丹琦女神新作：对比学习，简单到只需要Dropout两下

原创 花小花Posy 夕小瑶的卖萌屋 2021-04-26 12:46

文 | 花小花Posy

上周把 **《对比学习有多火？文本聚类都被刷爆了...》**分享到卖萌屋的群里后，遭到了群友们一波~~嫌弃~~安利。

小伙伴们表示，插入替换的数据增强方式已经Out了，SimCSE才是现在的靓仔。

> snowfloating说：看完**Danqi Chen**组里的**SimCSE**，再看这篇感觉就没什么惊喜了。
> 苏神：直接用dropout，居然work了。真见鬼了.....
> 奥多多奥多多：这篇有说法的。

抱着一颗好奇的心，想看看这篇SimCSE到底有什么说法，又哪里见鬼了？小花认认真真拜读了原文，今天跟大家分享分享SimCSE用的什么神奇招数。

看完你可能不信，但它真的很神奇！

SimCSE的全称是 *Simple Contrastive Learning of Sentence Embeddings*，**S代表Simple**。文中的方法完全对得起题目，它是真的**简单**！简单在哪儿呢？

1. 它简单地**用dropout替换了传统的数据增强方法**，将同一个输入dropout两次作为对比学习的正例，而且效果甚好。
2. 它简单地将NLI的数据用于监督对比学习，效果也甚好。

这么简单的方法，真的work！？WHY？

下面我们一起领略一下这篇文章的风骚吧！

论文题目：

SimCSE: Simple Contrastive Learning of Sentence Embeddings

论文链接：

https://arxiv.org/pdf/2104.08821.pdf

SimCSE开篇讨论的问题是：对比学习为何work？写上一篇文的时候，我就在想对比学习为什么work呢？今天看到本文给出了很好的解释。

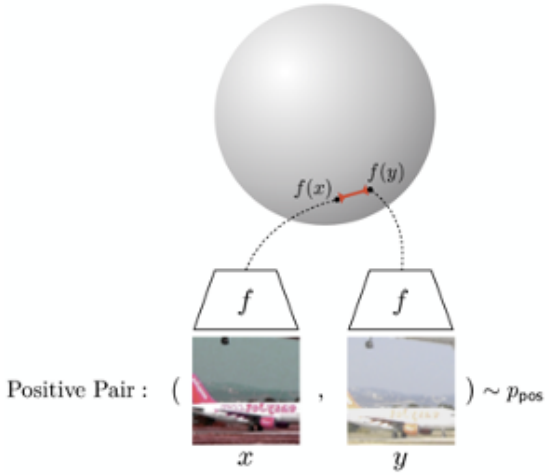
对比学习为何work？

原来 ICML2020 专门有一篇文章[1]研究了对比学习为什么work。[1]中指出，对比表示学习有用，主要是因为它优化了两个目标：

1. 正例之间表示保持较近距离
2. 随机样例的表示应分散在超球面上。

并提出这两个目标分别可以用指标**alignment**和**uniformity**来衡量。

下图可以直观理解这两个目标：



Alignment: Similar samples have similar features.

Uniformity: Preserve maximal information.

alignment计算正例对之间的向量距离的期望：$\ell_{\text{align}} \triangleq \underset{(x,x^+)\sim p_{\text{pos}}}{\mathbb{E}} \|f(x) - f(x^+)\|^2$

越相似的样例之间的alignment程度越高。因为alignment使用距离来衡量，所以距离越小，表示alignment的程度越高。

uniformity评估所有数据的向量均匀分布的程度，越均匀，保留的信息越多。

$\ell_{\text{uniform}} \triangleq \log \underset{x,y \overset{i.i.d.}{\sim}}{\mathbb{E}} e^{-2\|f(x)-f(y)\|^2}$

可以想象任意从表示空间中采样两个数据x和y, 希望他们的距离比较远。他们的距离越远，证明空间分布越uniform。所以uniformity的值也是越低越好。

SimCSE也采用这两个指标来衡量生成的句子向量，并证明了文本的语义空间也满足：alignment值越低且uniformity值越低，向量表示的质量越高，在STS任务上的Spearman相关系数越高。

SimCSE

SimCSE有两个变体：**Unsupervised SimCSE**和**Supervised SimCSE**，主要不同在于对比学习的正负例的构造。下面详细介绍下他们的构造方式。

