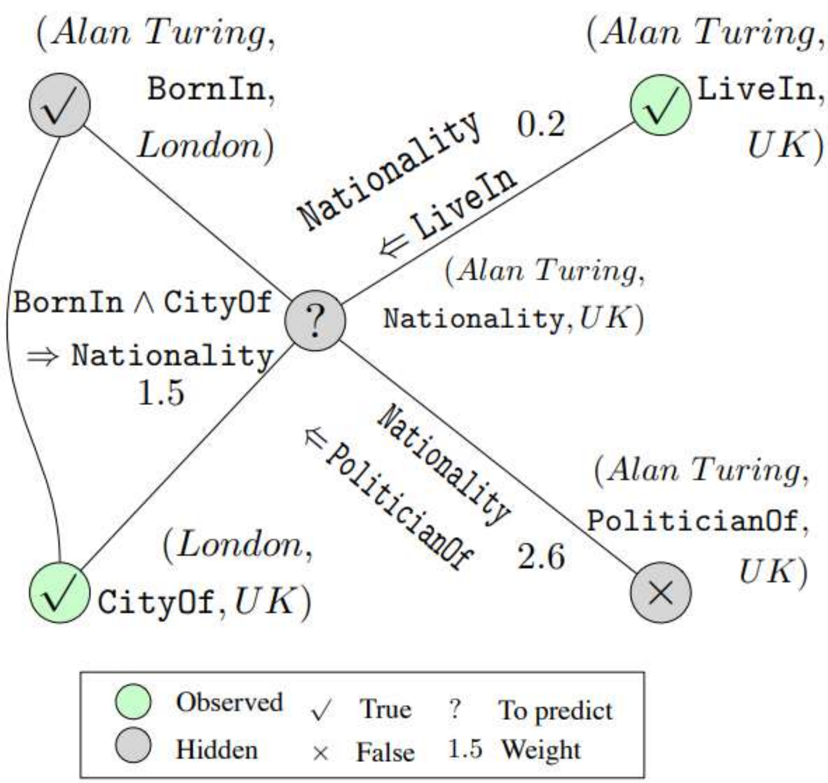
- 入力: $U_1\{A_1, \dots, A_n\}, U_2\{A_1, \dots, A_m\}, \dots$
- 出力: $s_{iht} \in S, s_{iht} = f_i(E_h, E_t)$



ユーザー行動歴史モデリング

1. 行動歴史モデリングとは？手順の全体像は？
2. Knowledge Graphとは？どのような従来手法がある？
3. MM-AHGNN
   - GNN: 図理論で動態数のデータでのモデリング技術
   - MM-AHGNN: GNN をKnowledge Graph領域に応用
   - 性能分析
4. まとめ

Knowledge Graphとは？



関連性の応用例**:**

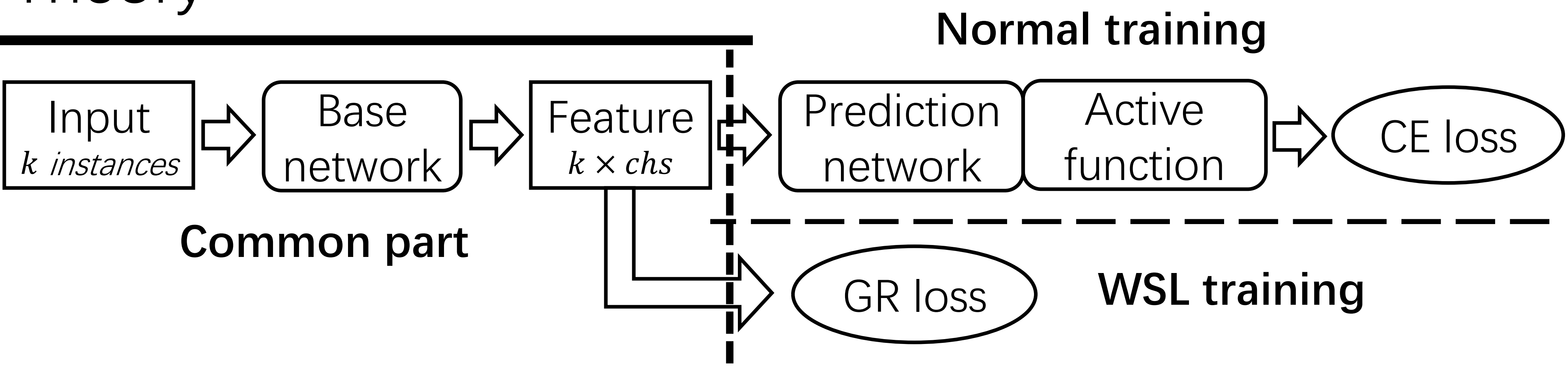
- 興味推論: Aの興味≈Bの興味
- 行動予測: A,B {行動1,行動2}; A{行動3} -> B{行動3}
- ユーザーグルーピング:ユーザーを種類別分類できる

