Dialogue Data Augmentation

Dec. 3 2020

Cao Yu

4 papers

1. Dialogue Distillation: Open-domain Dialogue Augmentation Using Unpaired Data
2. Filtering Noisy Dialogue Corpora by Connectivity and Content Relatedness
3. Sequence-Level Mixed Sample Data Augmentation (short)
4. SeqMix: Augmenting Active Sequence Labeling via Sequence Mixup

Dialogue Distillation: Open-domain Dialogue Augmentation Using Unpaired Data

Rongsheng Zhang , Yinhe Zheng, Jianzhi Shao,
Xiaoxi Mao, Yadong Xi, Minlie Huang
NetEase, THU

Main idea

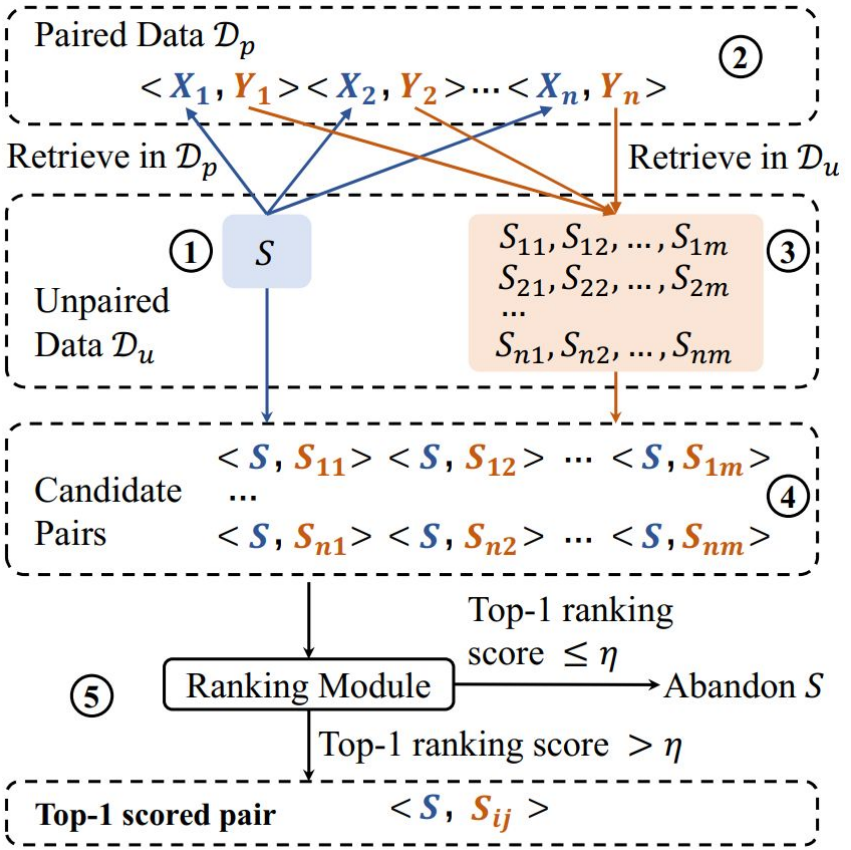
- Find an retrieve reasonable responses from unpaired data in a large-scale corpus for the given query in a existed dataset. The found responses can form some new samples to enhance the training process.
- The augmented training data is more diverse which will benefit the performance of model.

The framework

Assuming there are two sets of data, paired data $\mathcal{D}_p = \{\langle X_i, Y_i \rangle\}_{i=1}^{N}$ and unpaired data $\mathcal{D}_u = \{S_i\}_{i=1}^{M}$

There are mainly two steps:

1) finding candidate pairs in the unpaired data
2) filtering low-quality candidates and only remain the high-ranked ones.



Construct candidate pairs

1. A single sentence S is firstly randomly selected from $\mathcal{D}_u$
2. Using BM25 algorithm, treating S as a candidate query, retrieving n queries $X_i\,(1 \leq i \leq n)$ from $\mathcal{D}_p$. Then obtain n query-response pairs $\langle X_i, Y_i \rangle\,(1 \leq i \leq n)$
3. For each response Y_i, further retrieve m similar sentences from unpaired data $\mathcal{D}_u$, obtaining $n \times m$ candidates pairs $\langle S, S_{ij} \rangle,\ (1 \leq i \leq n,\ 1 \leq j \leq m)$

Filtering low-quality pairs

A BERT model is used to select which candidate pairs can benefit the model training, in other words, having high matching scores.

The BERT model is firstly fine-tuned on the paired dataset $\mathcal{D}_p$, where negative samples are obtained by replacing original response with a randomly selected one.