无监督SimCSE

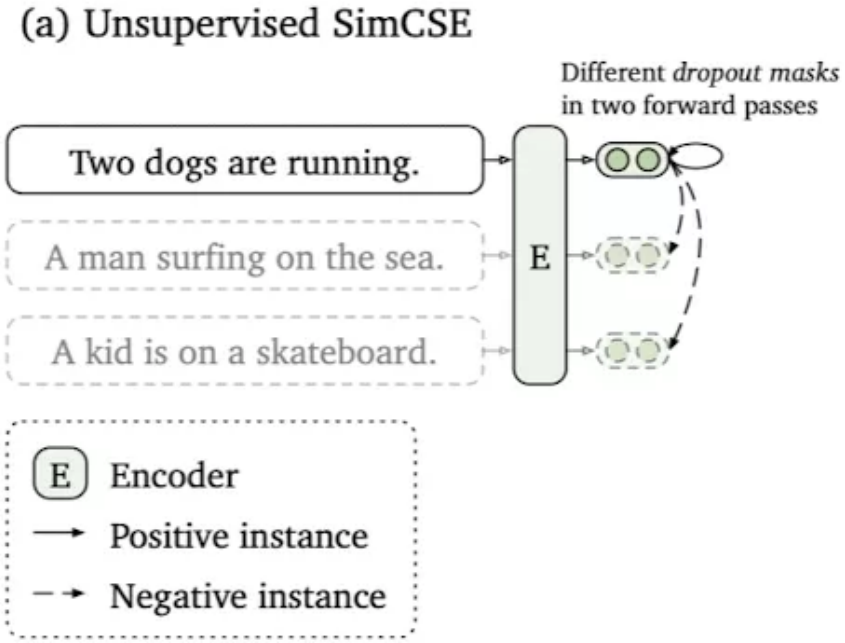
Unsupervised SimCSE 引入dropout给输入加噪声，假设加噪后的输入仍与原始输入在语义空间距离相近。其正负例的构造方式如下：

> 正例：给定输入x_i，用预训练语言模型f_θ编码x_i两次得到的两个向量$\mathbf{h}_i^{z_i}$和$\mathbf{h}_i^{z_i'}$作为正例对。

> 负例：使用in-batch negatives的方式，即随机采样一个batch中另一个输入x_j作为x_i的负例。

训练目标函数：$\ell_i = -\log \frac{e^{\text{sim}\left(\mathbf{h}_i^{z_i}, \mathbf{h}_i^{z_i'}\right)/\tau}}{\sum_{j=1}^{N} e^{\text{sim}\left(\mathbf{h}_i^{z_i}, \mathbf{h}_j^{z_j'}\right)/\tau}}$

下图展示了Unsupervised SimCSE的样例：



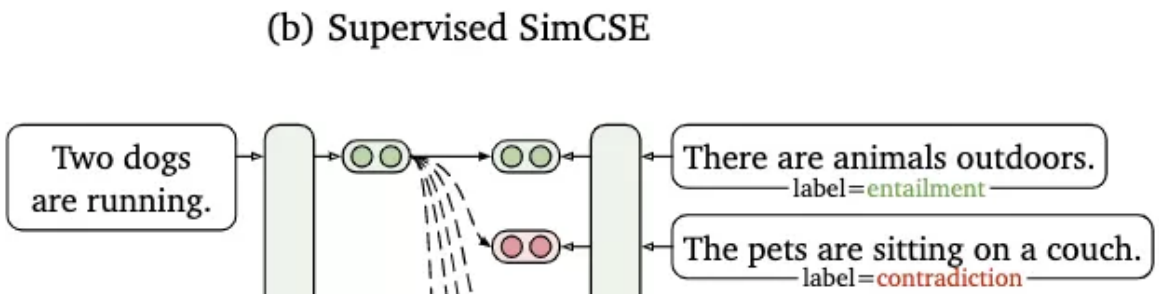
如何生成dropout mask?

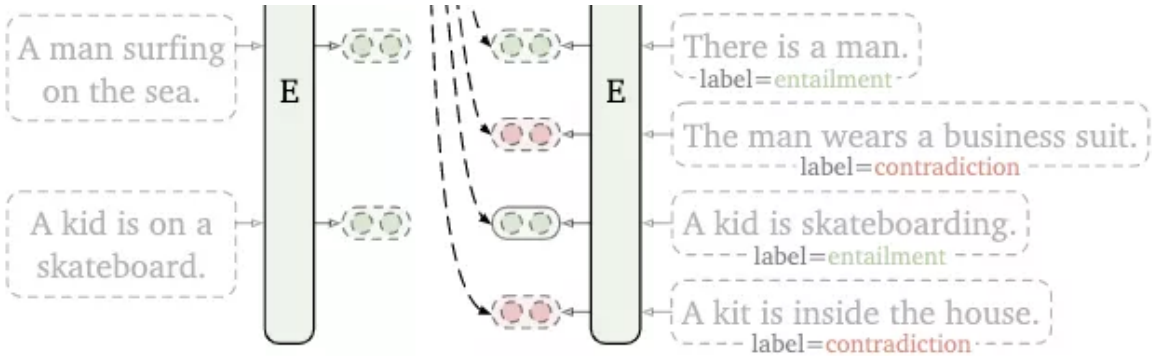
对于**Unsupervised SimCSE**，核心在于如何生成dropout mask。刚开始读完一遍的时候，惊叹原来dropout可以这么用，效果提升还挺大。后来细想，仍旧有些困惑两次dropout mask的生成过程是怎样的呢？仔细读了下，原文说：

