NLP 有什么好玩的?

Kevin

二〇二二年十月六日

文章导航

1 NLP 为什么这么吸引我？

> 快问快答：因为我觉得NLP很符合我的价值观。

1.1 词表

高中那会儿非常迷恋昆德拉，不记得是哪本书里面提到了词表这个概念。大概就是说大家都有自己的词表，人之间或多或少有些交集，但是各花入各眼，人们对同一个词的理解会有细微不同。各自的差异形成了默会知识的多样，喜欢存在主义的人会极其认同这个对词表的解释。

> “人的交往形同鬼魅。”

词向量是这样，用不同分布的语料来训练就是会产生一些独一无二的差异，我们的人类同伴要想在同一个视角下交流是很困难的。我喜欢跟那些词打交道，像是一堆玩不厌的积木，我想让它们待在哪儿，它们就待在哪儿（几乎不反抗）。为此，我热衷于了解人类的各种语言系统，甚至拓展到音乐、图像这些自然语言以外的动态系统上。我可以包容那些让人难以理解的行为，因为这些系统说：所有的差异性都是必然的。

2 一些理论基础

2.1 自然语言表示学习

自然语言是天然的非结构化数据，将其存储在计算机当中最直接的方法是通过符号表示，利用符号系统对每个字符进行编码。面向算法模型进行建模时，则将其转换为独热编码向量，向量的每个维度表示某个字符或者单词是否出现。显然，独热编码天然具有稀疏性，而且向量的每个维度都正交。在词表较大的时候，这样的表示向量并不能很好地表达自然语言的非结构化特征，模型也难以从稀疏向量中获取有效信息。因此，对于自然语言的表示学习研究一直的自然语言处理的重要课题。图 1 梳理了自然语言表示学习研究中的重要技术节点。

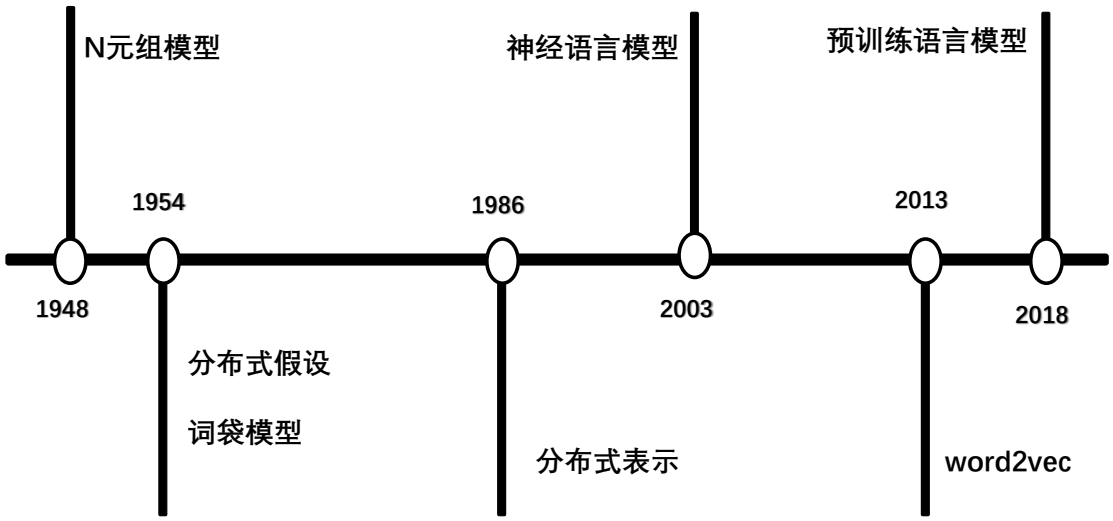

图 1: 自然语言表示学习理论发展时间线

最早的词表示模型是 N 元组模型：在预测语言序列中的下一个词的时候，通常会关注前 N 个词。语料规模非常大时，可以计算出非常良好的条件概率分布，这个概率分布能够很好地表达词的含义。随后而来的是词袋模型：将一篇文档看作其内容的单词集合，而不关注单词的顺序。N 元组模型和词袋模型揭示了自然语言处理当中的非常重要的分布式假设：具有相似分布的语义单元具有相似的含义。分布式假设奠定了后续众多语言模型的理论基础。

