# 第一章 语言与语言建模

第一节 语言

1、语言概述：语言，作为人类思想的一种编码方式，它按照语法规则将一组词进行排列，对人类的思想信息进行编码。人类思想信息丰富多彩，包罗万象，比如对世界的认识（常识）、疑问、内心的情感等等。但是语言并没有强大到能够编码人类思想的全部信息，有些东西，如一句老话所说：“只可意会，不可言传”。

2、语言如何进行编码：语言对人类思想的编码的套路是：语素按照构词法构成词；多个词可以组成有意义的词组；进而组成短语；然后按照语法规则构成句子，句子是表达完整意义的最小单位；多个句子组成段落；进而组成文章。

3、语言的特点：语言是一套很复杂的编码系统，它具有强大的表示能力。语言还具有多样性和歧义性：同一个意思可以用很多不同的方式来表示（语言的多样性）；同一个表达在不同语境中也可能具有不同的具体意思（语言的歧义性，歧义性使我们可以用尽量少的语句来表达尽量多的意思）。要想用语言表达出合理的意思，不仅需要考虑语法，还要考虑语义。比如：“西瓜打了人”、“梦想提取了金字塔”这样的话，虽然符合语法规则，但是不符合我们对世界的客观认识。

语言还有一个特点，就是它表达的意思非常依赖于上下文。语言虽然是一种符号系统，像数学语言那样，但是它比数学语言要复杂的多，它不可能简化成一个简单的符号系统。为了表达丰富的意思，它必须用尽量少的语句去表达尽量丰富的意思，因此需要借助上下文。

第二节 语言建模

对语言进行建模，本质上属于数据建模的问题。语言建模的根本目的，就是想让计算机自动地为我们处理语言问题。我们知道，数据建模需要对数据进行符号化或者数值化，然后才能被计算机处理，这就是语言的表征问题，我们将在第二节中介绍之。

关于语言建模，一个最朴素的想法是，我们希望模型能够理解语言（NLU），也就是说，我们输入一个句子（可能还需要输入相关的语境），模型能够理解这个句子的意思，然后基于这些理解来完成相关的各种语言任务（情感分类、摘要提取、人机对话、翻译等）。

但是“对语言的理解”如何定义？这实在是个很难严格定义的概念。对这个概念的不同解读形成不同的建模策略。具体的建模方法我们将在第三节介绍。

另外，我们都知道即便语言是我们表达思想、情感，传递信息的主要手段，它也不能**完全地**蕴含我们要表达的所有信息，比如知识。而我们之所以能用语言自由地表达这些信息，是因为我们已经将语言之外的因素隐式地编码进了大脑里。这些因素跟语言以复杂的方式混合在一起，形成我们的思想的全部信息。因此，要想让模型更充分地理解语言，可能需要不只利用语言，而是还要考虑这些隐式的因素，向模型中加入相应的先验信息，比如知识。

# 第二章 语言的表征，语义空间

第一节中介绍了语言建模的基本思想，并且提到，建模需要将语言进行数值化，这就是语言的表征问题。

语言作为数据，其表征的构造思路与其他数据有很大不同，比如图片数据，天然是连续的、数值型的数据；而语言，天然是离散的、符号型的。

对语言的表征方式可以分为离散型和连续型，另外由于语言数据可以从词的粒度上看也可以从句子的粒度上看，因此也可以分为词表征和句表征。词的离散型表征就是one-hot向量，句子的离散表征是词序列、词袋或tf-idf；词的连续型表征是词向量（word embedding），句子的连续型表征是句向量（sentence embedding）。

对于离散性表征来说，词的离散表征只是为了区分不同形态的词（从词的离散空间也能看出来：每个词在离散词空间中都是某个坐标轴上的单位向量），因此这种表征不能包含词的“词法”和“词义”信息，从离散空间上来看，词与词之间无法计算相似度。离散的句子表征虽然可以计算某种形式的“相似度”，但是也很少能反映“语法”和“语义”信息。