> In other words, we pass the same input sentence to the pre-trained encoder twice and obtain two embeddings as “positive pairs”, by applying independently sampled dropout masks。

还是不太清楚。后来看了作者在GitHub的issue里面的回复才懂了。因为BERT内部每次dropout都随机会生成一个不同的dropout mask。所以SimCSL不需要改变原始BERT，只需要将同一个句子喂给模型两次，得到的两个向量就是应用两次不同dropout mask的结果。然后将两个向量作为正例对。（真的simple）

有监督SimCSE





本文还提出**Supervised SimCSE**，利用标注数据来构造对比学习的正负例子。为探究哪种标注数据更有利于句子向量的学习，文中在多种数据集上做了实验，最后发现NLI数据最有利于学习句子表示。下面以NLI数据为例介绍Supervised SimCSE的流程。

Supervised SimCSE 引入了NLI任务来监督对比学习过程。该模型假设如果两个句子存在蕴含关系，那么它们之间的句子向量距离应该较近；如果两个句子存在矛盾关系，那么它们的距离应该较远。因此NLI中的蕴含句对和矛盾句对分别对应对比学习中的正例对和负例对。所以在Supervised SimCSE中，正负例的构造方式如下:

> 正例：NLI中entailment关系样例对。负例：a) in-batch negatives b)NLI中关系为contradiction的样例对。

训练目标：

$$\ell_i = -\log \frac{e^{\text{sim}\left(\mathbf{h}_i, \mathbf{h}_i^+\right)/\tau}}{\sum_{j=1}^{N}\left(e^{\text{sim}\left(\mathbf{h}_i, \mathbf{h}_j^+\right)/\tau} + e^{\text{sim}\left(\mathbf{h}_i, \mathbf{h}_j^-\right)/\tau}\right)}$$

实验结果

Dropout优于传统数据增强?

下图中对比了使用Unsupervised SimCSE （第一行None）和常见的数据增强方法在STS-B验证集上的Spearman's Correlation。

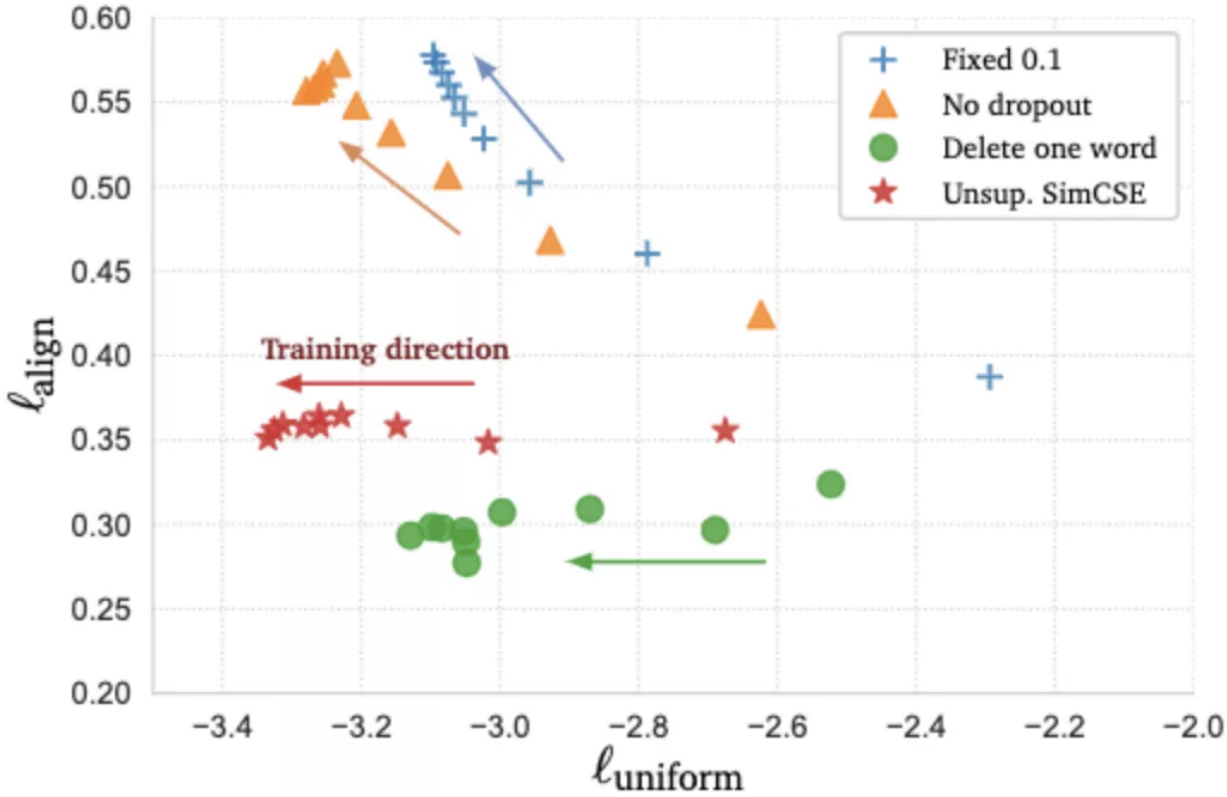
Data augmentation			STS-B
None			**79.1**
Crop	*10%*	*20%*	*30%*
	75.4	70.1	63.7
Word deletion	*10%*	*20%*	*30%*
	74.7	71.2	70.2
Delete one word			74.8
w/o dropout			71.4
MLM 15%			66.8
Crop 10% + MLM 15%			70.8