Method	Ent. embedding	Rel. embedding	Scoring function $f_r(h,t)$	Constraints/Regularization
TransE [14]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$-\lVert\mathbf{h}+\mathbf{r}-\mathbf{t}\rVert_{1/2}$	$\lVert\mathbf{h}\rVert_2=1, \lVert\mathbf{t}\rVert_2=1$
TransH [15]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r},\mathbf{w}_r\in\mathbb{R}^d$	$-\lVert(\mathbf{h}-\mathbf{w}_r^\top\mathbf{h}\mathbf{w}_r)+\mathbf{r}-(\mathbf{t}-\mathbf{w}_r^\top\mathbf{t}\mathbf{w}_r)\rVert_2^2$	$\lVert\mathbf{h}\rVert_2\le 1, \lVert\mathbf{t}\rVert_2\le 1$ $\lvert\mathbf{w}_r^\top\mathbf{r}\rvert/\lVert\mathbf{r}\rVert_2\le\epsilon, \lVert\mathbf{w}_r\rVert_2=1$
TransR [16]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r\in\mathbb{R}^{k\times d}$	$-\lVert\mathbf{M}_r\mathbf{h}+\mathbf{r}-\mathbf{M}_r\mathbf{t}\rVert_2^2$	$\lVert\mathbf{h}\rVert_2\le 1, \lVert\mathbf{t}\rVert_2\le 1, \lVert\mathbf{r}\rVert_2\le 1$ $\lVert\mathbf{M}_r\mathbf{h}\rVert_2\le 1, \lVert\mathbf{M}_r\mathbf{t}\rVert_2\le 1$
TransD [50]	$\mathbf{h},\mathbf{w}_h\in\mathbb{R}^d$ $\mathbf{t},\mathbf{w}_t\in\mathbb{R}^d$	$\mathbf{r},\mathbf{w}_r\in\mathbb{R}^k$	$-\lVert(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}+\mathbf{r}-(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\rVert_2^2$	$\lVert\mathbf{h}\rVert_2\le 1, \lVert\mathbf{t}\rVert_2\le 1, \lVert\mathbf{r}\rVert_2\le 1$ $\lVert(\mathbf{w}_r\mathbf{w}_h^\top+\mathbf{I})\mathbf{h}\rVert_2\le 1$ $\lVert(\mathbf{w}_r\mathbf{w}_t^\top+\mathbf{I})\mathbf{t}\rVert_2\le 1$
TranSparse [51]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^k, \mathbf{M}_r(\theta_r)\in\mathbb{R}^{k\times d}$ $\mathbf{M}_r^1(\theta_r^1), \mathbf{M}_r^2(\theta_r^2)\in\mathbb{R}^{k\times d}$	$-\lVert\mathbf{M}_r(\theta_r)\mathbf{h}+\mathbf{r}-\mathbf{M}_r(\theta_r)\mathbf{t}\rVert_{1/2}^2$ $-\lVert\mathbf{M}_r^1(\theta_r^1)\mathbf{h}+\mathbf{r}-\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\rVert_{1/2}^2$	$\lVert\mathbf{h}\rVert_2\le 1, \lVert\mathbf{t}\rVert_2\le 1, \lVert\mathbf{r}\rVert_2\le 1$ $\lVert\mathbf{M}_r(\theta_r)\mathbf{h}\rVert_2\le 1, \lVert\mathbf{M}_r(\theta_r)\mathbf{t}\rVert_2\le 1$ $\lVert\mathbf{M}_r^1(\theta_r^1)\mathbf{h}\rVert_2\le 1, \lVert\mathbf{M}_r^2(\theta_r^2)\mathbf{t}\rVert_2\le 1$
TransM [52]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$-\theta_r\lVert\mathbf{h}+\mathbf{r}-\mathbf{t}\rVert_{1/2}$	$\lVert\mathbf{h}\rVert_2=1, \lVert\mathbf{t}\rVert_2=1$
ManifoldE [53]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$-(\lVert\mathbf{h}+\mathbf{r}-\mathbf{t}\rVert_2^2-\theta_r^2)^2$	$\lVert\mathbf{h}\rVert_2\le 1, \lVert\mathbf{t}\rVert_2\le 1, \lVert\mathbf{r}\rVert_2\le 1$
TransF [54]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d$	$(\mathbf{h}+\mathbf{r})^\top\mathbf{t}+(\mathbf{t}-\mathbf{r})^\top\mathbf{h}$	$\lVert\mathbf{h}\rVert_2\le 1, \lVert\mathbf{t}\rVert_2\le 1, \lVert\mathbf{r}\rVert_2\le 1$