For each selected sentence S from unpaired, only the pair with the top-1 score from $n \times m$ candidate pairs whose score is also higher than a threshold will be added to the augmented dataset $\mathcal{D}_a$

Combine the dialog-distillation loss with model

1. For retrieval-based model. The matching loss and knoledge distillation (KD) loss are

$$\mathcal{L}_{m-nll}(\theta) = -(1-l)\log\mathcal{P}_\theta(0|X,Y) - l\log\mathcal{P}_\theta(1|X,Y)$$

$$\mathcal{L}_{m-kd}(\theta) = -\sum_{i=0}^{1}\mathcal{P}_{\theta_t}(i|X,Y)\cdot\log\mathcal{P}_\theta(i|X,Y)$$

The final mathcing model is trained using loss

$$\mathcal{L}_M(\theta) = \mathcal{L}_{m-nll}(\theta) + \alpha_m\mathcal{L}_{m-kd}(\theta)$$

2. For generation-based model, the generation loss and the KD loss are

$$\mathcal{L}_{g-nll}(\phi) = -\sum_{i=1}^{|Y|}\log P_\phi(y_i|y_{<i},X)$$

$$\mathcal{L}_{g-kd}(\phi) = -\sum_{i=1}^{|Y|}\sum_{j=1}^{|\mathcal{V}|}P_{\phi_t}(y_i=j|y_{<i},X) \times \log P_\phi(y_i=j|y_{<i},X),$$

The model is trained using loss

$$\mathcal{L}_G(\phi) = \mathcal{L}_{g-nll}(\phi) + \alpha_g\mathcal{L}_{g-kd}(\phi)$$

Experiment

Data: The method is evaluated on data collected from Weibo, in which $\mathcal{D}_p$ contains 300K pairs and $\mathcal{D}_u$ contains 2M sentences.

Models: For retrieval-based, the matching model is directly used a the teacher model for KD and the final model is trained on $\mathcal{D}_p \bigcup \mathcal{D}_a$. For generation-based model, a GPT-based Encoder-decoder model is firstly trained on $\mathcal{D}_p$ using NLL loss as the teacher model and then the final model with the same architecture is trained using the combined loss.

Baselines: it considers CAVE, ,Back-translation (BT), Sampling pairs from $\mathcal{D}_p$ and retrieve a best-matched pair from $\mathcal{D}_u$ (SP).

The main results are shown as below. It conducts evaluation using Distinct-X for diversity, Novelty-X for the n-gram diversity of new obtained samples and human evaluation. η is the filtering threshold. It can be found that the method can benefit the diversity. But it may not be good for generation coherence when η is small.

Model	Distinct-1,2,3,4				Novelty-1,2,3,4				Flu.	Coh.
CVAE	0.178‡	09.40‡	34.54‡	60.73‡	00.25‡	08.47‡	25.45‡	40.62‡	1.529‡	0.862‡
BT	0.193‡	12.42‡	43.43‡	70.38‡	03.07‡	21.66‡	35.28‡	45.18‡	1.771‡	1.408†
SP	**0.228**	11.56‡	37.76‡	57.73‡	18.48‡	46.65‡	73.56‡	87.79‡	1.839‡	0.777‡
DL η=0.90	0.226‡	**13.72**	**48.24**	**76.21**	**23.76**	**55.95**	**80.64**	**92.10**	1.835‡	1.183‡
DL η=0.95	0.224‡	13.44‡	47.51‡	75.55‡	22.81‡	55.51‡	80.37‡	91.97‡	1.856†	1.358‡
DL η=0.99	0.213‡	12.61‡	45.06‡	72.87‡	21.59‡	54.40‡	79.69‡	91.62‡	**1.877**	**1.428**
$\mathcal{D}_p$(human)	0.199	13.51	47.70	75.52	N/A				1.868	1.617

It also conducts experiment on some variants. Teacher: only train a teacher model on the paired data. AP: train the model only on augmented data with NLL loss. UP: firstly fine-tune model on unpaired data, then using the weights to initialize the final model trained on u$\mathcal{D}_p$g NLL loss. NP+ML: randomly sampling 300k pairs from Weibo as the augmented data. DL+ML: the proposed method. DL+PreT: first train the model on tl$\mathcal{D}_a$ on . $\mathcal{D}_p$

w/o ML: without model KD loss, w/o DL: without data distillation (augmentation), w/o PD: without original paired data $\mathcal{D}_p$, w/o Ranking: candidates are directly used without filtering.

Model	MAP	R_{10}@1	R_{10}@2	R_{10}@5
Teacher	80.2	69.7	82.1	95.1
AP	76.5	65.1	78.0	92.1
UP+PreT	80.6	70.3	82.6	**95.3**
NP+ML	80.8	70.5	82.9	95.2
CVAE+ML	80.3	69.8	82.5	94.9
BT+ML	80.3	69.8	82.0	95.2
SP+ML	80.4	70.0	82.0	95.2
DL+PreT	80.7	70.2	82.7	**95.3**
DL+ML	**81.0**	**70.8**	**83.1**	**95.3**
w/o ML	80.4	69.9	82.5	95.0
w/o DL	80.5	70.1	82.3	95.1
w/o PD	79.5	68.9	81.3	94.1
w/o Ranking	80.5	70.1	82.5	95.2