依照分布式假设，分布式表示向量的每一个维度都为连续的实数，这些表示向量张成的向量空间即为被表示实体的语义向量空间。随着算力增长和深度学习的不断发展，分布式表示已经成为表示学习中最常用的表示方法。

分布式表示的最典型研究是神经语言模型：首先用随机初始化的向量表示每一个词，将其作为神经网络的权重，之后依据语料计算语言序列中每个词的联合概率分布，并预测下一个词。通过这种方法训练神经网络获取词表示的方法被成为词嵌入方法，即将词嵌入到一个连续的语义向量空间当中。最后得到的向量在每个维度上并不具有良好的解释性，但是这些向量的确能够表达词的语义信息。受神经语言模型的影响，Word2Vec、

GloVe、fastText 等技术应运而生。在通用或者特定语料上训练词向量，再将其应用到下游任务中，成为了各个自然语言处理任务的范式。

2.2 预训练语言模型

随着 ELMo 和 BERT 一类预训练语言模型的提出，自然语言处理迎来了新的飞跃。与之前 Word2Vec 一类的词表示模型不同，预训练语言模型使用规模更大的语料、模型的参数和结构更加巨大繁复。而且预训练语言模型并不为每个词指定一个固定的词表达，而是依据上下文为每个词动态计算词向量。因此，这种动态的语言表示对文本语义的表达能力更强。以迁移学习的形式，在下游任务上对大规模预训练模型进行微调则成为当今自然语言处理的最常用范式。

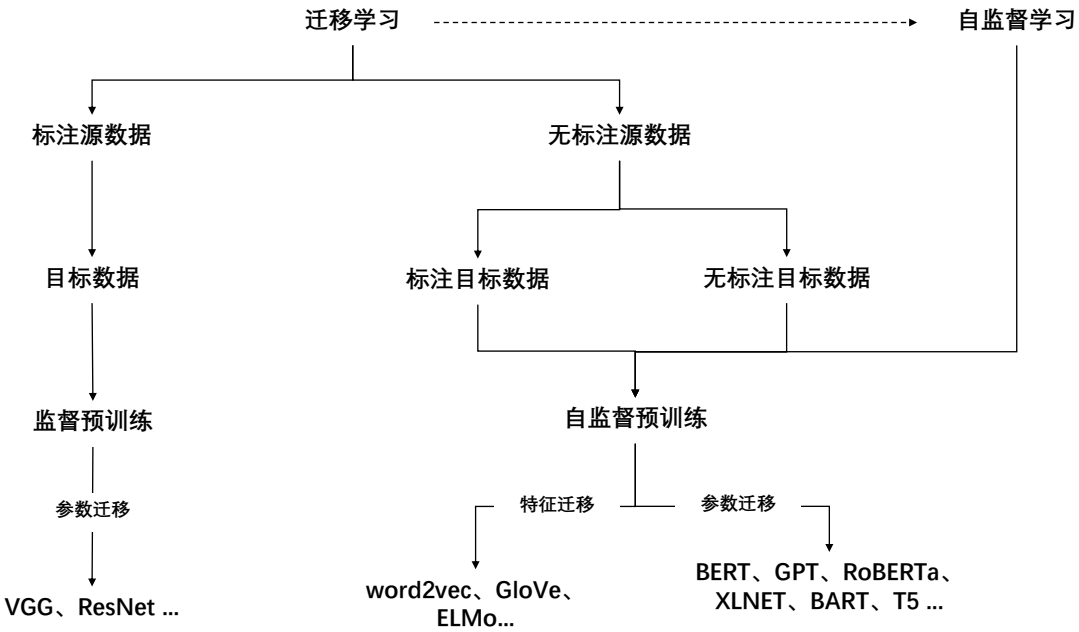

图 2: 预训练方法谱系图

预训练模型在下游任务上进行微调本质上属于迁移学习，图 2 展示了不同预训练方法的训练过程及对应示例。在自然语言处理领域中的预训练均基于自监督学习，即源数据本身并没有标签。预训练时，利用语料的上下文顺序关系构造伪数据标签，随后进行自监督预训练。