TransA [55]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{r}\in\mathbb{R}^d, \mathbf{M}_r\in\mathbb{R}^{d\times d}$	$-(\lvert\mathbf{h}+\mathbf{r}-\mathbf{t}\rvert)^\top\mathbf{M}_r(\lvert\mathbf{h}+\mathbf{r}-\mathbf{t}\rvert)$	$\lVert\mathbf{h}\rVert_2\le 1, \lVert\mathbf{t}\rVert_2\le 1, \lVert\mathbf{r}\rVert_2\le 1$ $\lVert\mathbf{M}_r\rVert_F\le 1, [\mathbf{M}_r]_{ij}=[\mathbf{M}_r]_{ji}\ge 0$
KG2E [45]	$\mathbf{h}\sim\mathcal{N}(\boldsymbol{\mu}_h,\boldsymbol{\Sigma}_h)$ $\mathbf{t}\sim\mathcal{N}(\boldsymbol{\mu}_t,\boldsymbol{\Sigma}_t)$ $\boldsymbol{\mu}_h,\boldsymbol{\mu}_t\in\mathbb{R}^d$ $\boldsymbol{\Sigma}_h,\boldsymbol{\Sigma}_t\in\mathbb{R}^{d\times d}$	$\mathbf{r}\sim\mathcal{N}(\boldsymbol{\mu}_r,\boldsymbol{\Sigma}_r)$ $\boldsymbol{\mu}_r\in\mathbb{R}^d, \boldsymbol{\Sigma}_r\in\mathbb{R}^{d\times d}$	$-\mathrm{tr}(\boldsymbol{\Sigma}_r^{-1}(\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_t))-\boldsymbol{\mu}^\top\boldsymbol{\Sigma}_r^{-1}\boldsymbol{\mu}-\ln\frac{\det(\boldsymbol{\Sigma}_r)}{\det(\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_t)}$ $-\boldsymbol{\mu}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}-\ln(\det(\boldsymbol{\Sigma}))$ $\boldsymbol{\mu}=\boldsymbol{\mu}_h+\boldsymbol{\mu}_r-\boldsymbol{\mu}_t$ $\boldsymbol{\Sigma}=\boldsymbol{\Sigma}_h+\boldsymbol{\Sigma}_r+\boldsymbol{\Sigma}_t$	$\lVert\boldsymbol{\mu}_h\rVert_2\le 1, \lVert\boldsymbol{\mu}_t\rVert_2\le 1, \lVert\boldsymbol{\mu}_r\rVert_2\le 1$ $c_{min}\mathbf{I}\le\boldsymbol{\Sigma}_h\le c_{max}\mathbf{I}$ $c_{min}\mathbf{I}\le\boldsymbol{\Sigma}_t\le c_{max}\mathbf{I}$ $c_{min}\mathbf{I}\le\boldsymbol{\Sigma}_r\le c_{max}\mathbf{I}$
TransG [46]	$\mathbf{h}\sim\mathcal{N}(\boldsymbol{\mu}_h,\sigma_h^2\mathbf{I})$ $\mathbf{t}\sim\mathcal{N}(\boldsymbol{\mu}_t,\sigma_t^2\mathbf{I})$ $\boldsymbol{\mu}_h,\boldsymbol{\mu}_t\in\mathbb{R}^d$	$\boldsymbol{\mu}_r^i\sim\mathcal{N}(\boldsymbol{\mu}_t-\boldsymbol{\mu}_h,(\sigma_h^2+\sigma_t^2)\mathbf{I})$ $\mathbf{r}=\sum_i\pi_r^i\boldsymbol{\mu}_r^i\in\mathbb{R}^d$	$\sum_i\pi_r^i\exp\left(-\frac{\lVert\boldsymbol{\mu}_h+\boldsymbol{\mu}_r^i-\boldsymbol{\mu}_t\rVert_2^2}{\sigma_h^2+\sigma_t^2}\right)$	$\lVert\boldsymbol{\mu}_h\rVert_2\le 1, \lVert\boldsymbol{\mu}_t\rVert_2\le 1, \lVert\boldsymbol{\mu}_r^i\rVert_2\le 1$
UM [56]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	——	$-\lVert\mathbf{h}-\mathbf{t}\rVert_2^2$	$\lVert\mathbf{h}\rVert_2=1, \lVert\mathbf{t}\rVert_2=1$
SE [57]	$\mathbf{h},\mathbf{t}\in\mathbb{R}^d$	$\mathbf{M}_r^1,\mathbf{M}_r^2\in\mathbb{R}^{d\times d}$	$-\lVert\mathbf{M}_r^1\mathbf{h}-\mathbf{M}_r^2\mathbf{t}\rVert_1$	$\lVert\mathbf{h}\rVert_2=1, \lVert\mathbf{t}\rVert_2=1$

