字节跳动团队获 ACL 最高奖：新的词表学习方案 VOLT

- 《Vocabulary Learningvia Optimal Transport for Machine Translation》
- ACL 大会由国际计算语言学协会主办，是自然语言处理与计算语言学领域最高级别的学术会议。
- 论文地址：https://arxiv.org/abs/2012.15671
- 项目地址：https://github.com/Jingjing-NLP/VOLT
- 提出一种新的词表学习方案 VOLT：“对机器翻译中一个重要问题提出了有效且新颖的解决方案，能显著减少词表的学习和搜索时间”
  - ✓ 使用经济学的“**边际收益**”概念定义了词表质量的评价指标：
    1) 信息熵可以理解为蕴含在每个字中的平均语义含量。信息熵越小，越加利于模型学习。
    2) 在基于频率的方法下，词表越小，稀疏标记（token）越少，参数也越少，那么更加有利于模型学习。
    3) 信息熵和词表大小不可以兼顾。一般来说，词表越大，所需参数越大，稀疏标记越多，但是信息熵在减小。为此，论文引入了“边际收益”的概念。
    4) “边际收益”衡量了付出单位代价所能获得的利益的数量。作者将信息熵看成是利益，词表大小看成是代价。随着词表的增加，不同大小的词表对应的信息熵收益是不同的。通过使用“边际收益”的概念，作者定义了衡量词表质量的指标 MUV，并且观测到了 MUV 指标和下游任务的相关性。
  - ✓ 以“**最优运输**”的数学方法尝试解决最优词表的生成问题：
    1) 词表搜索空间不仅庞大，而且是离散空间。论文作者巧妙地将词表学习转化成了搜索具有最大 MUV 分数词表的离散优化问题。

层次聚类 11.22

- 与 **k-means** 对比：
  - ✓ K-means 工作原理可以简要概述为：决定簇数（k）；从数据中随机选取 k 个点作为质心；将所有点分配到最近的聚类质心；计算新形成的簇的质心；重复步骤 3 和 4；这是一个迭代过程，直到新形成的簇的质心不变，或者达到最大迭代次数。
  - ✓ K-means 缺点：必须在算法开始前就决定簇数 K 的数量，但实际我们并不知道应该有多少个簇，所以一般都是根据自己的理解先设定一个值，这就可能导致我们的理解和实际情况存在一些偏差。
  - ✓ 层次聚类完全不同，它不需要我们开始的时候指定簇数，而是先完整的形成整个层次聚类后，通过决定合适的距离，自动就可以找到对应的簇数和聚类。
- **层次聚类**：
  - ✓ 凝聚层次聚类：先让所有点分别成为一个单独的簇，然后通过相似性不断组合，直到最后只有一个簇为止。
  - ✓ 分裂层次聚类：反过来。是从单个集群开始逐步分裂，直到无法分裂，即每个点都是一个簇。
  - ✓ 相似度的计算：计算这些簇的质心之间的距离。距离最小的点称为相似点，我们可以合并它们，也可以将其称为基于距离的算法。
  - ✓ 邻近矩阵：存储了每个点之间的距离。
  - ✓ 选择聚类的簇数：绘制树状图，设置一个阈值距离，绘制一条水平线

Pandas/Sklearn 特征筛选 11.20

- 连续型特征变量：
  - ✓ 计算一下各个变量之间的相关性：筛选出对于因变量相关性比较大的自变量；自变量之间的相关性强的话，也可以只保留部分自变量。
  - ✓ 递归消除法：选择一个基准模型，起初将所有的特征变量传进去，在确认模型性能的同时通过对特征变量的重要性进行排序，去掉不重要的特征变量，然后不断地重复上面的过程直到达到所需数量的要选择的特征变量。
  - ✓ 正则化：例如对于 Lasso 的正则化而言，对于不相关的特征而言，该算法会让其相关系数变为 0，因此不相关的特征变量很快就会被排除掉了，只剩下相关的特征变量
- 离散型特征变量：
  - ✓ 可以根据缺失值的比重来进行判断：要是对于一个离散型的特征变量而言，绝大部分的值都是缺失的，那这个特征变量也就没有存在的必要了，
  - ✓ 计算特征的重要性：在基于树的众多模型当中，会去计算每个特征变量的重要性，也就是 feature_importances_属性，得出各个特征变量的重要性程度之后再进行特征的筛选
  - ✓ Select_K_Best 算法：在 Sklearn 模块当中还提供了 SelectKBest 的 API，针对回归问题或者是分类问题，挑选合适的模型评估指标，然后设定 K 值也就是既定的特征变量的数量，进行特征的筛选。（分类：卡方；回归：f_regression）