以 BERT 模型的预训练为例，BERT 的预训练任务包括遮罩预测和下一句话预测。遮罩预测任务为：在原始的语料中，使用“[MASK]”随机替换部分 token，再用模型预测这些位置原本的 token。下一句话预测任务为：将原始语料中不同的语句使用“[SEP]”分隔，通过“[CLS]”的表示向量预测这些语句是否存在先后顺序的关系。

3 一些好玩的应用

3.1 ChatBot：时代的眼泪

大三的时候上了一门安卓开发的课，那会儿对安卓一点兴趣都没有，又是早八的课，所以..... 最后课程设计是小组作业，感谢组员的呵护，让我在安卓的课上写 python。

这个项目的聊天机器人是基于 Facebook（Meta）开源的向量搜索引擎 Faiss 完成构建，结构如图 3 所示。Faiss 搜索引擎通过 KNN 的方法来确定查询向量和语料库向量的相似度。那么对于海量的待检索语料，每次查询不可能进行全表扫描。故 Faiss 对语料库进行事前聚类，聚类过程中所用的距离度量可灵活更换。随后 Faiss 维护一个聚类树，每次收到查询请求时，对树的叶子节点进行遍历，计算查询向量到每个叶子节点中心的距离。选择最近的类簇进行类内搜索，随后返回相似度最高的语料索引，最终能够做到快速近似搜索（ANN）。

为了构建可维护、可升级的检索式聊天机器人，对模块进行拆解。聊天机器人可被拆解为三个模块：检索语料库管理、词嵌入管理、Faiss 搜索引擎管理。解耦之后的项目可以通过配置文件灵活更换检索语料，以及预训练的词向量。检索语料模块负责语料库的构建及更新，提供语料路径后可以完成对句子的读取以及存储维护；词嵌入管理模块负责语料分词和词向量模型读取，分词器可灵活定义，词向量也可通过文件路径灵活配置；Faiss 搜索引擎管理模块主要负责搜索引擎的索引训练和更新，并接收用户的查询请求，返回符合条件的语料索引，相似度定义也可灵活配置。