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)

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

Distance function, usually L2

Accumulate all pairs of samples

Weight, calculated by A,B, optional.

Instance A,B, selected by strategy

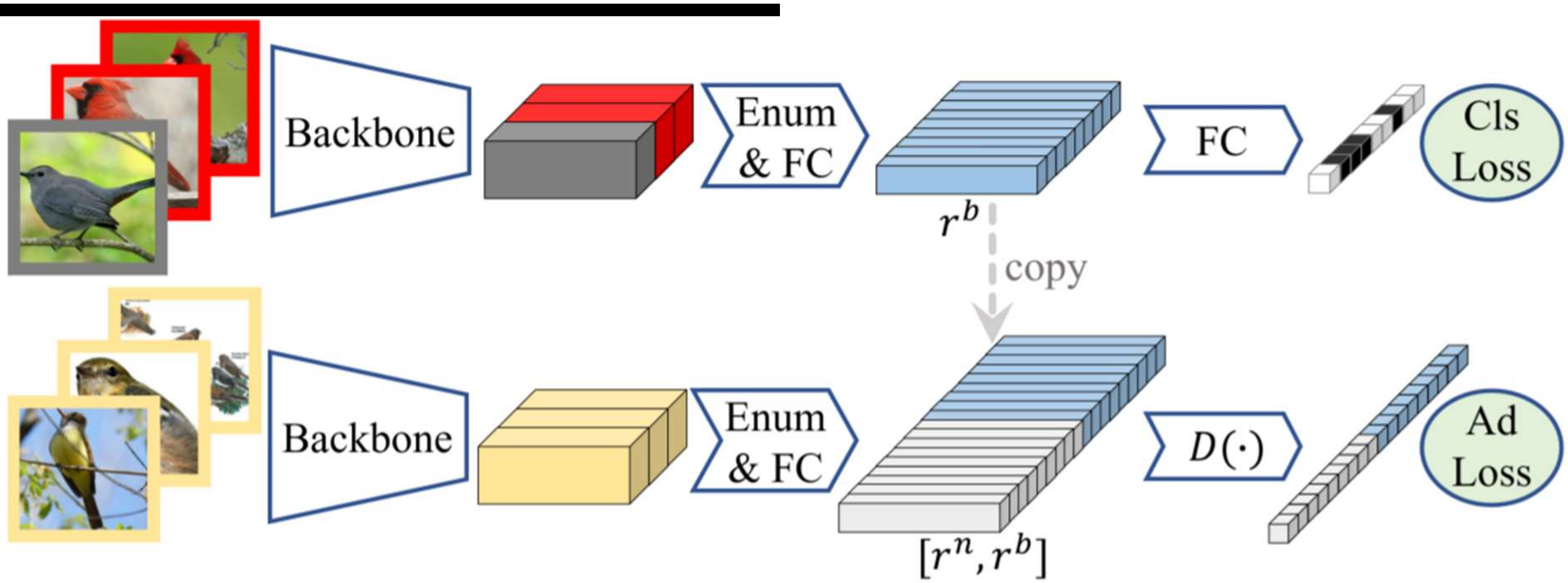
What people focus:

- 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:

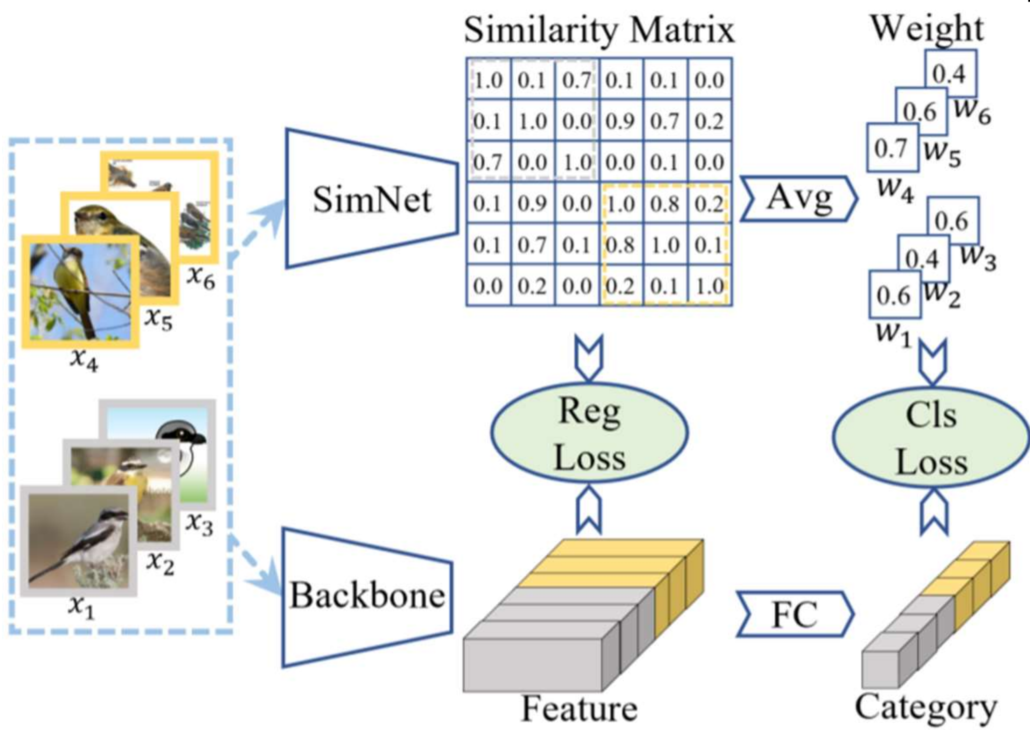
- Use *transferred similarity* + *denoise strategy* to tackle web training data.
- Apply adversarial loss to similarity net.

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



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

Training steps:

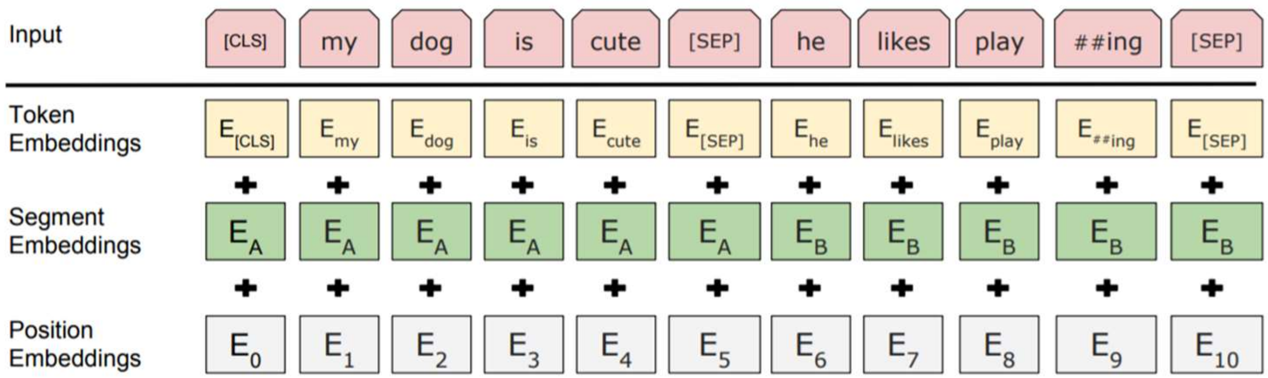
1. Train network on labeled data.
   - Prediction target: binary classification of similar or not.
   - CE loss.
   - Strong constraint: batch size k s.t. k^2 items pre-step.
2. Cluster unlabeled data via pre-trained network.
   - Select k_1 samples from labeled data.
   - Select k_2 samples from unlabeled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

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



Embedding E_*: codebook consisted by trainable parameters.

Token: vocabulary + start token + end token

Segment: (2, ch) parameters

Position: trainable parameters based on location.