Model	PPL	BLEU-1,2		Dist.-1,2	
Teacher	23.9‡	12.25‡	6.61‡	3.83‡	29.69‡
AP	50.0‡	10.86‡	5.52‡	3.29‡	23.37‡
UP+PreT	24.0‡	12.60	6.81†	3.99‡	30.50‡
NP+ML	23.1‡	11.63‡	6.25‡	3.99‡	28.47‡
CVAE+ML	23.9‡	12.27‡	6.59‡	3.73‡	26.75‡
BT+ML	23.8‡	11.93‡	6.48‡	3.84‡	27.38‡
SP+ML	23.6‡	12.47‡	6.74‡	4.04	30.66‡
DL+PreT	23.7‡	**12.66**	6.92	3.95‡	30.30‡
DL+ML	**22.6**	12.42‡	**6.93**	**4.13**	**31.39**
w/o ML	23.3‡	12.30‡	6.65‡	4.06	30.89‡
w/o DL	23.5‡	12.54†	6.88	3.96‡	29.79‡
w/o PD	26.7‡	11.08‡	5.86‡	3.48‡	26.84‡
w/o Ranking	22.8‡	12.54‡	6.78‡	3.90‡	28.93‡

Filtering Noisy Dialogue Corpora by Connectivity and Content Relatedness

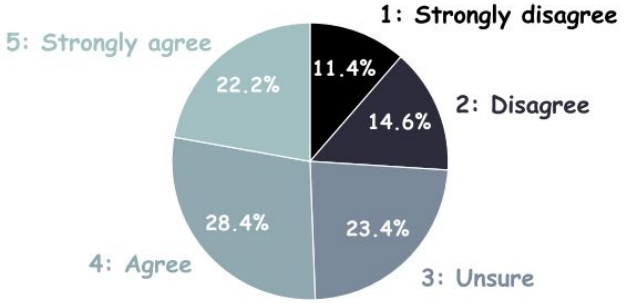
Reina Akama , Sho Yokoi, Jun Suzuki, Kenrato Inui
Tohoku University, RIKEN

Main idea

- Some dialogue datasets are noisy (e.g. OpenSubtitles) so it is worthwhile to make it applicable for training neural models with satisfactory performance.
- This paper proposes a noisy data filtering method based on the connectivity (key phrases) and content relatedness (topic commonality), which is used to remove the unacceptable utterance pairs. It is a statistical method.
- The method shows high consistency with human evaluation as well as results in a better performance of generation model.

Human evaluation on noisy data

- Human evaluation is firstly conducted on OpenSubtitles dataset to check the acceptance status of utterances.
- It found that only half samples are acceptable. And it also found that utterance pair with high score usually have some ***specific patterns*** (e.g. (why, because), (what do you want, I want)) or on ***the same topic***. These forms the intuition of their method.



	Utterance	Response	Human
1:	It'll be like you never left. [??]	I painted a white line on the street way over there. [painting]	1.4
2:	You're gonna get us assimilated. [??]	Switch to a garlic shampoo. [??]	1.8
3:	I probably asked for too much money. [money]	Money's always a problem, isn't it? [money]	4.2
4:	I wonder who I should call back. [phone]	They're saying they want to call one of you back. [phone]	4.4
5:	Okay, so where's the rest? [??]	Electronically scanned and archived at headquarters but you'll have to speak with them about that. [work]	4.4

Data filtering method

Given a utterance pair (x, y)

1. Connectivity: f and e are phreases obtained from x and y respectively. f is al$\phi(x, y)$ə (n-gram) pairs from (x, y), is all phı$\overline{\mathcal{P}}_{\mathcal{D}}$ə pairs obtained from entire dataset. It uses a phrase extraction technique from SMT (e.g. Moses) to obtain the subset of , which are ¸$\mathcal{P}$ıra$\overline{\mathcal{P}}_{\mathcal{D}}$ that contributes to the connectivity.

The connectivity is estimated via

$$S_{\mathbf{C}}(x, y) := \sum_{(f,e)\in\phi(x,y)\cap\mathcal{P}} \max\big(\mathrm{nPMI}(f, e), 0\big) \cdot \frac{|f|}{|x|} \cdot \frac{|e|}{|y|}$$

nPMI is normalized pointwise mutual information to ensure that the low-frequency phrases do not take very large values.

2. Content relatedness: Let $\boldsymbol{v}(x)$ and $\boldsymbol{v}(y)$ as the sentence vector of x and y, which is obtained via the mean value of token-level FastText embedding, the content relatedness is calculated via

$$S_{\mathbf{R}}(x, y) := \max\big(\cos(\boldsymbol{v}(x), \boldsymbol{v}(y)), 0\big)$$

It uses cosine simialrity to measure the topic relatedness.

The final score is the combination of two scores.

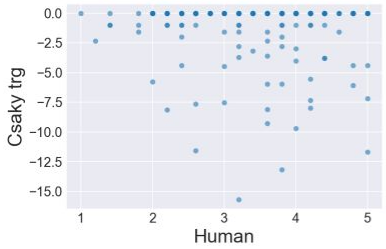
$$S_{\mathbf{C+R}}(x, y) := \alpha S_{\mathbf{C}}(x, y) + \beta S_{\mathbf{R}}(x, y) \qquad \alpha = \frac{1}{\frac{1}{|\mathcal{D}|}\sum_{(x,y)\in\mathcal{D}} S_{\mathbf{C}}(x, y)},\ \beta = \frac{1}{\frac{1}{|\mathcal{D}|}\sum_{(x,y)\in\mathcal{D}} S_{\mathbf{R}}(x, y)}$$

Each score will be normalized by the summation of all scores on the whole dataset. Utterance pairs with low scores will filtered in experiment.