对于连续型表征来说，词向量通常是基于NNLM得到的，属于NNLM的“副产品”；句向量有多种获取方式，有的是类比于词向量的获取方式——通过神经网络的方式得到的；有的是基于组成该句子的词向量，以某种方式组合而成，比如：所有词向量排列成矩阵；所有词向量拼接；所有词向量求平均等。

词向量和句向量本质上都是相当于神经网络从无标签的语料中获取上下文信息，然后利用上下文信息，将词或者句子映射到连续的词法词义空间或者语法语义空间中。词法词义空间中词向量之间的相似度就代表两个词的词法或者词义的相近程度；语法语义空间中句向量之间的相似度就代表两个句子的语法或者语义的相近程度。

总结一下：连续的词向量和句向量中蕴含了上下文关系信息，从而能够反映词法词义或语法语义信息，离散词向量和句向量没有蕴含上下文关系信息，所以也无法反映词法词义信息或语法语义信息。

# 第三章 语言模型，统计语言模型

第一节 语言模型

我们在第一节中已经说过，语言建模的终极目的是NLU，即是让模型理解句子的意思。但这是个很复杂的问题，而且“理解语言”这个概念本身都很难定义。因此在实际中很难完全实现NLU模型，所以对于不同的具体的NLP任务，我们会使用不同的具体模型。而对于一般化的“语言理解”的目标，我们将实现这种目标的模型称为语言模型，并基于对“语言理解”这个概念的不同解读，使用不同的策略进行建模（虽然都未达到至纯至臻的结果，但是都是对这个终极问题的一种朴素的探索并都有一定的应用价值）。

从古至今，有两种建模思想在历史上留下其足迹：基于规则的语言模型和基于概率的语言模型。

基于规则的建模思想认为，既然语言是一种对人类思想、意图等信息的编码方式，我们对语言的理解可以通过对其编码规则的总结来实现，这些规则包括语法规则和语义规则，为了充分表达出人类所要表达的所有信息，可能还需要加上知识相关的规则。通过总结这些规则，形成模型，我们就能判断一段文本是否合理。事实证明，语言的复杂性和不充分性导致我们几乎无法总结出其完整的规则。

而基于概率的建模思想认为，我们建立语言模型根本就不需要理解语言，或者说，所谓的理解，其实就是，只要模型能用来做NLP的任务，那么就算是理解了。（从这里可以看出基于概率的建模思想的局限性——它有点类似于行为主义的思想，它本来也没有想真正地理解语言，因此可能最终也无法真正地理解语言）

基于概率的建模将NLP的任务都归结为一个任务：给定一段文本，预测后面一个单词是什么单词。这本质上是提取了词典中的词在语料库中的上下文关系信息。概率语言模型的这个预测任务可以归结为概率的计算。那么如何计算概率呢？这就是后面我们将要介绍到的统计语言模型和神经网络语言模型。

概率语言模型通常使用一个大的无标注的语料库来训练模型，一个语料库对应一个词典。上下文关系信息蕴含了一门语言的语法和语义信息，概率语言模型将这些信息蕴含到模型的参数中。

概率语言模型比规则语言模型更实用，应用更广泛，因此目前一般所说的语言模型，都是指概率语言模型。

第二节 统计语言模型

统计语言模型是概率语言模型的一种，它用统计的方法计算概率。

给定一段文本：，计算下一个词为的概率：

为了简化问题，我们提出n-gram语言模型，它实际上相当于给概率语言模型加上了n阶马尔科夫性假设，即假设出现的概率只跟其前面n个词有关，因此概率计算式变为：

其实，概率语言模型还有另一个解释，那就是以计算一句话或一段文本出现的概率为目标：

因为我们认为，一句话的概率就能反映这句话是否合乎规则地表达出一些意思，或者说反映这句话的“流畅性”。

由于语料的稀疏性，直接在语料库上统计这个概率会造成很大的误差，因此我们将这个概率分解为条件概率乘积的形式：

在n-gram语言模型下，它就变成：

句子的概率，可以看出，n-gram语言模型的一个好处是，一定程度上缓解了语料稀疏性给概率语言模型带来的麻烦。因为使用模型计算一个句子的概率时，如果这个句子是一个合理的表达，但它没有在语料库中出现过（这是很有可能的，这就是语料的稀疏性），那么直接统计这个句子的概率，就是0；但是使用n-gram语言模型的话，虽然句子没有出现过，但是它的n-gram子句很有可能都出现过，那么这个句子的概率就可以不为0。