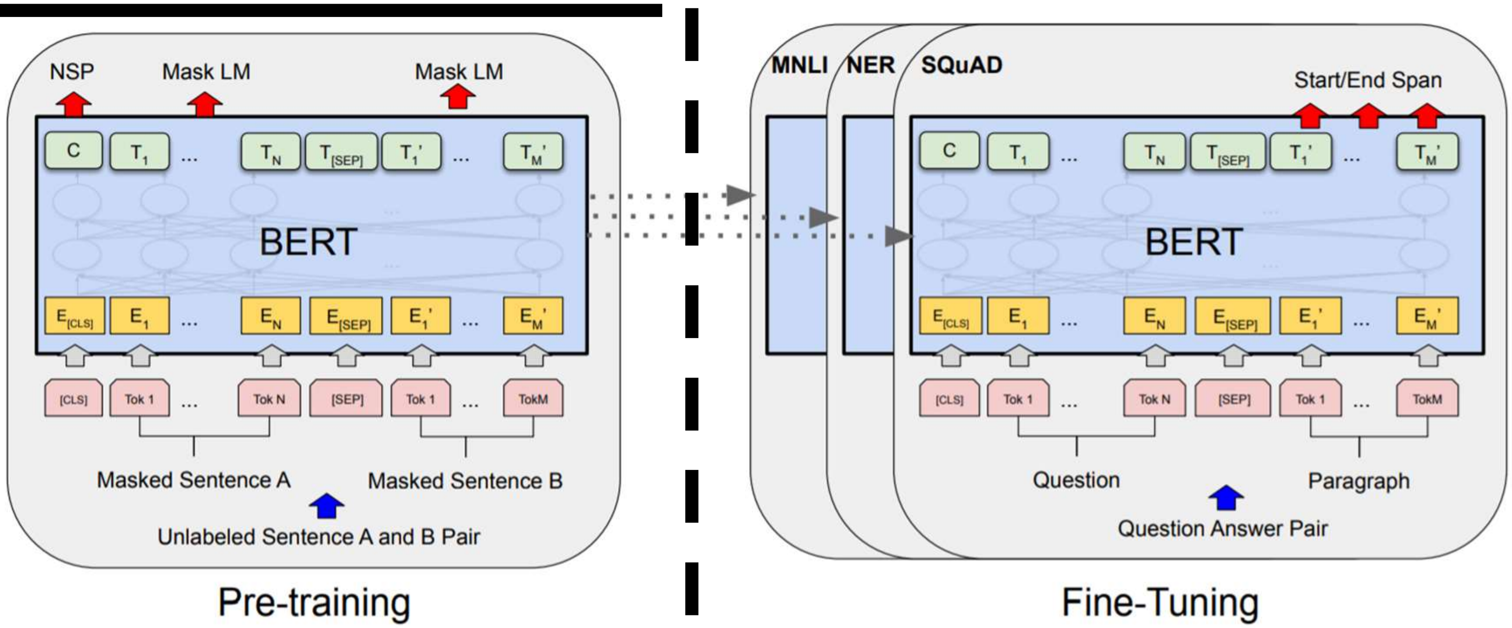
Input: 2 sentences A,B from articles.

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



Bidirectional encoder network: transformer with unfixed sequence.

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:

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

select G rows $e_1, \dots, e_G$ by $\mathrm{p_{g,v}}$ via $argmax$

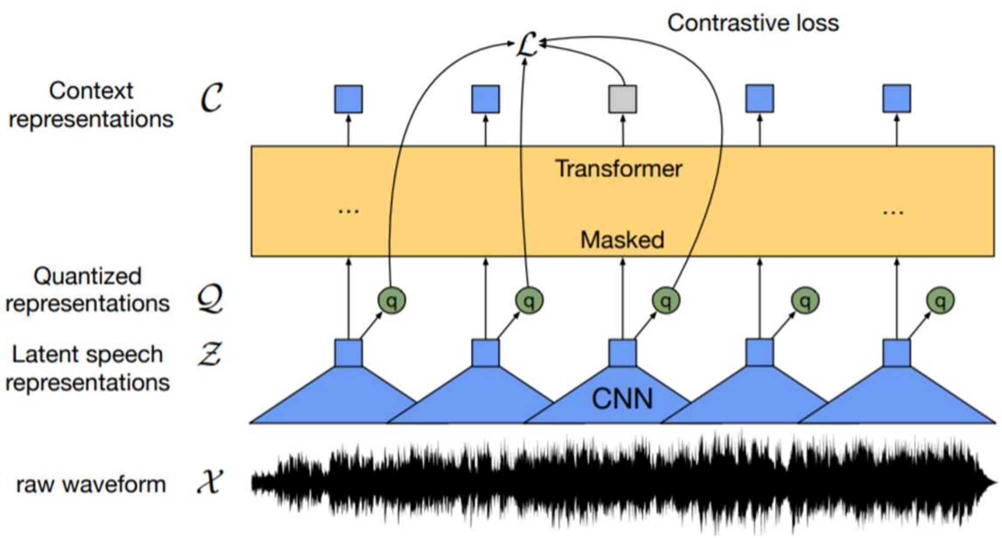
concatenate into $e \in \mathbb{R}^{G\times d} \xrightarrow{f(z)} q \in \mathbb{R}^{ch}$

