对比学习调研-持续更新

思想

通过自动构造正负样本，要求模型得到一个表示学习模型，通过这个模型，使得相似的实例在投影空间中比较接近，而不相似的实例在投影空间中距离比较远。

归纳三部走：

- 通过一些方式构造人工正样本对；
- 在一个Batch内构造负样本对；
- 设计一个loss，拉近正样本对embedding间的距离，扩大负样本对embedding间的距离，距离可使用余弦距离。

对齐性和均匀性

好的对比学习模型具备两个属性：对齐性、均匀性。

对齐性

相似的例子，映射到单位超球面后，应该有接近的特征，也就是说，在超平面上距离比较近。

均匀性

模型应该倾向在特征里保留尽可能多的信息，等价于映射到单位超球面的特征应尽可能均匀分布在球面上，分布得越均匀，意味着保留的信息越充分，可直接用线性分类器划分。

SimCSE

抛出一个数据增强问题

样本正例增强：传统方法有 上下采样、EDA、回译、同义词替换等方法，不仅费时构建，还很可能改变原句意思，降低模型效果。

负样本构建：人工构建负样本难，需要大量实验成本。

总之，不管正样本还是负样本，基于人工或者传统方法都不好构建。

SimCSE如何解决数据问题

很简单，一个batch内，一个样本经过两次encoder，得到该样本的正例，负样本则是同一batch内其它的样本。

如何训练

1. 基于bert系列预训练模型训练；
2. 一个batch经复制后以2*batch的数据量喂给模型；
3. 构建一个损失函数infoNCE loss，思想是去最大化正例对的相似度，最小化负例对的相似度；
4. 损失计算：y_pre为句子和batch内其它句子计算的cos相似度值，y_true为被拿来计算句子的索引；

本质是让模型在训练过程中拉近正样本间的距离，推远负样本间的距离。

模型结果

1、语义文本相似度(Semantic textual similarity)

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Unsupervised models								
GloVe embeddings (avg.)♣	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
$\text{BERT}_{\text{base}}$ (first-last avg.)	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
$\text{BERT}_{\text{base}}$-flow	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
$\text{BERT}_{\text{base}}$-whitening	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
IS-$\text{BERT}_{\text{base}}$♡	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
∗ SimCSE-$\text{BERT}_{\text{base}}$	**66.68**	**81.43**	**71.38**	**78.43**	**78.47**	**75.49**	**69.92**	**74.54**
$\text{RoBERTa}_{\text{base}}$ (first-last avg.)	40.88	58.74	49.07	65.63	61.48	58.55	61.63	56.57
$\text{RoBERTa}_{\text{base}}$-whitening	46.99	63.24	57.23	71.36	68.99	61.36	62.91	61.73
∗ SimCSE-$\text{RoBERTa}_{\text{base}}$	**68.68**	**82.62**	**73.56**	**81.49**	**80.82**	**80.48**	**67.87**	**76.50**
∗ SimCSE-$\text{RoBERTa}_{\text{large}}$	**69.87**	**82.97**	**74.25**	**83.01**	**79.52**	**81.23**	**71.47**	**77.47**
Supervised models								
InferSent-GloVe♣	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder♣	64.49	67.80	64.61	76.83	73.18	74.92	76.69	71.22
$\text{SBERT}_{\text{base}}$♣	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
$\text{SBERT}_{\text{base}}$-flow	69.78	77.27	74.35	82.01	77.46	79.12	76.21	76.60
$\text{SBERT}_{\text{base}}$-whitening	69.65	77.57	74.66	82.27	78.39	79.52	76.91	77.00
∗ SimCSE-$\text{BERT}_{\text{base}}$	**75.30**	**84.67**	**80.19**	**85.40**	**80.82**	**84.25**	**80.39**	**81.57**
$\text{SRoBERTa}_{\text{base}}$♣	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
$\text{SRoBERTa}_{\text{base}}$-whitening	70.46	77.07	74.46	81.64	76.43	79.49	76.65	76.60
∗ SimCSE-$\text{RoBERTa}_{\text{base}}$	**76.53**	**85.21**	**80.95**	**86.03**	**82.57**	**85.83**	**80.50**	**82.52**
∗ SimCSE-$\text{RoBERTa}_{\text{large}}$	**77.46**	**87.27**	**82.36**	**86.66**	**83.93**	**86.70**	**81.95**	**83.70**

可以看出在STS任务上，SimCES表现非常好，在所有任务上取得SORT，而却仅仅是无监督方法就超越了有监督。

2、 下游任务(fine turing)