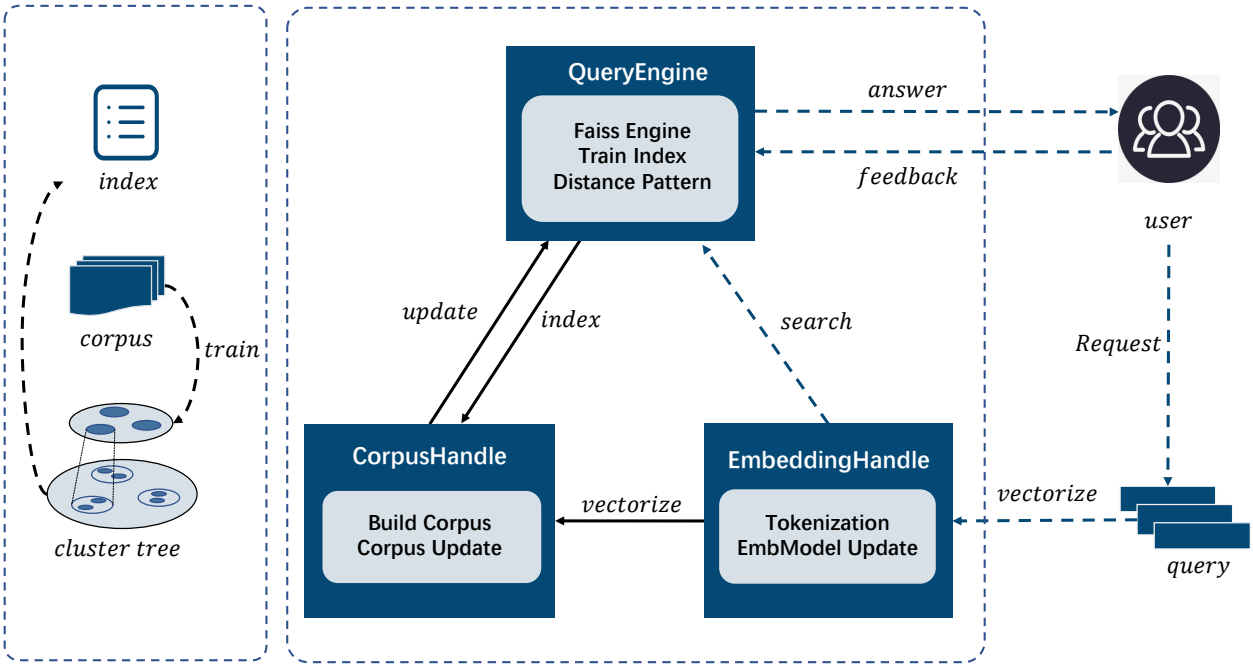

图 3: ChatBot 结构图

3.2 古琴 NLP：大一的梦想

大一的时候，和同学一起参加了中南杯（计算机设计大赛的校赛）。那会儿不懂技术，所以选择了微课制作的赛道。当时的题目选了我提的古琴，后来我们就做了“古琴中的成语”这个选题。本来想做个小程序来展示课程内容，无奈水平有限，最后是以 PPT 录播 +H5 课堂材料来呈现。最后没有入围省赛，拿了一个点击就送的校三，后来还发了 150 块钱的奖金。

这次以古琴为主题的中南杯是我本科期间参加各种比赛的开始，没想到的是，三年之后的毕业论文也是围绕古琴展开的。

> “研究整理古琴音乐遗产（包括打谱），历来都是靠‘人脑’进行，能否用‘电脑’进行辅助是值得研究的。”

我 17 岁开始弹古琴，21 岁的时候在陈长林先生的论文里读到了上面那段话，22 岁完成了“电脑”辅助打谱的初步尝试。很神奇，如果我没接触过古琴，如果我没学计算机，如果我不学 NLP，如果我没有在知网里面搜索那个关键词，3.2 节或许会是点别的东西。太巧了，我一直喜欢古琴，阴差阳错地学了计算机，从数模跑偏到 NLP，翻江倒海找到了那篇论文，在写毕业论文的时候体会到了科研开荒的艰辛。这种艰辛难以言表，奇怪的孤独感不知道跟谁诉说，孤军奋战的责任感又让人对这个选题不离不弃。

我国著名琴家，中国科学院研究员陈长林先生早在 1989 年就开始尝试将计算机技术应用到古琴研究当中，主要完成了琴谱的编码设计，以及计算机的输出现实工作。其实最早了解他，还是知道他把《春江花月夜》的琵琶曲移植到古琴上（百听不厌）。

通常情况下，减字由四部分组成：左手指法、徽位、右手指法、弦位，示例如图 4 所示。

图 4: 古琴减字示例

打谱是指将古琴的减字谱转录为通用的简谱或者五线谱，打谱人员通过确定每个音符的音高、时值和各个小节的节奏。从相关工作的调查来看，计算机辅助打谱并没有确定的工作范式。相关工作中，已经有研究人员设计了通过减字提取音高的算法，那么当前工作的重点则落在时值和节奏问题上。古琴节奏自由多变，一首琴曲中可能会出现多种节奏类型的小节。如果把每个音的时值确定下来，通过一定的推断，就可以把小节划分出来，节奏问题就迎刃而解了。关于时值的确定，可以通过判断每个音的音符类型来确定，如：“全音符”“四分音符”“八分音符”等。

古琴的减字谱通过减字组合的方式描述演奏方法。通常来说，一个或多个减字会对应一个或多个左手和右手的动作，所以并不是每个减字都对应一个音符，那么古琴打谱的过程可以就可以描述为序列标注或者序列生成。从辅助打谱的角度来说：序列生成的方法不便于减字和音符的对齐，打谱者难以对成谱进行识别和排版；如果使用序列标注方法，只需要引入一个新的类别标签即可解决多个减字对应一个音符的问题。综上所述，可以将计算机辅助打谱任务定义为序列标注任务：确定每个减字所对应的音符类型。

古琴有 7 根弦，13 个徽位，几十余种左右手指法，自由排列构造出的减字非常丰富。如果直接将减字当作 token，词表的复杂度会很大，语言模型难以从有限的语料中学习各个减字之间的关系。

所以使用一种解构减字的方法对减字进行编码，只需要将左右手指法、徽位和弦位确定，即可表达出一个动作。对于古琴减字谱中非指法类的减字，则直接填充四个部分。如泛音开始的提示“泛起”，左右手指法、徽位和弦位均填充为“泛起”。