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)

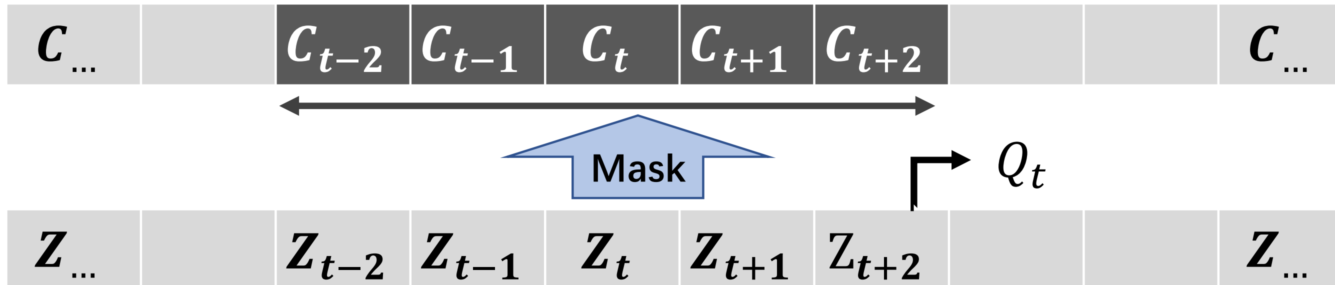
Task 1: identify the true quantized latent speech representation

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*):

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

c: context representation
q: quantized representation
sim: cosine similarity

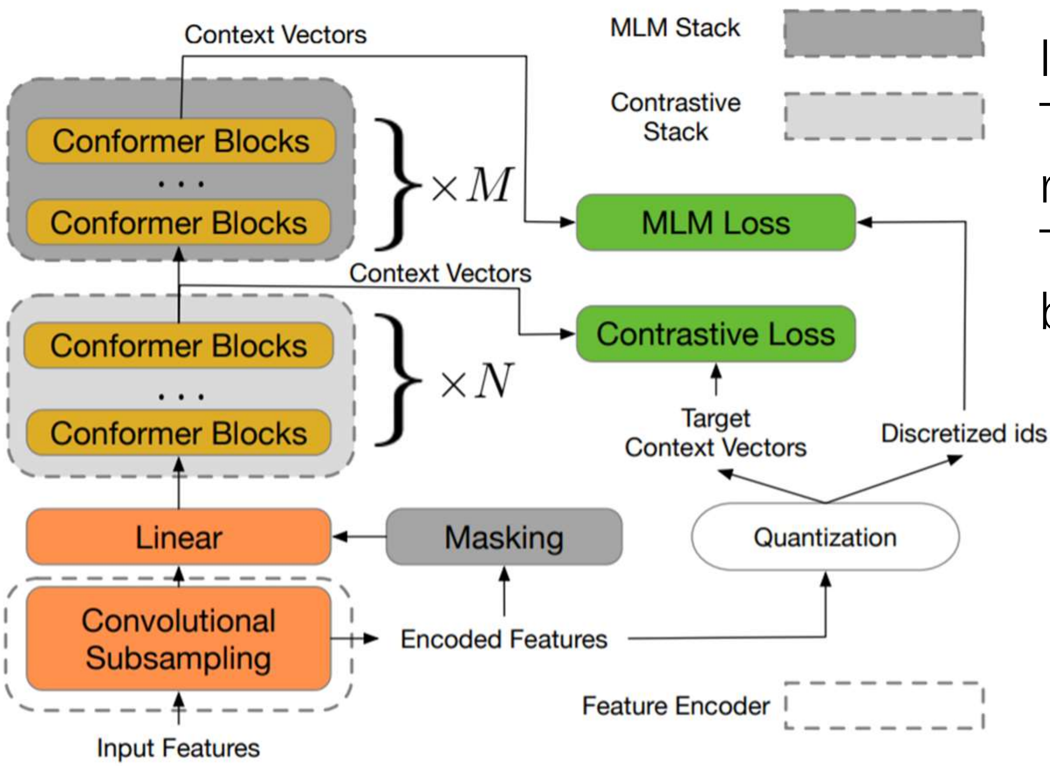
$$sim(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T\mathbf{b}/\|\mathbf{a}\|\|\mathbf{b}\|$$