但仍然可能存在预测一段新文本时，n-gram串没有出现在语料库中，如果将它的概率置为0，则会使整个句子的概率变成0。这时，可以采用Laplace平滑的方法来解决这个0概率问题。

第三节 概率语言模型的评价指标（PPL）

我们上面说道，概率语言模型可以计算一句话出现的概率，这个概率能反映句子的“流畅性”，实际中，我们用一个也是基于条件概率但更好的指标——PPL来衡量句子的流畅性。对一段文本，其PPL为：

其中，（这里是基于uni-gram语言模型计算的）

PPL可以用来比较不同语言模型之间的优劣。

第四节 概率语言模型的应用：拼写纠错（spell correction）

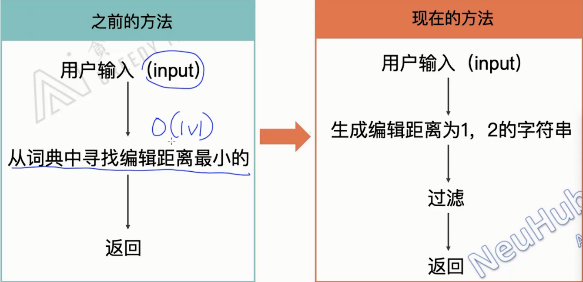


图1 拼写纠错的流程

拼写纠错就是对用户输入的错误的词（如何判断用户输入错误也是需要研究的），给出最可能的正确的词。目前主流的方法是先给定一些“候选词”（与错误词的编辑距离为1或2的词），然后用统计语言模型来找出候选词中最可能的正确词。

问题就是：

其中w是错误词，c是可能的正确词，是最佳候选词。的意义是当用户输入了错误词w时，其真正想输入的正确的词是c的概率。

我们用贝叶斯定理对它变形：

其中的意思是，当用户想输入正确词c时，他有多大可能性会将其输成w。这可以通过统计大量的用户输入的历史来得出，也可以将它当做是词w和词c的相似度（可以用编辑距离等衡量）来给出，因为一个词与正确词的相似度越高，用户越可能输错成该词。

而可以用语言模型来给出，即在当前用户输入的上下文的条件下，词c出现的概率。

# 第四章 神经网络语言模型

前面已经介绍过，神经网络语言模型是概率语言模型的又一种实现方式，它用神经网络结构实现概率计算。

神经网络语言模型作为一种语言模型，它可以使用语言模型的原始目标“给定一段文本，预测后面一个单词出现的概率”作为目标，也可以使用跟语言模型原始的目标稍有不同的目标：输入上下文词，预测其中心词，或者输入中心词，预测其上下文词，可以看出，两种目标的根本目的是一致的，都是为了提取上下文关系信息。

神经网络语言模型的特点是用有监督的方式做无监督的任务，这也称为“自监督”，它用无标注的语料数据中的上下文关系作为标签。

神经网络语言模型的一个重要作用——也可以说是它的一个副产品——就是词向量。

那么相比于统计语言模型，它有什么优势呢？

统计语言模型本身的主要问题是参数量巨大。因为它的参数量跟语料库的词典大小的平方成正比。另外，统计语言模型虽然利用n-gram和Laplace平滑的方法解决了语料稀疏性导致的难以泛化的问题（也就是0概率问题），但是神经网络语言模型能够以更优雅的方式进行泛化。

