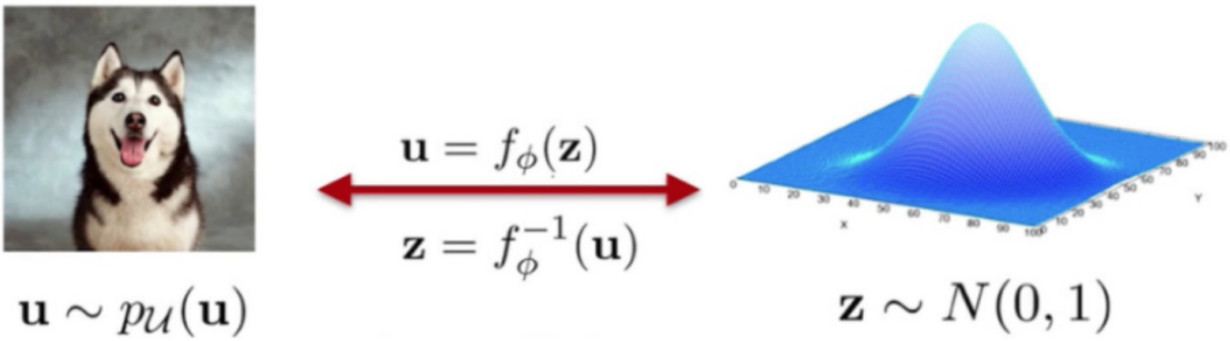
还在用[CLS]？从BERT得到最强句子Embedding的打开方式！

原创 涅生 夕小瑶的卖萌屋 2020-12-24 12:30

收录于话题

#卖萌屋@自然语言处理

69个



文：涅生

编：兔子酱

你有尝试从 BERT 提取编码后的 sentence embedding 吗？很多小伙伴的第一反应是：不就是直接取顶层的[CLS] token的embedding作为句子表示嘛，难道还有其他套路不成？

nono，你知道这样得到的句子表示捕捉到的语义信息其实很弱吗？今天向大家介绍一篇来自于 CMU 和字节跳动合作，发表在 EMNLP2020 的 paper， 详尽地分析了从预训练模型得到 sentence embedding 的常规方式的缺陷和最佳打开方式，是一篇非常实用、轻松帮助大家用BERT刷分的文章。论文质量蛮高，分析和发现很有趣，通读之后感觉收获多多。

论文题目:

On the Sentence Embeddings from Pre-trained Language Models

论文链接:

https://arxiv.org/pdf/2011.05864.pdf

Github:

https://github.com/bohanli/BERT-flow

Arxiv访问慢的小伙伴也可以在 **【夕小瑶的卖萌屋】**订阅号后台回复关键词 **【1224】** 下载论文PDF~

背景

自2018年BERT惊艳众人之后，基于预训练模型对下游任务进行微调已成为炼丹的标配。然而近两年的研究却发现，没有经过微调，直接由BERT得到的句子表示在语义文本相似性方面明显薄弱，甚至会弱于GloVe得到的表示。此篇论文中首先从理论上探索了masked language model 跟语义相似性任务上的联系，并通过实验分析了BERT的句子表示，最后提出了BERT-Flow来解决上述问题。

为什么BERT的句子Embeddings表现弱?

由于Reimers等人之前已实验证明 context embeddings 取平均要优于[CLS] token的embedding。因而在文章中，作者都以最后几层文本嵌入向量的平均值来作为BERT句子的表示向量。

语义相似性与BERT预训练的联系

为了探究上述问题，作者首先将语言模型(LM)与掩盖语言模型(MLM) 统一为: 给定context（c）预测得到 token(x) 的概率分布，即

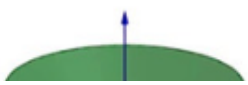
$$p(x|c) = \frac{exp(h_c^T w_x)}{\sum_{x'} exp(h_c^T w_{x'})}$$

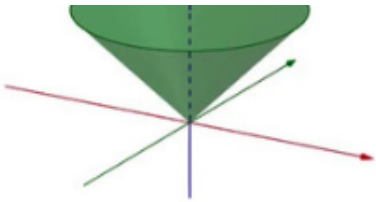
这里 h_c是context的embedding，w_x 表示 x 的word embedding。进一步，由于将 embedding 正则化到单位超球面时，两个向量的点积等价于它们的cosine 相似度，我们便可以将BERT句子表示的相似度简化为文本表示的相似度，即 $h_c^T \cdot h_c'$ 。

另外，考虑到在训练中，当 c 与 w 同时出现时，它们对应的向量表示也会更接近。换句话说，context-context 的相似度可以通过 context-words 之间的相似度推出或加强。