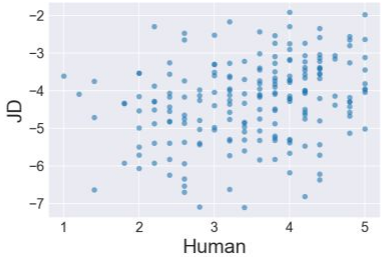
Human evaluation consistency experiment

- It first test the consistency between the score and human evaluations.
- Two baselines: 1) Token-level entropy of source/target utterance, which is used to rank the pair (Casky SRG/TRG); 2) A conditional cross-entropy computed based on a neural encoder-decoder model. originally for NMT (Junczys)
- The proposed $S_{\mathbf{C+R}}$ shows higher correlation with subjective evaluation.

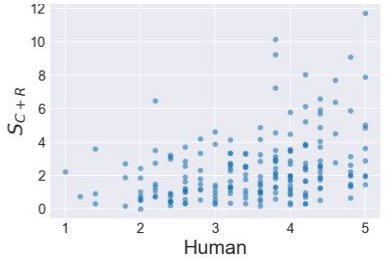
Scoring method	Spearman's ρ	p-value
Csáky et al. (2019) SRC	-0.1173	9.8×10^{-2}
Csáky et al. (2019) TRG	0.0462	5.2×10^{-1}
Junczys-Dowmunt (2018)	0.2973	1.9×10^{-5}
Ours $S_{\mathbf{C+R}}$	**0.3751**	$\mathbf{4.4 \times 10^{-8}}$
Ours $S_{\mathbf{C}}$ (ablation study)	0.2044	3.7×10^{-3}
Ours $S_{\mathbf{R}}$ (ablation study)	0.3007	1.5×10^{-5}



(a) Csáky et al. (2019) TRG



(b) Junczys-Dowmunt (2018)



(c) Ours $S_{\mathbf{C+R}}$

Some examples are given below, which proves that the combination of two scores is better than a single score.

	Utterance	Response	$S_{\mathbf{C}}$	$S_{\mathbf{R}}$	$S_{\mathbf{C+R}}$	Human
1 :	What is the anarchy facing the jail of the sick passion?	Gosh, it's really cold!	0.32	0.00	**0.32**	1.4
2 :	Pushers won't let the junkie go free.	Across 110th Street.	0.00	0.42	**0.42**	2.4
3 :	It started when I was 17.	They'd make a cash drop,	0.63	0.00	**0.63**	2.0
4 :	A big nail should be put in your head	Who are they	0.74	0.00	**0.74**	1.2
5 :	He told me so.	Oh, he did, huh?	2.21	0.00	**2.21**	4.8
6 :	There's a laundry.	Have your clothes dry-cleaned, okay?	0.81	2.89	**3.70**	4.4
7 :	Then if I win, what are you going to do?	When you win?	1.04	7.01	**8.05**	4.2
8 :	But what do you want me to do?	We want you to kick her off the team.	10.20	1.53	**11.72**	5.0

And such filtering will not affect the diversity of samples after filtering.

Scored data	len	distinct-1	distinct-2
Top 50% (remained)	9.02	0.028	0.472
Worst 50% (removed)	9.00	0.030	0.470

Experiment on neural models

Transformer-based encoder-decoder model is used as the base model. Model will be trained on non-filtered data and data after filtering using different methods. English OpenSubtitles is used as the dataset. ✘/✔ means the ratio of low/high scored responses by human. The model trained on data filtered by this method is superiority to other methods especially on diversity.

Training data	# of pairs	Automatic evaluation				Human evaluation		
		len	distinct-1	distinct-2	BLEU-1	Avg.	✘$^{\downarrow}$	✔$^{\uparrow}$
non-filtered	79,445,453	8.44	127/0.030	238/0.064	8.8	3.37	38 %	62 %
Csáky et al. (2019) SRC	40,000,000	7.97	165/0.041	329/0.094	9.1	3.56	25 %	75 %
Csáky et al. (2019) TRG	40,000,000	18.25	213/0.023	591/0.069	5.4	2.85	65 %	35 %
Junczys-Dowmunt (2018)	40,000,000	**8.63**	206/0.048	478/0.125	**9.4**	3.43	32 %	68 %
Ours $S_{\mathbf{C+R}}$	40,000,000	7.13	**345/0.097**	**853/0.278**	**9.4**	3.73	**15 %**	**85 %**
Ours $S_{\mathbf{C}}$ (ablation study)	40,000,000	7.31	201/0.055	466/0.148	9.2	3.69	19 %	81 %
Ours $S_{\mathbf{R}}$ (ablation study)	40,000,000	7.91	270/0.068	662/0.192	**9.4**	**3.76**	20 %	80 %
reference		9.04	1301/0.288	3244/0.807	-	-	-	-

Conclusion

This paper proposes a simple dialogue data filtering (augmentation) method purely based on the statistics of utterance pairs, whose intuition comes from the subjective experiment on the acceptability of utterance pairs in existed dataset. It proves its effectiveness on noisy data.