神经网络语言模型利用神经网络的结构，将上下文关系的信息都蕴含到模型的参数——也就是模型的权重——中（更具体地，将上下文关系蕴含到embedding层所表示的词向量之间的相似性关系之中），极大地压缩了模型参数。与之相比，统计语言模型是将上下文关系的信息都蕴含到条件概率之中。并且神经网络语言模型相当于将离散的数据映射到连续的空间，从而起到泛化的作用。

此外，神经网络语言模型还有一个好处，就是词向量作为词的连续表征方式，可以用来形成句向量。句向量相对于离散的句子表征，对于具体的NLP任务的模型来说，也可以解决因语料稀疏性而导致的难以泛化的问题。

# 第五章 预训练语言模型

预训练语言模型是在大量语料上训练语言模型，然后以feature extraction或者fine-tune的方式应用到下游任务中。

比较著名的预训练语言模型有：ELMO、BERT、GPT、XLNet。其中ELMO是属于双向RNNLM；BERT和GPT分别是Transformer的Encoder和Decoder；XLNet则是BERT的改进版，它的改进点在于它提出了“排列语言模型”。

这里需要说明的是，前面我们提到语言模型的目标是给定一段文本预测下一个单词，其根本动机在于获取语料中上下文信息。这些预训练的语言模型对语言模型的原始目标做了一些扩展，根据使用的具体目标的不同，语言模型可以分为三类：自回归语言模型、自编码语言模型、排列语言模型。

关于预训练模型的细节，我们将在另外一篇笔记中详细介绍。这里解释一下为什么预训练模型可以以fine-tune甚至不需要fine-tune的方式应用到下游任务中：

因为各种下游任务也都是基于同一种语言的，并且很多下游任务非常依赖于文本的表征的质量。而BERT使用这种语言的极其大量的语料，进行无监督学习，它大概率包含了各种下游任务的数据，或者说涵盖了下游任务语料的语义分布，并且尽管预训练模型采用无监督学习，它仍然能够基于这大量的语料，学到词和文本的很好的语义表征，因此将这些好的表征输入到下游任务，就能得到比较好的结果。

比如对于垃圾邮件分类、情感分类等任务，在预训练模型的语义空间中（前面说过，语言模型都对应于一个语义空间，词和文本在该空间中都有比合适的嵌入），垃圾邮件和正常邮件位于不同的区域、正面情感和负面情感的文本也位于不同区域，这样的话，仅仅需要一个简单的分类器，就能够得到比较好的分类结果。

# 第六章 语言模型的关键问题和局限性

前面已经介绍过，语言建模的问题从属于数据建模的问题，而数据建模，无非是数据+模型=任务，也就是说，用数据集训练模型，模型从数据集中学到任务相关的某些信息，然后用这个模型去完成任务。从概率的视角来看，数据建模的过程就是在样本空间中，由样本集去推断总体的分布。

语言建模也是如此，语言建模中的数据就是语料，模型就是语言模型。但是语言建模问题与其他数据建模问题相比，有其独特的性质，比如语言的序列性、上下文相关性等。下面我们分别从数据和模型两个方面深入探讨语言建模的一些问题。

第一节 数据方面的问题

对语言建模来说，数据就是语料。在语言建模或者其他具体的NLP任务建模中，面临的一个大问题是语料的“稀疏性”。它的意思是说，在实际中，无论你收集到多大的语料库，也无法覆盖整个语言的语义模式。

怎么理解这句话呢？这要从一般的数据建模过程出发去理解。我们知道，一般的数据建模就是为了拟合样本和标签之间的关系，或者挖掘数据本身的“模式”（或者说“结构”、“信息”、“规律”等）。对于语言数据来说，这个“模式”就是语义。以句子为单位构成样本，每个句子表达不同的意思，有的句子之间意思相同或者相近，有的句子之间意思相反，有的句子之间意思无关。句子的意思取决于句子中用到哪些词，这些词的排列组合关系是什么，更重要的是，句子的意思还与上下文有关。我们对语言模型的期望并不高——只要能够学到句子之间的语义的相近、相反、无关的这些关系就行。但是语言具有很强的多样性（一个意思可由多个句子表达）、歧义性（一个句子可能有多种意思），以及刚才说的，一个句子的意思与上下文高度相关，也就是语义模式非常的复杂。这就使得我们用有限的语料和简单结构的模型，很难提取到语言的完整的语义模式。这时，我们就说语料具有“稀疏性”，或者说语料的“语义覆盖度”不够。

