中文信息处理期末 Project

对文言文进行选词填空

10302010029 田应涛

背景以及具体描述

本次任务选为实现计算机自动对文言文片段选词填空。其形式如下，对于一个文言文句子、一个空位以及若干选项，选出最合适的选项填入其中。例如，

學「而/以/爲/故」時習之，不亦說乎。（《論語》）

选题的动机主要是个人兴趣。在中学时期学习文言文之时，即遇到此类问题，当时的主要考察点是虚词的用法，考察方法如同上述例子，空出一空，要求选择合适的虚词填入。因此现在希望利用课程知识以及以往的研究经验，尝试解决这个问题。在解决问题途中，使用了 Recurrent Neural Network, Language Model, N-Gram 的模型和思想，并在单纯的选择之外，给出了每个选项符合句子的概率以给出更多信息。

关键词

Recurrent Neural Network, Language Model, N-Gram

语料选取

选取语料有若干考量。其中最为重要的是，文言文作为以先秦口语（特制指的是春秋战国时期的典雅口语）为基础的一种书面语言，随着时代的发展与当世口语差距愈发拉大，后世作品效仿先秦语言难免受到当世口语影响。为此所选语料应该充分反映语言特征和时代特征。我选取的是先秦两汉的经典，具体而言，主要是春秋战国诸子百家经典加上两汉的诸多论述（以史书为主）。百家经典作为先秦口语的代表，为文言文经典，不可不用；两汉去秦不远，语言几乎一致，其中的史书（《史记》，《汉书》等）文学价值尤其高。因此，选取语料如下：

類別	書目
儒家	《論語》《孟子》《禮記》《荀子》《說苑》《春秋繁露》《韓詩外傳》《大戴禮記》《新書》《新序》《孔子家語》

	《潛夫論》《論衡》《風俗通義》《孔叢子》《申鑒》《新語》《蔡中郎集》
墨家	《墨子》
道家	《莊子》《列子》《文子》《鬻子》《老子河上公章句》
法家	《韓非子》《商君書》《管子》
名家	《公孫龍子》
史書	《史記》《逸周書》《國語》《吳越春秋》《越絕書》《戰國策》《鹽鐵論》《列女傳》《春秋穀梁傳》《春秋公羊傳》《漢書》《前漢紀》《東觀漢記》《後漢書》《竹書紀年》《穆天子傳》《西京雜記》

其中，选取诸子百家保证了形式上覆盖文言文的诸多可能形式，同时也不能用简单的数条手写规则解决问题，选取诸多史书的目的在于，史书往往是一个人或者一组熟识的人编纂而成，遣词造句统一。所有选取的文书无论诸子百家或是史籍文献均为大师所著述，以保证语言质量，同时大量的史籍也提供了充足的数据用于训练。

先秦典籍中有一些经典文献没有选为语料，兹列举如下并说明缘由：

- 《尚书》。《尚书》艰涩难懂，其中保存大量周代以至于商代语言（例如：「爾尚輔予一人致天之罰，予其大賚汝。爾無不信，朕不食言。爾不從誓言，予則孥戮汝，罔有攸赦。」《尚书・湯誓》），表达方式迥异于春秋战国语言。因而不选为语料。
- 《易经》。周易艰涩，释文可用但是不便自文本中分离，因而不选为语料。
- 《楚辞》。楚辞体虽然文学水平甚高，但是语言形式与我们所感兴趣之文字大相径庭，因而不选为语料。

语料来源

如何获得语料是一个技术性的问题。古汉语文学方面参考的数字化资料，有系统性整理的数据库，也有爱好者自发架设的网站。其中我最为满意的是「中國哲學書電子化計劃」（http://ctext.org），库存详尽，整理明晰。但是注解繁冗，格式不一，且限于授权不得自动化下载，因此我退而求其次，依照书单，在「中文版维基文库」（https://wikisource.org/wiki/Main_Page）上爬取相关书目，为此特地写程序以自动化爬取过程。

实验步骤

处理文本

文本从维基文库爬取之时就使用了 Unicode。以 UTF-8 编码，可以免去文字处理中的诸多烦恼。

首先去除标点。鉴于之后预计使用 RNN 语言模型，因此我希望将文本拆解为一个一个句子，其做法是把“。？！”作为分隔符，其他标点直接忽视。

维基文库的书目质量较高，但是并非到达可以直接使用的程度，需要进行再处理。经过观察可以得知，数据略有噪音，主要是一些非汉字字符，考虑到噪音字符数量极少，在这一步直接排除了这些字符，只保留「Unicode 中日韩统一表意文字列表」（即 Unicode 编码 4E00 到 9FFF）中的字符。