Model	MR	CR	SUBJ	MPQA	SST	TREC	MRPC	Avg.
Unsupervised models								
GloVe embeddings (avg.)♣	77.25	78.30	91.17	87.85	80.18	83.00	72.87	81.52
Skip-thought♡	76.50	80.10	93.60	87.10	82.00	92.20	73.00	83.50
Avg. BERT embeddings♣	78.66	86.25	94.37	88.66	84.40	**92.80**	69.54	84.94
BERT-`[CLS]` embedding♣	78.68	84.85	94.21	88.23	84.13	91.40	71.13	84.66
IS-$\text{BERT}_{\text{base}}$♡	**81.09**	**87.18**	**94.96**	**88.75**	**85.96**	88.64	74.24	**85.83**
∗ SimCSE-$\text{BERT}_{\text{base}}$	80.41	85.30	94.46	88.43	85.39	87.60	71.13	84.67
w/ MLM	80.74	85.67	94.68	87.21	84.95	89.40	**74.38**	85.29
∗ SimCSE-$\text{RoBERTa}_{\text{base}}$	79.67	84.61	91.68	85.96	84.73	84.20	64.93	82.25
w/ MLM	**82.02**	**87.52**	**94.13**	**86.24**	**88.58**	**90.20**	**74.55**	**86.18**
∗ SimCSE-$\text{RoBERTa}_{\text{large}}$	80.83	85.30	91.68	86.10	85.06	89.20	**75.65**	84.83
w/ MLM	**83.30**	**87.50**	**95.27**	**86.82**	**87.86**	**94.00**	75.36	**87.16**
Supervised models								
InferSent-GloVe♣	81.57	86.54	92.50	90.38	84.18	88.20	75.77	85.59
Universal Sentence Encoder♣	80.09	85.19	93.98	86.70	86.38	93.20	70.14	85.10
$\text{SBERT}_{\text{base}}$♣	**83.64**	**89.43**	94.39	**89.86**	**88.96**	**89.60**	76.00	**87.41**
∗ SimCSE-$\text{BERT}_{\text{base}}$	82.69	89.25	**94.81**	89.59	87.31	88.40	73.51	86.51
w/ MLM	82.68	88.88	94.52	89.82	88.41	87.60	**76.12**	86.86
$\text{SRoBERTa}_{\text{base}}$	84.91	90.83	92.56	88.75	90.50	88.60	**78.14**	87.76
∗ SimCSE-$\text{RoBERTa}_{\text{base}}$	84.92	**92.00**	**94.11**	**89.82**	91.27	88.80	75.65	88.08
w/ MLM	**85.08**	91.76	94.02	89.72	**92.31**	**91.20**	76.52	**88.66**
∗ SimCSE-$\text{RoBERTa}_{\text{large}}$	88.12	92.37	95.11	90.49	92.75	91.80	76.64	89.61
w/ MLM	**88.45**	**92.53**	**95.19**	**90.58**	**93.30**	**93.80**	**77.74**	**90.23**

下游任务并没有做到最好，句子级别的任务可能并不会有益于下游任务训练。

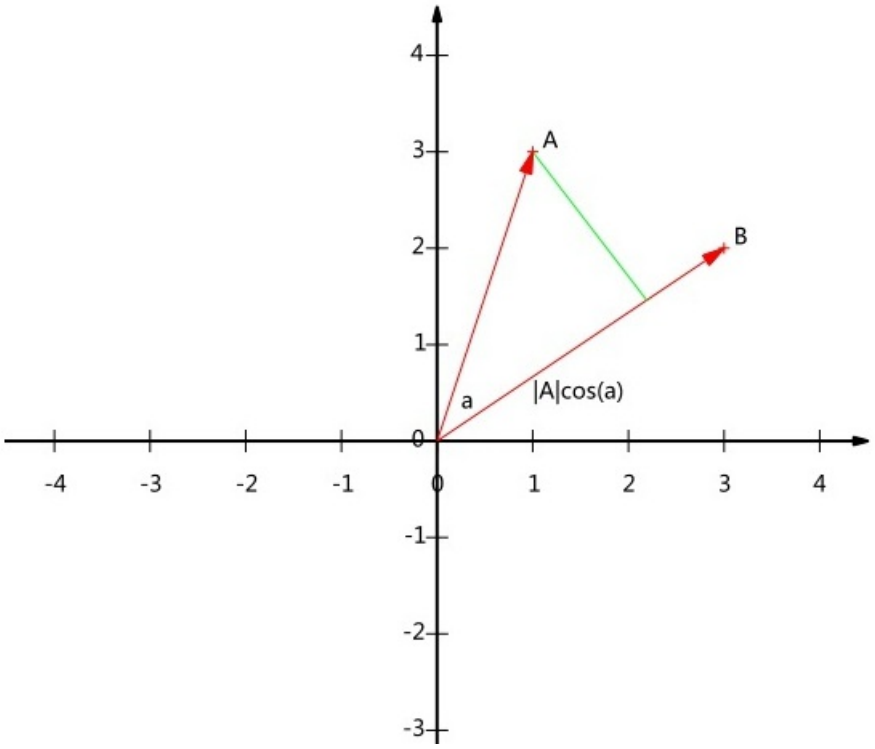
下游任务往往是一个有监督的任务，有监督就代表蕴含人的主观性，如我很高兴 和 我不高兴，去除人的主观因素，这两句话的相似度很高的，但是加了人的主观定义在里面 这两句话意思是截然相反的，而simCSE只是在无监督数据上通过自学习客观提供一个好的句向量表示模型，并不会益于下游任务。

所以SimCSE适合作为辅助模型，而核心业务如意图识别等还是要有标注的有监督模型来训练。

BERT-WHITENING

一个线性变换，就可以得到更好的bert句向量表示。

抛出一个cos问题



我们知道，cos可以用来计算两个文本向量的相似度，但前提是公式只在 标准正交基 下成立，如果基底不同，那么计算 cos相似度的公式就不一样。

所以BERT的CLS向量为什么在文本语义计算上表现很差，很可能是此时的句向量处于一个 非标准正交基(斜着的坐标)，自然不能用 标准正交基 下的cos来计算相似度。

如何解决这个问题

原作者用了大量公式来推理，但是原理很简单。

我们知道标准正态分布的均值为0，协方差矩阵为单位阵 (标准正交基)，那么我们可以学习一个参数W，将BERT句向量分布变换成均值为0，协方差矩阵为单位阵就可以了 ，这就是BERT-WHITENING模型的思路。

还能增效又提速

作者发现变换后的句子向量矩阵，经过PCA(只保留投影较大的维度)降维，作者实验时将bert-base 768 降至 256维，实验模型结果有轻微提高， 速度大大增加。