从概率的视角来看，数据建模就是在样本空间中，用样本去拟合总体分布。那么对于语言模型，我们的目标就是拟合整门语言的“语义分布”。也就是说，我们假设有一个“语义空间”，空间中的每个点表示一个“意思”，而我们的目标就是把语言所能表达出的每个有意义的句子（最好是加上句子的上下文）定位到它表示的意思上去。我们可以想象，在这个空间中的某个区域里密集地分布着众多的句子，有的地方聚集着很多句子，表示那个地方的意思可以用很多的句子来表示；有的地方的句子很稀疏，表示那个地方的意思只能有很少的句子来表示；有的地方甚至没有句子，表示那个地方不是任何实际的意思。

我们使用神经网络进行建模，能够产生连续词向量，进而产生连续句向量，这就相当于构建了一个这样的语义空间。当然，严格说来，实际中我们的模型对应的样本空间是“语法语义空间”，它既包含语法信息，也包含语义信息，因为上下文关系信息是同时包含两者的。

可以看出，从概率的视角来看，语料的稀疏性就是指语料只覆盖了语义空间中的很少一部分点。

那么语料稀疏性有什么影响呢？

对于一般的数据建模过程——用数据训练模型而得总体分布——我们也可以从两个角度来理解：它可以看成是一个“泛化”（样本泛化）的过程，因为我们从估计的总体分布中能够预测训练数据之外的样本；它也可以看成是类似于贝叶斯估计（数据+先验信息=总体分布）的过程，模型所起的作用就是先验，因为我们只知道数据，而没有任何其他先验的话，就只能根据最大熵原则来得到总体分布，而模型相当于对数据施加了先验信息，因此能估计出更好的总体分布，也就是能更好地泛化。

但是对于语言模型，语言具有复杂性，模型又相对简单，相当于施加了较为简单的先验（类似于用一次函数而不是二次函数来拟合数据），当语料稀疏的时候，我们的模型的简单先验就很难正确地将语料泛化到语言的总体语义分布。

以上深入地解释了语料稀疏性的含义及可能造成。但是还有一个问题：实际中，我们目前还无法量化地衡量一个语料的“稀疏性”强弱，或者说一个语料的“语义覆盖度”是多少，并且对于多少的覆盖度才能覆盖整个语言的语义空间（或者说，如何判断一个语料库是“完备的”），或者至少满足我们的具体任务，也不确定。因此未来我们可以考虑发明一种度量语料的“语义覆盖度”的手段，以及多大覆盖度的语料才能使我们可以用简单的模型将语料泛化到语言的总体语义分布。甚至我们可能构建一个可以被认为是语义上“全覆盖”的语料库。

第二节 模型方面的问题

前面已经详细地介绍过了语言建模的目标和方法。这里我们再总结一下：语言建模的目的是从语料中提取上下文关系信息所蕴含的语法语义信息。目前主流的语言模型是基于概率的神经网络语言模型，它的目标是给定一段文本，预测后面的一个单词（或者扩展目标：自编码、排列）。

下面我们讨论目前的语言模型在理解语言方面的局限性：

首先，模型的目标是拟合语义分布，而并非是理解语言。也就是说，模型只关心每个句子在空间中的位置，或者说句子跟句子之间的距离远近，而不关心句子所在位置处的意思究竟有什么意义。那么这样做究竟在多大程度上理解语言？

其次，模型是概率模型，它完全从上下文关系中提取信息，那么基于概率的模型能充分提取上下文关系信息吗？也就是说，我们以预测单词为目标，计算概率分布，那么这个目标对于提取语料中上下文关系信息所蕴含的所有语法语义信息来说，是否充分？