其中crop k%表示随机减掉k%长度的span，word deletion表示随机删除k%的词，delete one word只删除一个词，MLM 15%表示用$\mathrm{BERT}_{\mathrm{base}}$随机替换掉15%的词。上表中所有dropout的方法的dropout的比例都是0.1。(因为文中对比了不同比例的dropout，p=0.1效果最好。)

实验结果很明显的可以看出，SimCSE是远超其余数据增强方法的。小花的理解是传统数据增强的方法是对原始输入直接进行改变，在编码后，增强的数据与原始数据在语义空间的距离是不是要比直接用dropout的方式要远。

Dropout与对比学习的关系

为了理解dropout为什么work，作者可视化了不同方法下alignment和uniformity在训练过程中的变化趋势。



上图中对比了在不同数据增强/dropout方式下，ℓ_{uniform} 和 ℓ_{align} 在训练过程中的变化方向（每训练10步采样一次）。Fix 0.1表示p=0.1时，两次使用相同dropout mask。对于Fixed 0.1和No dropout来讲，正例对的句子表示是完全相同的，

可以看到随着训练步数增加，Unsup. SimCSE的 ℓ_{uniform} 平稳地递减。虽然 ℓ_{align} 降低的趋势却不明显，但其初始化的值就相对较低。上图进一步验证了SimCSE有效的原因是，它可以让alingnment和uniformity的值逐渐降低。

> 小花在这里有一个问题请教：使用Fixed 0.1和No dropout与另外两种方式相比较，是不是不太公平？因为当正例对两个向量完全相同时，其实是缺失了一些变体的对比信息在里面的。还有既然两个向量完全相同，ℓ_{align} 为什么会上升呢？还望理解的小伙伴留言讨论下呀！(ㅣ▽ㅣ)