Data augmentation is not only increasing sample numbers but also removing distractor samples to improve the data quality.

Sequence-Level Mixed Sample Data Augmentation

Demi Guo, Yoon Kim, Alexander M. Rush
Harvard, MIT-IBM, Cornell

Main idea

This paper proposes a text sequence augmentation method SeqMix, in which a new sample is crafted by softly mixing two sentences via a convex combination of the original examples. It is a simple method that may be effective for various language applications.

It can be regarded as a sequence-level variant of MixUp approach which has been used in image classification.

Method

Let $X \in \mathbb{R}^{s \times V}$ and $Y \in \mathbb{R}^{t \times V}$ be the source and target sequence in which V is the vocabulary size. Then a binary combination vector is sampled where $m_X \in \{0,1\}^s, m_Y \in \{0,1\}^t$, each element satisfies Bernoulli distribution.

The new sample is obtained via

$$(\hat{X}, \hat{Y}) = (m_X \odot X + (1 - m_X) \odot X', \\ m_Y \odot Y + (1 - m_Y) \odot Y').$$

Such a new sample may contain valid subparts for model to learn compositional structure.

The training objective is below, where $f_\theta(X, Y_{<t})$ is the log-softmax layer

$$\mathcal{L} \approx \mathop{\mathbb{E}}_{\substack{(X,Y)\sim D \\ (X',Y')\sim D'}} \left[\sum_{t=1}^{T} \mathbb{E}[\hat{Y}_t]^\top f_\theta \left(\mathbb{E}[\hat{X}], \mathbb{E}[\hat{Y}_{<t}] \right) \right]$$

In fact, the soft version of the expected sample is

$$(\mathbb{E}[\hat{X}], \mathbb{E}[\hat{Y}]) = (\lambda X + (1-\lambda)X', \lambda Y + (1-\lambda)Y').$$

Here λ is the parameter of Bernoulli distribution sampled from Beta distribution.

In fact, the new sample can be regarded as a weighted sum between two samples. It also lists the difference between it and previous methods.

Method	Intuition	Combination vector $m \sim p_\lambda(m)$	$(x', y') \sim D'$	Relaxed
WordDrop	*Drop words at random*	Fixed hyperparameter ρ, $p_\lambda(m_i)$ $m_i \sim p_\lambda(m_i) \propto \text{Bernoulli}(1-\rho)$	$D' =$ zero vectors	N
SwitchOut	*Random words by position*	$\lambda \sim p(\lambda) \propto e^{-\lambda/\eta}$, $\lambda = \{0, \cdots, s\}$, $m_i \sim p_\lambda(m_i) \propto \text{Bernoulli}(1-\lambda/s)$	$D' =$ vocabulary	N
GECA	*Enumerate valid swaps*	$x_{i:j} = x'_{i':j'}$ if $x_{i:j}$ and $x'_{i':j'}$ is a valid swap (i.e. co-occurs in context)	$D' =$ training	N
SeqMix (Hard)	*Random hard swaps*	$\lambda \sim \text{Beta}(\alpha, \alpha)$, $m_i \sim p_\lambda(m_i) \propto \text{Bernoulli}(\lambda)$	$D' =$ training	N
SeqMix	*Random soft swaps*	$\lambda \sim \text{Beta}(\alpha, \alpha)$, $p_\lambda(m_i) \propto \text{Bernoulli}(\lambda)$, $m_i = \mathbb{E}[m_i] = \lambda$	$D' =$ training	Y

Experiment

It is tested on machine translation datasets, command execution dataset (SCAN) and semantic parsing dataset (SQL Queries), compared with other augmentation baselines. GECA is a method that enumerate valid swaps in text-piece level.

$x_{i:j} = x'_{i':j'}$ if $x_{i:j}$ and $x'_{i':j'}$ is a valid swap (i.e. co-occurs in context)

	IWSLT				*WMT*	*SCAN*			*SQL Queries*	
	de-en	en-de	en-it	en-es	en-de	jump	around-r	turn-l	query	question
w/o GECA										
Baseline	34.7	28.5	30.6	36.2	27.3	0%	0%	49%	39%	68%
WordDrop	35.6	29.2	31.1	36.4	27.5	0%	0%	51%	27%	66%
SwitchOut	35.9	29.0	31.3	36.4	27.6	0%	0%	16%	39%	67%
SeqMix (Hard)	35.6	28.9	30.8	36.3	27.6	19%	0%	53%	35%	68%
SeqMix	**36.2**	**29.5**	**31.7**	**37.3**	**28.1**	**49%**	0%	**99%**	**43%**	68%
w/ GECA										
Baseline (Andreas, 2020)						87%	82%	-	49%	68%
WordDrop						51%	61%	-	47%	67%
SwitchOut						77%	73%	-	50%	67%
SeqMix (Hard)						81%	82%	-	51%	68%
SeqMix						**98%**	**89%**	-	**52%**	68%

SeqMix: Augmenting Active Sequence Labeling via Sequence Mixup

Rongzhi Zhang, Yue Yu, Chao Zhang
Georgoa Tech

Main idea

Similar to the former paper, it is also a MixUp paper. The difference is that this paper introduce more complex MixUp strategy (whole sequence-level, subsequence-level, and label constrained subsequence-level), as well as a scoring function to determine whether current text will be augmented.