数据录入时，没有明确指定每个减字对应的四个部分。因此，在进行编码时首先构建指法字典，再利用 jieba 分词器对减字进行分词，从而得到解构的减字编码。例如“大指七徽九分挑四弦”会被切分为“大指/七徽九分/挑/四弦”，得到的四个部分则分别对应了编码当中的左手指法、徽位、右手指法和弦位。

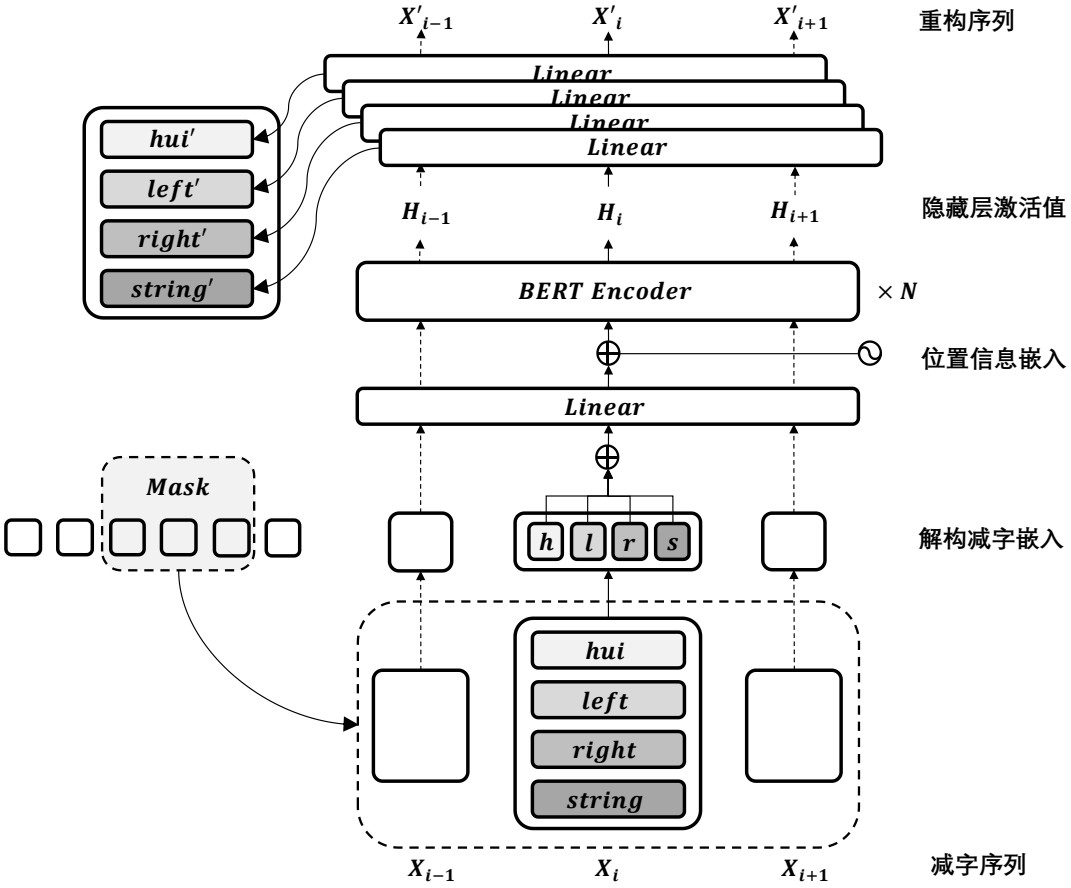

图 5: QinBERT 语言模型

如图 5 所示，QinBERT 模型的主干结构采用了 BERT 模型的编码器，Embedding 部分将左右手指法、徽位和弦位 4 个输入分别进行嵌入表达，随后将四个嵌入表示向量加在一起并输入一个全连接层。完成 token 嵌入表达之后，为了引入序列位置信息，在 token 嵌入向量当中加入相对位置嵌入向量，随后一起输入到若干个 BERT 编码器中进行特征提取。特征提取之后，将 BERT 隐藏层的输出向量分别输入到 4 个全连接层当中，对掩码部分进行预测。模型机理梳理完之后，就是训练语言模型，最后在打谱任务（序列标注）上微调，整篇论文的主要想法就是这些。