136 | ACL 2018论文精读：什么是“端到端”的语义哈希？

2018-08-15 洪亮劼

AI技术内参 进入课程 >

讲述：初明明
时长 07:27 大小 3.41M

今天，我们来看今年 ACL 大会的一篇最佳论文提名，题目是《NASH：面向生成语义哈希的端到端神经架构》（NASH: Toward End-to-End Neural Architecture for Generative Semantic Hashing）。

先来简单介绍下论文的作者群，我着重介绍三位。

第一作者沈丁涵（Dinghan Shen 音译）是杜克大学计算机科学系的博士生。他已经发表了多篇自然语言处理和机器学习相关的论文，并且在 NEC 实验室和微软研究院都实习过。

论文的共同第一作者苏勤亮（Qinliang Su 音译），目前是中山大学数据科学与计算机学院的副教授。他在香港大学取得博士学位，之后曾在杜克大学从事博士后研究工作。

作者中的劳伦斯·卡林（Lawrence Carin）是杜克大学教授。卡林是机器学习的权威，也是沈丁涵的导师。

论文的主要贡献

在很多的应用中，我们都需要根据一个已有的文档表达和一个文档库，找到最相近的，或者说最类似的文档。这经常被叫作“**相似查找**”（Similarity Search）或者“**最近邻查找**”（Nearest-Neighbor Search），在推荐系统、信息检索、图片检索等领域都有非常广泛的应用。

“语义哈希”（Semantic Hashing）被认为是解决“相似查找”的一个重要并且行之有效的方法。简单来说，“语义哈希”要做的就是把文档表达为离散的，也就是二元的向量。这些向量保留了文档在原始空间中的相似关系。因此常常被认为是带有语义的哈希过程，这也就是“语义哈希”这个名字的来历。

当我们把文档转换为语义哈希空间之后，文档之间相似度的计算就变成了利用“**汉明距离**”（Hamming Distance）来计算离散向量之间的距离。在当下的计算机体系架构中，上百万文档之间的“汉明距离”都可以在几个毫秒间完成计算。因此，我们可以看到，**“语义哈希”的一个优势就是计算快捷，并且保持了原始空间的语义信息。**

那么，看似这么有优势的“语义哈希”有没有什么劣势呢？

虽然已经有相当多的研究针对文字数据产生哈希，但是这些现有的方法都有一些明显的问题，其中最紧要的一个问题就是这些方法大多都需要两个阶段。

具体是哪些方法呢？我把这些方法归纳为两种思路。

第一种思路，我们首先需要在无监督的条件下学习文档的二元哈希；然后，我们需要训练 L 个二元分类器来预测 L 个二元位的哈希值，这个步骤是监督学习过程。

第二种思路，我们首先针对文档学习连续的表达向量，然后在测试阶段再把连续值进行二元离散化。

很明显，不管是哪一种思路，这种两个步骤的方法都不可避免地会仅仅得到次优的结果。这是因为两个步骤的优化流程是脱节的。而且，在从连续的表达向量到二元离散化的过程中，往往利用的是经验法则（Heuristic），因此语义信息可能被丢失。

基于这些问题，这篇论文提出了“**端到端**”（End-to-End）的“语义哈希”训练过程。作者们认为，经过一个阶段就可以得到完整哈希值的研究工作，这篇文章是第一个。在此之上，作者们利用了最新的**NVI 框架**（Neural Variational Inference，神经化的变分推断），来学习文档的二元编码，在无监督和监督环境下都取得了不错的结果。

这篇论文的另一个贡献就是在提出的方法和“比率损失理论”（Rate Distortion Theory）之间建立了联系。在这个联系的基础上，作者们展示了如何在模型的训练过程中“注入”（Inject）“**数据相关的噪音**”（Data-Dependent Noise）来达到更好的效果。

论文的核心方法

作者们首先把从文档生成“语义哈希”看作是一种**编码（Encode）和解码（Decode）的流程。**文档的二元哈希向量则被看成了表达文档的一种隐变量（Latent Variable）。也就是说，作者们认为文档的哈希向量是从文档的特性（可以是 TF-IDF 值）产生的一组隐变量，这也被认为是一种编码的过程，是从文档的特性向量到哈希向量的编码。

在过去的模型中，编码过程是被反复关注的，但是解码过程则很少有模型去直接建模。所谓的解码过程就是从已经产生的哈希向量转换成为文档的特性向量的过程。也就是说，我们希望能够重新从哈希向量中生成原始的数据。

对原始数据和中间隐变量的编码过程统一进行建模，是当前神经网络生成式模型的一种标准方法。在这里，编码和解码都各自有不同的神经网络，用于表达相应的条件概率分布。

具体来说，数据的原始信息 X 首先经过一个多层感知网，然后再变换成为二元的中间变量 Z。这时候，Z 其实就是我们需要得到的哈希向量了。只不过在提出的模型中，还有第二个部分，那就是从 Z 得到 X 的一个重现，也就是我们刚才提到的利用哈希来重构数据。很明显，我们希望重构的 X 和原始的 X 之间要非常相似，也就是说距离最小。

作者们发现，从数据中学习一个二元编码是“信息论”（Information Theory）中典型的“有损源编码”（Lossy Source Coding）问题。因此，**“语义哈希”其实也可以被看作是一个“比率损失平衡”（Rate Distortion Tradeoff）问题。**

什么意思呢？就是说，我们希望用较少的比率来对信息进行编码，同时又希望从编码中重构的数据能够和原始的数据尽量相近。很明显，这两者有一点“鱼与熊掌不可兼得”的意思，也就是这两者需要一个平衡才能达到最优。

把重写模型的目标函数定为“比率损失平衡”，通过这种形式，作者们意识到模型中的从编码到重构数据的条件分布，也就是一个高斯分布中的**方差值**，其实控制了这个平衡的关系。那么，就需要针对不同的文档对这个方差值进行调整，从而达到最优的编码效果，同时又是比率损失平衡的。作者们并没有采用去优化这个方差值的办法，而是在一个固定的方差值周围加入一些随机噪声，从而在实际实验中收到了不错的效果。

论文的实验结果

作者们利用了三个数据集进行实验，所有的数据集都首先转换成为 TF-IDF 的形式。作者们把提出的方法和其他的五种基本方法进行了比较。

从总体上来说，文章提出的方法在没有随机噪声的情况下，已经比其他五种方法要好得多。加入随机噪声之后，模型就有了更好的表现力。同时，作者还展示了学到的二元哈希值的确能够保持语义信息，相同文本类别的文档，它们的哈希值非常类似，也就是我们之间说过的，他们之间的汉明距离很近。

小结

今天我为你讲了今年 ACL 的一篇最佳论文提名，至此，我们关于 ACL 2018 的分享就告一段落。

一起来回顾下要点：第一，这篇文章针对语义哈希产生过程的劣势，提出了“端到端”的语义哈希训练过程；第二，论文的核心方法是把文档生成语义哈希看作是一种编码和解码的流程，进一步发现“语义哈希”其实也可以被看作是一个“比率损失平衡”问题；第三，论文取得了不错的实验效果。

最后，给你留一个思考题，在现实中利用语义哈希，有没有什么障碍？比如要在推荐系统中做语义哈希，最大的挑战会是什么？

欢迎你给我留言，和我一起讨论。

上一篇　135 | ACL 2018论文精读：什么是对话中的前提触发？如何检测？

下一篇　137 | 如何做好人工智能项目的管理？

精选留言

写留言

> 由作者筛选后的优质留言将会公开显示，欢迎踊跃留言。