各向异性嵌入空间

Jun Gao, Lingxiao Wang 等人在近几年的ICLR paper中有提到语言模型中最大似然目标的训练会产生各向异性的词向量空间，即向量各个方向分布并不均匀，并且在向量空间中占据了一个狭窄的圆锥体，如下图所示~





这种情况同样也存在于预训练好的基于Transformer的模型中，比如BERT，GPT-2。而在这篇paper中，作者通过实验得到以下两个发现：

- **词频率影响词向量空间的分布**：文中通过度量BERT词向量表示与原点 l_2 距离的均值得到以下的图表。我们可以看到**高频的词更接近原点**。由于word embedding在训练过程中起到连接文本embedding的作用，我们所需的句子表示向量可能会相应地被单词频率信息误导，且其保留的语义信息可能会被破坏。

High-frequency → Low-frequency

Rank of word frequency	$(0, 100)$	$[100, 500)$	$[500, 5K)$	$[5K, 1K)$
Mean ℓ_2-norm	0.95	1.04	1.22	1.45

Close to the origin → Far away from the origin

- **低频词分布偏向稀疏**：文中度量了词向量空间中与K近邻单词的 l_2 距离的均值。我们可以看到高频词分布更集中，而低频词分布则偏向稀疏。然而稀疏性的分布会导致表示空间中存在很多“洞”，这些洞会破坏向量空间的“凸性”。考虑到BERT句子向量的产生保留了凸性，因而直接使用其句子embeddings会存在问题。

High-frequency → Low-frequency

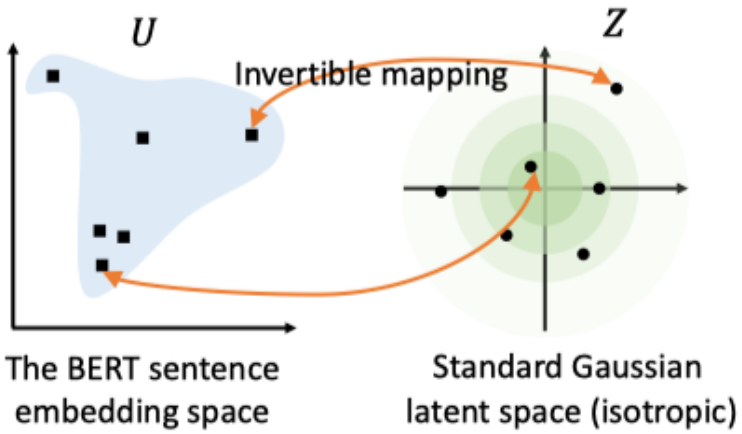
Rank of word frequency	$(0, 100)$	$[100, 500)$	$[500, 5K)$	$[5K, 1K)$
Mean k-NN ℓ_2-dist. ($k = 3$)	0.77	0.93	1.16	1.30
Mean k-NN ℓ_2-dist. ($k = 5$)	0.83	0.99	1.22	1.34
Mean k-NN ℓ_2-dist. ($k = 7$)	0.87	1.04	1.26	1.37
Mean k-NN dot-product. ($k = 3$)	0.73	0.92	1.20	1.63
Mean k-NN dot-product. ($k = 5$)	0.73	0.91	1.19	1.61
Mean k-NN dot-product. ($k = 7$)	0.72	0.90	1.17	1.60

Dense → Sparse

Flow-based 生成模型

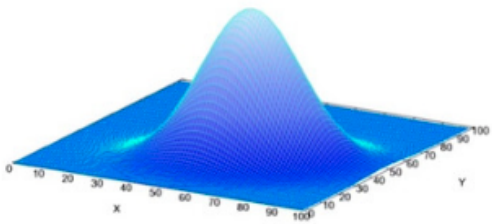
那么，如何无监督情况下充分利用BERT获得的句子语义信息呢？为了解决上述存在的问题，作者提出了一种将BERT embedding空间映射到一个标准高斯隐空间的方法（如下图所示），并称之为“BERT-flow”。而选择 Gaussian 空间的动机也是因为其自身的特点：

1. 标准高斯分布满足各向同性
2. 高斯分布区域没有“洞”，即不存在破坏“凸性”的情况



$$\mathbf{u} = f_\phi(\mathbf{z})$$

$$\mathbf{z} = f_\phi^{-1}(\mathbf{u})$$



$$\mathbf{u} \sim p_{\mathcal{U}}(\mathbf{u})$$

$$\mathbf{z} \sim N(0,1)$$

上图中$\mathcal{Z}$ 表示隐空间，$\mathcal{U}$ 表示观测到的空间，f: $\mathcal{Z} \rightarrow \mathcal{U}$ 是可逆的变换。根据概率密度函数中变量替换的定理，我们可以得到观测变量u的概率密度函数如下：

$$p_{\mathcal{U}}(u) = p_{\mathcal{Z}}(f_\phi^{-1}(u))|det\frac{\partial f_\phi^{-1}(u)}{\partial u}|$$

