Transformer 之 NLP 概述

写作背景

研究 Transformer 其实有段时间了，但总感觉应用方面没有像 LSTM、CNN 等模型得心应手。所以，特编该文以便更有心得体会。本文分三个部分，第一部分概述下 NLP 相关概念，第二部分详细说明下 Transformer 模型，第三部分是 Transformer 实践。该文属于第一部分。

什么是 NLP

简单来说，就是通过向计算机输入语言文字，利用预先设计的计算模型，输出所预期的结果。其实，这里也没有太深奥的体系，计算机绝大数领域问题，就是数据加算法。那么，我们首先需要解决的就是怎样让计算机理解语言。

人工智能是人类一直以来追求的梦想。自然语言处理（NLP）作为人工智能的一个分支，主要解决计算机和人类语言的交互问题，其任务是能够实现计算机与人类的自然交互。人类的自然语言，无论汉语、英语还是其他语种，都承载着说话者的意图、情感，能够通过人类语言准确理解说话人所要传达的信息，并能通过人类语言与人进行准确自然的沟通，就是是自然语言处理的最终目标。

大家都知道，计算机处理的二进制数值，那么，怎么把语言转为数字呢？目前，常见的处理方式 TF-IDF、OneHot、Word2Vec。TF-IDF 网上有很多资料，大家可以搜索找下。这里简单简述下 N-Gram、OneHot、Word2Vec。

N-Gram

从模型分类上说，N-Gram 属于统计概率模型，这是与后面两个 OneHot、Distributed Representation 有所区别的。简单来说，N-Gram

就是计算文字之间的概率，但该模型在实际应用中的场景会有所限制，不能完全覆盖 NLP 的应用领域。

这里也许需要先说说朴素贝叶斯，假设有五个元素 w1、w2、w3、w4、w5，那么计算 P（w1，w2，w3，w4，w5）的概率，在朴素贝叶斯理论假设中，是认为所有元素就是独立的，那么 P（w1，w2，w3，w4，w5）等于 P(w1)* P(w2)* P(w3)* P(w4)* P(w5)。但是，这么假设其实是破坏了语言的上下文关系，毕竟语言是具有连续性的。那么，怎么体现连续性呢，如果取值为 0，就是朴素贝叶斯，取值为 1 概率模型就变为了 P(w1 ,w2 ,w3 ,w4 ,w5)等于 P(w1)* P(w2|w1)* P(w3|w2)* P(w4|w3)* P(w5|w4)，如果取值为 2 呢，概率模型就变为了 P（w1，w2，w3，w4，w5）等于 P(w1)* P(w2|w1)* P(w3|w1w2)* P(w4|w2w3)* P(w5|w3w4)，取值为 3 呢，概率模型就变为了 P（w1，w2，w3，w4，w5）等于 P(w1)* P(w2|w1)* P(w3|w1w2)* P(w4|w1w2w3)* P(w5|w2w3w4)。实际应用中，常用的取值就是 2 或者 3，经过相关实验，取值超过 3 并未有提高模型的精确率，反而会带来计算量的大幅度增加。

N-Gram 模型中的概率怎么进行训练呢，其实，相对很简单，从语料库中，获取每 N 个相邻文字的概率。举例来说，当 N 取值为 1 时候，整个语料库的，两个字的数量是 1000，其中，含有“中”的两个字的数量是 200，“中国”出现了 50 次，那么 P（中国）等于 P（中）*P（国|中），也就是（200 除以 1000）乘以（50 除以 200）。

举个实际应用的例子吧，比如，现在大家常用的输入法，都具备了联想功能，如果输入的是“我是中国”，下一字应该提示什么呢，大家一定会通过常识，认为肯定是“人”。那计算机怎么知道是“人”这个字呢，如果用 N-gram 模型的话，P（我是中国人）成句的概率数值大于所有的其他文字，如 P（我是中国菜）。假设在 N 等于 1 情况下，也就是 P（我）*P（是|我）*P（中|是）*P（国|中）*P（人|国）大于 P（我）*P（是|我）*P（中|是）*P（国|中）*P（菜|国）。