The work is targeted on sequence-labeling task (e.g. NER), but it can also be used on generation tasks. It considers low-resource setting so the samples will be augmented iteratively.

Sequence mixup in embedding space

Given two sequence $\mathbf{x_i} = \{\mathbf{w}_i^1, \cdots, \mathbf{w}_i^T\}$, $\mathbf{x_j} = \{\mathbf{w}_j^1, \cdots, \mathbf{w}_j^T\}$ and their corresponding embeddings $\mathbf{e_{x_i}} = \{\mathbf{e}_i^1, \cdots, \mathbf{e}_i^T\}$, $\mathbf{e_{x_j}} = \{\mathbf{e}_j^1, \cdots, \mathbf{e}_j^T\}$, and the vocabulary $\mathcal{W}$ and all embedding $\mathcal{E}$ obtained from BERT, the mixed token is

$$\mathbf{e}^t = \arg\min_{\mathbf{e} \in \mathcal{E}} \left\| \mathbf{e} - (\lambda \mathbf{e_i^t} + (1-\lambda)\mathbf{e_j^t}) \right\|_2$$

The label for two sequences are $\{\mathbf{y}_i^1, \cdots, \mathbf{y}_i^T\}$ and $\{\mathbf{y}_j^1, \cdots, \mathbf{y}_j^T\}$, the mixed label is

$$\mathbf{y}^t = \lambda \mathbf{y}_i^t + (1-\lambda)\mathbf{y}_j^t$$

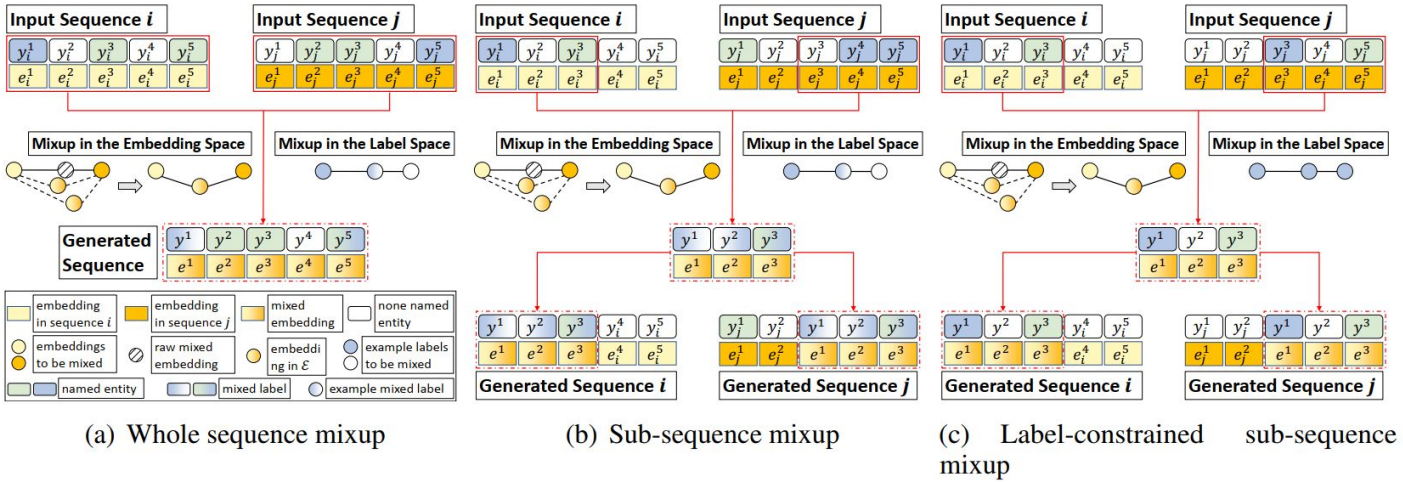
To select paired sequence, it measure the label density for each sequence, $\zeta(\cdot)$ means the ratio of valid labels among all labels. Only sequence with density higher than a threshold will be regarded as a candidate $\eta \geq \eta_0$.

Three kinds of sequence mixup strategies

1.Whole sequence mixup: given two sequences $\langle \mathbf{x_i}, \mathbf{y_i} \rangle$, $\langle \mathbf{x_j}, \mathbf{y_j} \rangle$with the same length and satisfy $\eta \geq \eta_0$, all tokens will be performed mixup.

2. Subsequence mixup: given a window with a fixed-length s, two series of sub-sequences, $\mathbf{X}_{isub} = \{\mathbf{x}^1_{isub}, \ldots, \mathbf{x}^s_{isub}\}$nd $\mathbf{X}_{jsub} = \{\mathbf{x}^1_{jsub}, \ldots, \mathbf{x}^s_{jsub}\}$very sub-sequence pair satisfies label density will be mixed.