然后，即使模型能充分提取上下文关系信息，但是上下文关系信息所代表的语法语义信息对于理解语言来说，是否足够呢？或者说，上下文关系信息是否包含了我们希望通过语言表达出来的信息的全部呢？

最后，上下文关系信息里既有语法信息，又有语义信息，那么将它们混着都提取出来，合适吗？

# 第七章 新一代语言模型的畅想

以上介绍了语言建模的思想、语言模型的原理和细节，以及目前语言建模的局限性。我们可以看到，“理解语言”是一个终极问题，但目前仍然是一个很模糊的概念。语言模型用朴素的思路，尽其所能地获取语料中蕴含的信息，但仍然很难说它真正理解了语言。

为了克服现有的语言模型的局限性。我们需要跳出其框架，构想一些新的语言建模的思路。

第一个思路就是，放弃基于概率进行建模的思路。诚然，以计算概率为目标，是提取上下文关系信息的最自然、最朴素的思路，但是它很可能在提取上下文关系信息上是有局限的，因此可以考虑基于新的形式的目标构建模型。

第二个思路是，放弃神经网络结构的“连续空间”思路。我们前面已经介绍过神经网络模型的特点，它将离散的语料样本映射到连续的、数值的语义空间，以求得泛化能力，但是在语料稀疏的情况下，它无法保证泛化的很好。因此我们可以不拘泥于连续空间，甚至不拘泥于数值空间，而是考虑使用其他的结构（比如图、群等）来构建模型。

第三个思路是，构建可扩展的（或者说可进化的），真正能“持续学习”的模型结构。它可以从简单的状态出发，慢慢向里面加入信息（或数据），从而使它的性能慢慢提升。传统的语言模型，统计语言模型和神经网络语言模型，都太过于依赖数据，并且都是“完全让模型自己从数据中学习规律”这种策略，这使得它们的扩展性很差，很难通过不断加入新的数据或者信息来稳步提升性能。

使用新的结构和新的策略来构建语言模型，不能囿于传统的思路。不必过分依赖数据，不必想着只让模型自己学习而不加指导。

但是一定要注意，不要盲目地去构想新的模型，要先考虑我们究竟希望模型能学到什么信息、希望它能干什么。我们一般都说希望模型能充分地学到语言的语义信息，但实际上对这个“语义信息”我们无法给出清晰的定义，对于传统的模型，我们也无法清晰地说它是否充分地学到了语义信息、它学到了多少语义信息。而我们又希望模型学到“知识”和具备“推理能力”，那么我们训练模型用的大量的语料中，究竟是否充分蕴含了这两项信息？如果没有充分蕴含，那么传统的语言模型这种过分依赖数据的建模就无法使模型学到“知识”和具备“推理能力”。

# 第八章 附录

第一节 对“泛化”概念的理解

我们一般所说的模型的泛化，其实有两层意思：样本泛化和模式泛化。

1、样本泛化：是指模型能够对不在训练集中的样本给出预测。这是模型的基本能力。是个模型就需要具备这种能力。我们知道，有监督模型的一个原则是：对相似的输入给出相似的输出。这就要求样本之间能计算相似度。如果样本之间能计算相似度，那么即使新的样本不在训练集中，模型也能根据学到的相似关系估计出该样本的最佳输出。

对于NLP的模型来说，一般输入的是词的id，无法计算相似度，因此需要接一层embedding层，将id映射到连续空间，从而能够计算相似度。或者直接输入预训练好的词向量。

在论文《A Neural Probabilistic Language Model》中，作者说NNLM能产生词向量，给使用它作为输入的模型带来泛化性，这里的泛化指的就是样本泛化。

2、模式泛化：是“泛化”的原本意思，是指对给定的训练集，模型能够很好地拟合出总体分布。最终拟合的结果与训练集和模型（作为先验）都有关，因此泛化性也由这两个因素共同决定。训练集越大、从总体分布中采样的随机程度越高，则泛化性越好；模型的先验越符合总体的特性，泛化性越好。