OneHot

OneHot 就是用向量表示文字，每一个字占用向量一个位置，并

设置为 1，其余位置设置为 0。举个例子来说，假设整个语料库文字为：我是中国人，一共 5 个字，那么向量的长度为 5，“我”用 OneHot 表示就是 10000，“是”就是 01000，“中”就是 00100，“国”就是 00010，“人”就是 00001。大家从例子上，基本可以清楚的了解到 OneHot 的处理方式了。但是，这么处理有什么问题呢，这里可以从两个方面考虑，一个是计算机的处理效率，刚刚语料库的长度是 5，但是真实世界，语料库的长度基本在十几万的数量等级上，那么，我们就需要设置一个长度为 10 万等级的向量矩阵，要在内存放置这么大的一个向量矩阵还是需要很大的资源与处理能力。另外一个问题就是语言的表达能力，语言文字从概率角度来说是有相关性的，比如，“女人”跟“购物”相关度肯定比“男人”与“购物”的相关度要高，那么，从 OneHot 表示方法上来说，任何两个向量的乘积都是 0，是无法表示语言文字之间的相关度的，那么必然会影响语言的处理精确度。

Distributed Representation

很多科学领域特别是应用领域的发展都是为了解决现有问题所出发的，怎么解决 OneHot 的问题呢，其实可以通过对矩阵进行降维度出来，从而减少向量的长度。其实，很早之前大神 Hinton 在 1986 年提出了它最早是 Hinton 于 1986 年提出的 Distributed Representation 模型（Word2Vec 的前身），就可以克服 OneHot 的上述缺点，当时并未引起足够的重视，随着深度学习的兴起，这些理论才赋予了新的生命。本文后续将重点说明的是 Word2Vec，从数学应用上说，OneHot、Word2Vec 都属于词向量，因为都是用向量来对语言进行表示，只是表达的方式上有所区别，但现在大家一提到词向量基本就是指 Word2Vec。从获取到 OneHot、Distributed Representation 方式说来看，区别在于，OneHot 是通过对文字在向量中的位置进行编码，而词向量是通过 Distributed Representation 是通过对语料库的训练而得到的向量编码。现在，常用的 Distributed Representation 训练模型是 Word2Vec，文中除非有特别说明，否则都是对 Word2Vec 的说明。

理解 Word2Vec 模型，其实只要了解神经网络模型，基本可以理

解模型运算方式。在 Word2Vec 模型中，常分为两个模型 CBOW 与 Skip-Gram，区别在于 CBOW 模型的训练输入是某一个特征词的上下文相关的词对应的词向量，而输出就是这特定的一个词的词向量，比如一句话“我是中国人”，把“中”去掉，“我是 国人”当作输入，而“中”当作预期的输出。而 Skip-Gram 模型和 CBOW 的思路是反着来的，即输入是特定的一个词的词向量，而输出是特定词对应的上下文词向量。对于这块描述，发现怎么写都表述有限，所以最终还是参见网上的一些资料整理成稿。我们先来看看 CBOW 模型是怎么进行词向量训练的。

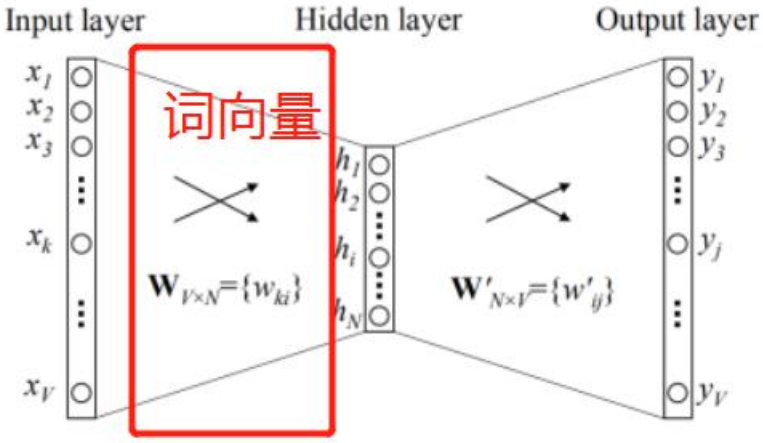

Figure 1: A simple CBOW model with only one word in the context

1. 输入层：上下文单词的 onehot，假设单词向量空间长度为 V，上下文单词个数为 C。
2. 所有 onehot 分别乘以共享的输入权重矩阵 W{V*N 矩阵*}，*N 为自己设定的数，也表示词向量的长度。*
3. *所得的向量相加求平均作为隐层向量, size 为 1*N，这里一般不采用激活函数。
4. 乘以输出权重矩阵 W' {N*V*}。
5. *得到向量 {1*V} 激活函数处理得到 V-dim 概率分布 {PS: 因为是 onehot 嘛，其中的每一维斗代表着一个单词}
6. 概率最大的 index 所指示的单词为预测出的中间词（target word）与 true label 的 onehot 做比较，误差越小越好（根据误差更新权重矩阵）