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

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

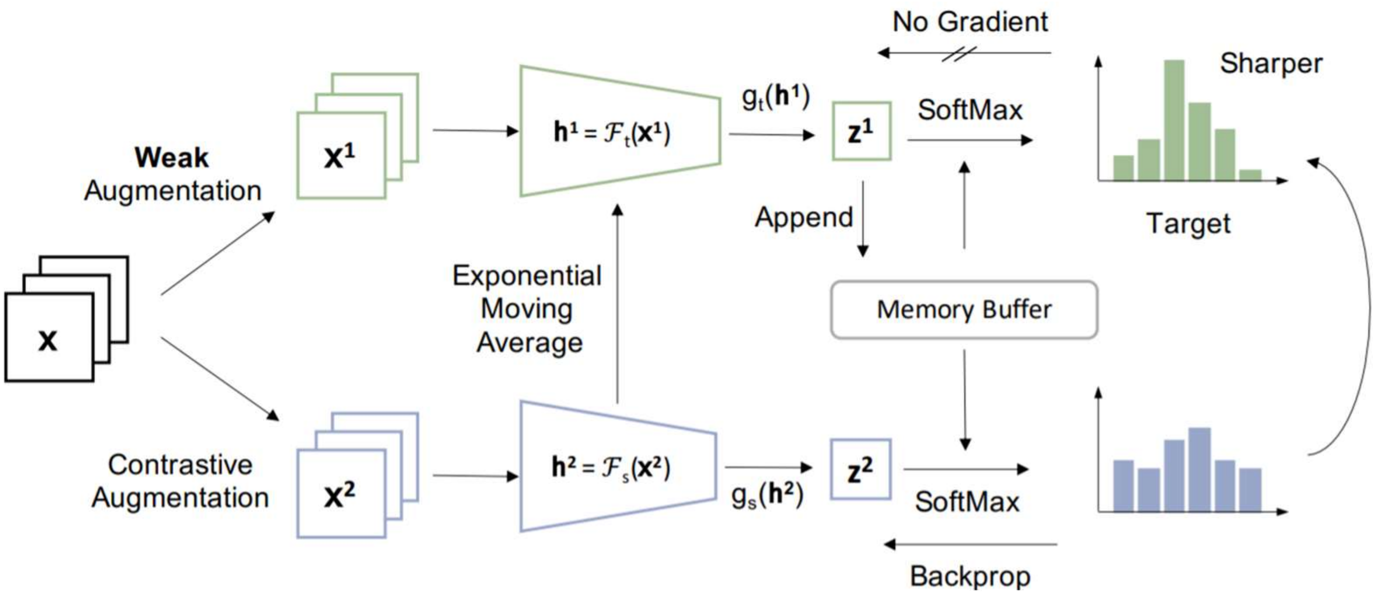
Method	Unlabeled Data (hrs)	AM Size (B)	LM Size (B)	**No LM**				**With LM**			
				dev	**dev-other**	**test**	**test-other**	**dev**	**dev-other**	**test**	**test-other**
Trained from Scratch											
Conformer L [21]*	N/A	0.1	0.1	1.9	4.4	2.1	4.3	–	–	1.9	3.9
Self-training Only											
Conformer 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	$> 0.4^\dagger$	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



Teacher-student structure:
x^1: teacher input, weak augmentation from x
x^2: student input, normal augmentation from x

Data augmentation based **S**elf-**S**upervised **L**earning: 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 **C**ontrastive **E**stimation

Reference (1)

Non-zero τ

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

Target term (1)

Local2global term $(1 \times N)$

sim: cosine similarity

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

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

1-1 similarity

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

1-all similarity

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

z_i: *i-th* original image in batch.

Minimize **K**ullback–**L**eibler divergence

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

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

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.