3.label-constrained subsequence: similar to 2 but the label must be the same.



(a) Whole sequence mixup (b) Sub-sequence mixup (c) Label-constrained sub-sequence mixup

Scoring and selecting plausible seqneces

To ensure the quality of generated mixed sequence, it introduces a scoring function that uses the perplexity given by a GPT2 on the augmented sequence.

$$\text{Perplexity}(\mathbf{x}) = 2^{-\frac{1}{\mathrm{T}}\sum_{\mathrm{i}=1}^{\mathrm{T}}\log \mathrm{p}(\mathrm{w_i})}$$

$$d(\mathbf{x}) = \mathbb{1}\left\{s_1 \leq \text{Perplexity}\left(\mathbf{x}\right) \leq \mathrm{s}_2\right\}$$

Only the new sequence with a PPL within a specific range will be put into the augmented set for training.

The algorithm to obtain the augmented samples using SeqMix is shown on the right.

Algorithm 2 The generation procedure of SeqMix

Input: Labeled set $\mathcal{L} = \langle \mathcal{X}, \mathcal{Y} \rangle$; Beta distribution parameter α; Pairing function $\zeta(\cdot)$; Discriminator function $d(\cdot)$; Number of expected generation N.

for $\langle \mathbf{x_i}, \mathbf{y_i} \rangle, \langle \mathbf{x_j}, \mathbf{y_j} \rangle, (i \neq j)$ *in* $\mathcal{L}$ **do**
  if $\zeta(\langle \mathbf{x_i}, \mathbf{y_i} \rangle, \langle \mathbf{x_j}, \mathbf{y_j} \rangle)$ **then**
    $\lambda \sim Beta(\alpha, \alpha)$
    // mixup the target sub-sequences
    for $t = 1, \cdots, T$ **do**
      Calculate $\mathbf{e}^t$ by Eq. (7);
      Get corresponding token $\mathbf{w}^t$ for $\mathbf{e}^t$;
      Calculate $\mathbf{y}^t$ by Eq. (8).
    end
    $\tilde{\mathbf{x}}_{sub} = \{\mathbf{w}^1, \cdots, \mathbf{w}^T\}$
    $\tilde{\mathbf{y}}_{sub} = \{\mathbf{y}^1, \cdots, \mathbf{y}^T\}$
    // replace the original sequences
    for k *in* $\{i, j\}$ **do**
      $\tilde{\mathbf{x}}_k = \mathbf{x}_k - \mathbf{x}_{\mathbf{k}sub} + \tilde{\mathbf{x}}_{sub}$
      $\tilde{\mathbf{y}}_k = \mathbf{y}_k - \mathbf{y}_{\mathbf{k}sub} + \tilde{\mathbf{y}}_{sub}$
      if $d(\tilde{\mathbf{x}_k})$ **then**
        $\mathcal{L}^* = \mathcal{L}^* \cup \langle \tilde{\mathbf{x}}_k, \tilde{\mathbf{y}}_k \rangle$
      end
      if $|\mathcal{L}^*| \geq N$ **then**
        break
      end
    end
  end
end

Output: Generated sequences and labels $\mathcal{L}^*$

The whole training procedure

It considers the low-resource setting. Given a large unlabeled corpus $\mathcal{U}$ and a small annotated set $\mathcal{L}$, a query policy will firstly used to get the of the most informative samples in $\mathcal{U}$ and get their annotation by initial model. Then mixup augmented samples are added to the training set together with previously queried samples.

This procedure will be done iteratively.

Algorithm 1 The procedure of active sequence labeling augmentation via SeqMix

Input: Labeled seed set $\mathcal{L}$; Unlabeled set $\mathcal{U}$; Query function $\psi(\cdot, K, \gamma(\cdot))$; The sequence labeling model θ; Beta distribution parameter α; Pairing function $\zeta(\cdot)$; Discriminator function $d(\cdot)$.

// seed set augmentation

$$\mathcal{L}^* = SeqMix(\mathcal{L}, \alpha, \zeta(\cdot), d(\cdot))$$
$$\mathcal{L} = \mathcal{L} \cup \mathcal{L}^*$$

// model initialization

$$\theta = \text{train } (\theta, \mathcal{L})$$

// active learning iterations with augmentation

for *round* ***in*** *active learning rounds* **do**

$\mathcal{X} = \psi(\mathcal{U}, K, \gamma(\cdot))$

$\mathcal{U} = \mathcal{U} - \mathcal{X}$

Annotate $\mathcal{X}$ to get $\langle \mathcal{X}, \mathcal{Y} \rangle$

$\mathcal{L}^* = SeqMix(\langle \mathcal{X}, \mathcal{Y} \rangle, \alpha, \zeta(\cdot), d(\cdot))$

$\mathcal{L} = \mathcal{L} \cup \langle \mathcal{X}, \mathcal{Y} \rangle \cup \mathcal{L}^*$