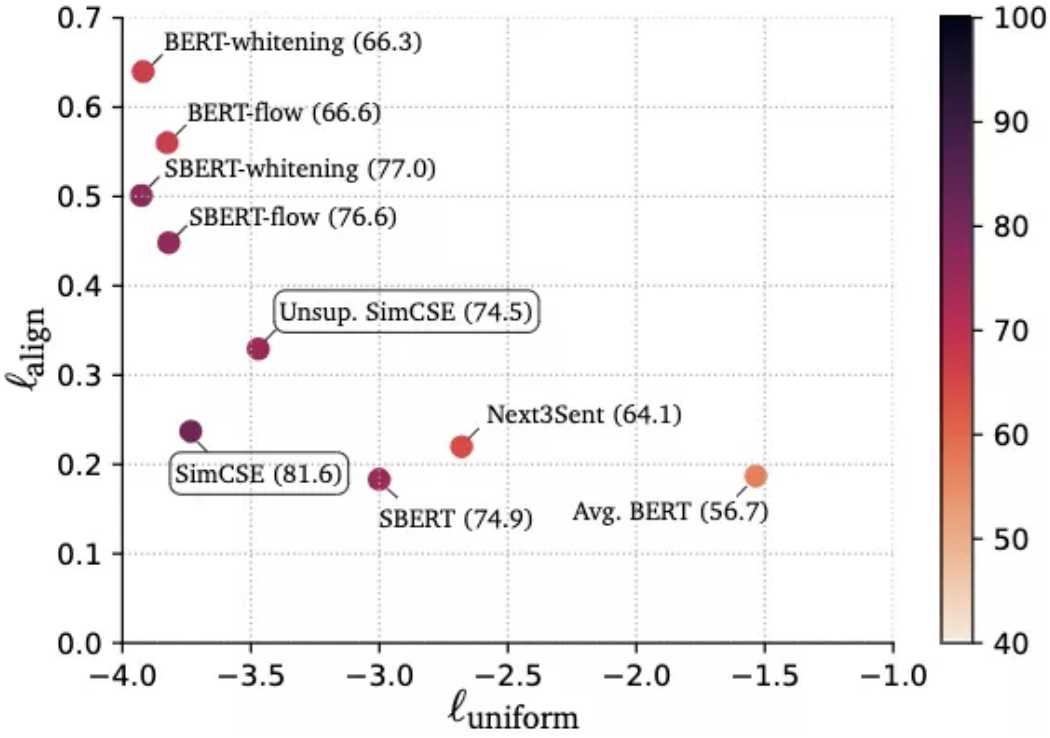
语义文本相似效果如何?

SimCSE在STS（语义文本相似）任务上进行了评估。评价指标是 Spearman's correlation。

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**	**83.76**

表格中对比了各种建模句子向量的方法，包括简单的对Glove向量取平均，到前不久的SOTA:BERT-Flow和BERT-Whitening。可以看到，在各种编码器和有无监督模式下，SimCSE都取得了显著的提升。比如无监督时，BERT_{base}和RoBERTa_{base}与BERT-Whitening相比，Avg. 分别提升了7.96%和14.77%。

此外，作者还对比了不同句子表示模型下 ℓ_{uniform} 和ℓ_{align} 与他们在STS任务上的结果：



可以看出：

- Avg.BERT模型的ℓ_{align} 较低，但ℓ_{uniform} 较高；

- 相反，对BERT表示进行后处理的BERT-flow和BERT-whitening的ℓ_{uniform}较低，但是ℓ_{align}却很高；
- Unsup.SimCSE和SimCSE的两个值都是较低的，他们的STS的结果也更好。

说明ℓ_{uniform}和ℓ_{align}需要结合使用，只有当二者的值都比较低时，模型学习到的句子向量表示才最适合STS任务。

迁移学习效果

除了STS任务上的评估外，本文还将训练好的句子向量迁移到7个任务上。

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-BERT$_{\text{base}}$♡	**81.09**	**87.18**	**94.96**	**88.75**	**85.96**	88.64	74.24	**85.83**
* SimCSE-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-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-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
SBERT$_{\text{base}}$♣	**83.64**	**89.43**	94.39	**89.86**	**88.96**	**89.60**	76.00	**87.41**
* SimCSE-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
SRoBERTa$_{\text{base}}$	84.91	90.83	92.56	88.75	90.50	88.60	**78.14**	87.76
* SimCSE-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-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**

迁移学习上的SimCSE并没有展现出明显的优势。作者的解释是句子级别的训练目标并不能直接有利于迁移学习。为了让迁移学习效果更好，文中还是尝试将MLM损失和对比学习损失一起训练，取得了少量的提升（上表中标有w/MLM的行）。

有开源嘛？

有的！4月23号刚开源的代码。

GitHub链接：

https://github.com/princeton-nlp/SimCSE

文中的预训练语言模型已经整合到了HuggingFace中，可以像BERT模型那样，直接通过API调用模型。

```
from transformers import AutoModel, AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained("princeton-nlp/sup-simcse-bert-base-uncased")
model = AutoModel.from_pretrained("princeton-nlp/sup-simcse-bert-base-uncased")
```

想动手试试的小伙伴们赶紧GitHub看看吧...

小结

本文提出了一个简单的对比学习的框架，SimCSE，用于学习句子表示。文中提出dropout+对比学习和NLI+对比学习，都非常有利于句子表示的学习。SimCSE大幅刷新STS任务榜单，取得了新一轮的SOTA。

> 这篇文章让小花很爱的一点是，明明是我们习以为常的dropuout和早就熟悉透了的的NLI数据，但是本文的作者们却能从一个全新的角度看待它们，将它们与比学习建立联系，取得非常显著的提升，并合理地解释为什么work。

参考文献

[1] Wang, T., & Isola, P. (2020). Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. ICML. https://arxiv.org/pdf/2005.10242.pdf