进一步，作者通过最大化BERT句子表示的边缘似然函数来学习基于流的生成模型，即通过如下的公式来训练flow的参数：

$$max_\phi \mathbf{E}_{u=BERT(sentence,sentence\sim D)} logp_{\mathcal{Z}}(f_\phi^{-1}(u)) + log|det\frac{\partial f_\phi^{-1}(u)}{\partial u}|$$

其中 $\mathcal{D}$ 表示数据集分布，f_ϕ 为神经网络。需要注意的是，在训练中，**不需要任何人工标注！** 另外，BERT的参数保持不变，仅有流的参数进行优化更新。其次，在实验中，作者基于Glow (Dinh et al., 2015)的设计（多个可逆变换组合）进行改动，比如将仿射耦合(affine coupling)替换为了加法耦合（additive coupling）。

实验及结果

论文的实验部分在7个数据集上进行衡量语义文本相似性任务的效果。

实验步骤：

1. 通过句子encoder得到每个句子的向量表示。
2. 计算句子之间的cosine similarity 作为模型预测的相似度。
3. 计算Spearman系数。

实验结果：

Dataset	STS-B	SICK-R	STS-12	STS-13	STS-14	STS-15	STS-16
Published in (Reimers and Gurevych, 2019)							
Avg. GloVe embeddings	58.02	53.76	55.14	70.66	59.73	68.25	63.66
Avg. BERT embeddings	46.35	58.40	38.78	57.98	57.98	63.15	61.06
BERT CLS-vector	16.50	42.63	20.16	30.01	20.09	36.88	38.03
Our Implementation							
$\text{BERT}_{\text{base}}$	47.29	58.21	49.07	55.92	54.75	62.75	65.19
$\text{BERT}_{\text{base}}$-last2avg	59.04	63.75	57.84	61.95	62.48	70.95	69.81
$\text{BERT}_{\text{base}}$-flow (NLI*)	58.56 (↓)	**65.44** (↑)	59.54 (↑)	64.69 (↑)	64.66 (↑)	72.92 (↑)	71.84 (↑)
$\text{BERT}_{\text{base}}$-flow (target)	70.72 (↑)	63.11(↓)	63.48 (↑)	72.14 (↑)	68.42 (↑)	73.77 (↑)	75.37 (↑)
$\text{BERT}_{\text{large}}$	46.99	53.74	46.89	53.32	49.27	56.54	61.63
$\text{BERT}_{\text{large}}$-last2avg	59.56	60.22	57.68	61.37	61.02	68.04	70.32
$\text{BERT}_{\text{large}}$-flow (NLI*)	68.09 (↑)	64.62 (↑)	61.72 (↑)	66.05 (↑)	66.34 (↑)	74.87 (↑)	74.47 (↑)
$\text{BERT}_{\text{large}}$-flow (target)	**72.26** (↑)	62.50 (↑)	**65.20** (↑)	**73.39** (↑)	**69.42** (↑)	**74.92** (↑)	**77.63** (↑)

上图汇报了sentence embeddings的余弦相似度同多个数据集上真实标签之间的Spearman等级相关性得分（$\rho \times 100$），其中flow-target 表示在完整的目标数据集（train+validation+test）上进行学习，flow-NLI 表示模型在NLI（natual language inference）任务的测试，绿色箭头表示相对于BERT的baseline，模型的效果有提升，红色反之。

我们可以注意到模型的改进对于效果的提升还是很显著滴！文章同样还在无监督问答任务证明模型的有效性，并将BERT-flow得到的语义相似度同词法相似度(通过编辑距离来衡量)进行对比，结果同样证明模型在引入流的可逆映射后减弱了语义相似性与词法相似性之间的联系！具体信息大家可查阅paper~

小结

总之，这篇paper探究了BERT句子表示对于语义相似性上潜在的问题，并提出了基于流的可逆映射来改进在对应任务上的表现。想多了解的童鞋可以看看原文，相信你们也会喜欢上这篇paper！

参考文献

[1] Nils Reimers and Iryna Gurevych. 2019. SentenceBERT: Sentence embeddings using siamese BERTnetworks. In Proceedings of EMNLP-IJCNLP.

[2] Ivan Kobyzev, Simon Prince, and Marcus A Brubaker. 2019. Normalizing flows: Introduction and ideas. arXiv preprint arXiv:1908.09257.

[3] Durk P Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. In Proceedings of NeurIPS.

[4] Li, Bohan, et al. "On the Sentence Embeddings from Pre-trained Language Models." arXiv preprint arXiv:2011.05864 (2020).

[5]xxxxxtesttestksdjfkahsdjkfhakjsdfhajksdfhjkahdfjkahdasjdfh