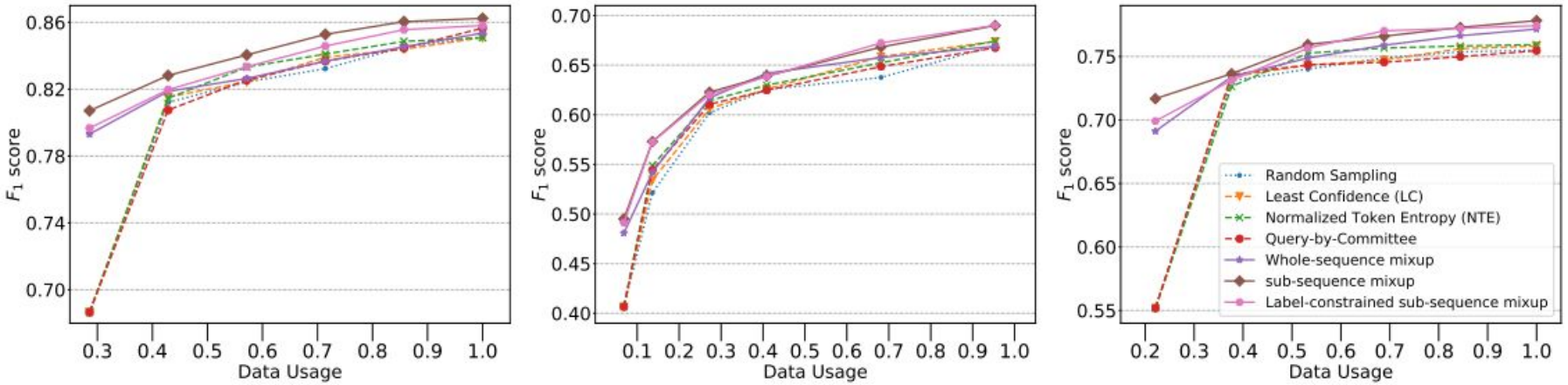
$\theta = \text{train } (\theta, \mathcal{L})$

end

Output: The sequence model trained with active data augmentation: θ

Experiments

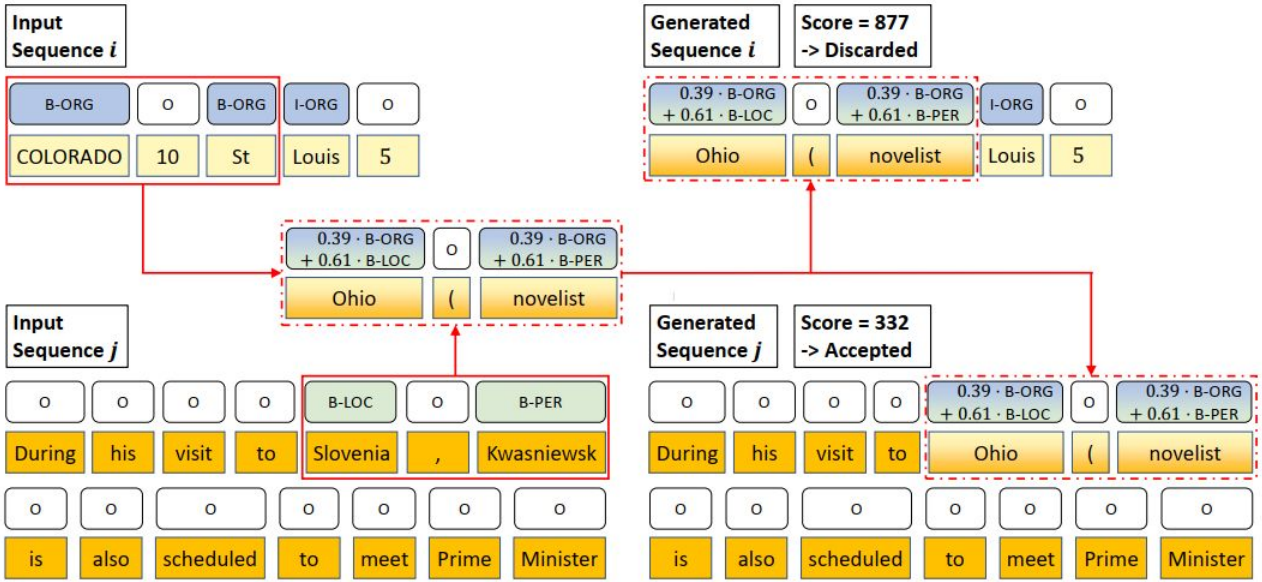
It conducts experiments on CoNLL-03(NER), ACE05(event detection) and Webpage(NER), each dataset is initialized with a small number of labeled samples, data usage is ratio of data used in training. The proposed 3 strategies are slightly better than baselines.



(a) CoNLL-03 (700 labeled data) (b) ACE05 (14k labeled data) (c) WebPage (385 labeled data)

Case study

It also provides a case study in which the subsequence (Slovenia, Kwasniewsk) and (Colorado 10 St) is mixed and obtain (Ohio (novelist).



Conslusion

This paper uses a more complicated mixup method in which the resulted embedding is existed in the vocabulary and it also considers mixup in different levels. The scoring function that removing low-confident sequence usually appears in filtering method but first time in mixup method.

Thank you!