7. 定义 loss function（一般为交叉熵代价函数），采用梯度下降算法更新 W 和 W'。训练完毕后，输入层的每个单词与矩阵 W 相乘得到的向量的就是我们想要的词向量（word embedding），这个矩阵（所有单词的 word embedding）也叫做 look up table（其实聪明的你已经看出来了，其实这个 look up table 就是矩阵 W 自身）,也就是说 ,任何一个单词的 onehot 乘以这个矩阵都将得到自己的词向量。有了 look up table 就可以免去训练过程直接查表得到单词的词向量了。

NLP **研究领域**

词法分析

词法分析包括分词、词性标注、命名实体识别和词义消歧。

分词和词性标注好理解。我们知道自然语言处理中词为最小的处理单元，当你的语料为句子、短文本、篇章时，我们要做的第一步就是分词。由于英语的基本组成单位就是词，分词是比较容易的。其句子基本上就是由标点符号、空格和词构成，那么只要根据空格和标点符号将词语分割即可。中文和英文就有很大不同了。虽然基本组成单位也是词，但是中文文本是由连续的字序列构成，词与词之间是没有天然的分隔符，所以中文分词相对来说困难很多。举个例子吧，“我是中国人”，最合适的分词结果应该是“我”、“是”、“中国人”，而不是“我”、“是”、“中国”、“人”。

命名实体识别的任务是识别句子中的人名、地名和机构名称等等命名实体。每一个命名实体都是由一个或多个词语构成的。词义消歧是要根据句子上下文语境来判断出每一个或某些词语的真实意思。

文本分类

文本分类问题可以定义为给定文档 p（可能含有标题 t），将文档分类为 n 个类别中的一个或多个 ,常应用在垃圾邮件识别、情感分析、

新闻分类等。常见的文本分类方法有传统机器学习方法(贝叶斯 ,svm 等), 深度学习方法（ fastText ， TextCNN 等)。

还是举个例子吧，假设新闻分为体育、财经、政治等，如果出现一篇关于贝克汉姆的新闻信息 ,那么该文章分为体育应该是最佳的预期。

文本相关度

其实，这个问题很好定义，就是任意给定两个句子，经过计算机系统能够判断是否可以认为是一个意思 ,或者是否可以判断出相关度很高。例如，别人问你“今年多大了”，你回答“23”，那么可以认为相关度很高 ,如果你回答“我是男性”,应该判断相关度很低。那么，在应用领域中，他可以解决什么问题呢，就拿百度知道来说，他有大量的问答预料，那么通过文本相关度的分析，如果你输入的是“2019 年国庆放假安排”,应该选取的是“10 月 1 日至 10 月 7 日放假，9 月 29 日（星期日）、 10 月 12 日（星期六）上班”作为最佳。其实，这个是细分的领域，跟下面的“问答与对话”有点类似。

问答与对话

问答系统可能是 NLP 领域研究最多，应用最广的部分，比如，当你输入“中国的首都是哪个城市”，通过 NLP 系统的处理，可以产生“北京”的回答。但这个领域也是难度较高的。对应流程中的三个过程有三个研究的基本问题 ,第一是问题分析 ,即如何去分析问题;第二是信息检索，也就是如何根据问题的分析结果去缩小答案 可能存在的范围;第三是答案抽取，如何从可能存在答案的信息块中抽取答案。在问答系统的不同发展阶段，对于这三个基本问题的解决方法随着数据类型的变化在不断变化，从而形成了不同类型的问答系统。

文本摘要

随着互联网产生的文本数据越来越多，文本信息过载问题日益严重，

对各类文本进行一个“降 维”处理显得非常必要，文本摘要便是其中一个重要的手段。文本摘要旨在将文本或文本集合转换为包含关键信息的简短摘要。文本摘要按照输入类型可分为单文档摘要和多文档摘要。单文档摘要从给定的一个文档中生成摘要，多文档摘要从给定的一组主题相关的文档中生成摘要。按照输出类型可分为抽取式摘要和生成式摘要。抽取式摘要从源文档中抽取关键句和关键词组成摘要，摘要全部来源于原文。生成式摘要根据原文，允许生成新的词语、短语来组成摘要。按照有无监督数据可以分为有监督摘要和无监督摘要。本文主要关注单文档、有监督、抽取式、生成式摘要。

机器翻译

机器翻译（Machine Translation），又叫自动翻译，是指利用计算机将一种自然语言转换为另一种自然语言的过程。这个任务很好理解，比如把中文翻译为英文等。