数据分割

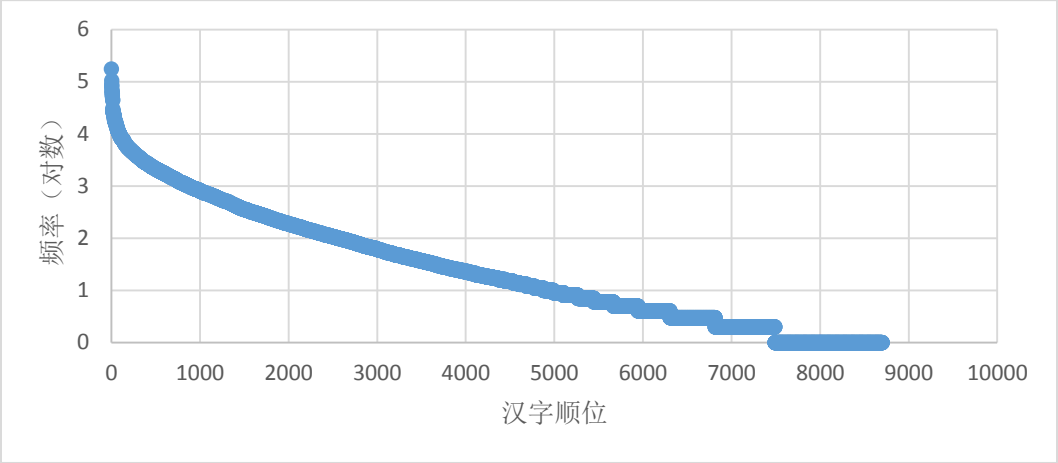
经过以上操作，已经得到以句子为单位的干净的文言文语料。之后需要分割测试集、训练测试集以及测试集，我采用 8:1:1 的划分方法。具体做法是将句子之间的顺序打乱，按照 8:1:1 的比例进行划分成三部分。

初步分析

初步的统计可知，在训练集之中一共有 263914 个句子，4502423 个汉字，平均每个句子长 17 个字。不同的汉字有 8450 个。最为高频的汉字如下表

汉字	频率
176233	之
104809	不
89370	也
83752	以
78781	而

依据下表亦可获知，汉字频率分布符合**幂律分布**(Power Law)



反馈神经网络(Recurrent Neural Network)语言模型(Language Model)

本次 Project 的核心，是使用**反馈神经网络**的**语言模型**。

为能够做到选词填空，一个最直接的思路是将每一选项带入之后，对句子**打分**(Scoring)，再根据分数做出抉择。

语言模型能够就给定的一个句子$\mathrm{A}_{1\ldots\mathrm{m}}$求出其**似然度**(likelihood) $\mathrm{P}(\mathrm{A}_{1\ldots\mathrm{m}})$，这个似然度可以简单地视为「给定句子有多么可能是一个合乎语言规律的句子」。因此一个正确合理的搭配（「學**以**致用」）将会比一个错误的搭配（「學**而**致用」）得到更高的分数。

依此性质，若将打分视为求出似然度，则原问题转化为对文言文构建语言模型，即在带入选项生成的句子之中，选择得分数最高者作为答案。

语言模型最为基本的方法是使用 N-gram 模型，即使用如下公式作为似然度的近似

$$\mathrm{P}(\mathrm{A}_{1\ldots\mathrm{m}}) = \mathrm{P}(\mathrm{A}_{1\ldots\mathrm{m}-1}) \times \mathrm{P}(\mathrm{A}_{\mathrm{m}}|A_{m-n+1\ldots m-1})$$

其中的第二项，对于 2-gram 是$\mathrm{P}(\mathrm{A}_{\mathrm{m}}|A_{m-1})$，对于 3-gram 是$P(A_m|A_{m-2}A_{m-1})$。其计算方法是

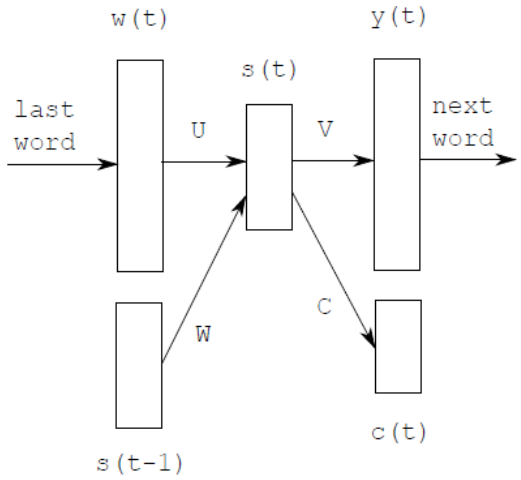
$$\mathrm{P}(\mathrm{A}_{\mathrm{m}}|A_{m-n+1\ldots m-1}) = \frac{\mathrm{C}(\mathrm{A}_{\mathrm{m}-\mathrm{n}+1\ldots\mathrm{m}})}{\mathrm{C}(\mathrm{A}_{\mathrm{m}-\mathrm{n}+1\ldots\mathrm{m}-1})}$$

根据我的经验，如此做虽然能够得出一个勉强能用的模型，但是数据的稀疏性质是无法避免的。考虑到文言文语料有限，则更可能导致训练语料中根本没有出现对应的 N-gram。在这种情况下，即使使用了**平滑**(Smoothing)技术，效果也令人堪忧。

在这种情况下，我采用了今年在微软亚洲研究院实习期间接触到的**反馈神经网络语言模型**(Recurrent Neural Network Language Model)，试图进行一些深度的学习，以便挖掘出浅显的统计无法覆盖的深度关系。

此模型的论文出自「RNNLM - Recurrent Neural Network Language Modeling Toolkit」，作者是 T. Kikolov，在参考文献中也一并列出。以下部分为我的简要说明，翻译自论文的引言和方法部分。

> 统计语言模型是诸多实用系统中的重要组成部分，任何重大进展都能立即应用于现有的语言识别和统计机器翻译系统中。但是，整个研究领域在几十年间一直都在力争超越极为简单却非常有效的 N-gram 的模型。有的模型虽然一时在小规模上以计算复杂性为代价超越了 N-gram，然而终究还是败于海量数据训练出来的 N-gram 模型。
>
> 反馈神经网络语言模型既能拥有神经网络的优秀表现，又可以避免巨大的训练用时。同时，反馈神经网络可以有所记忆，因而能够克服 N-gram 模型的最大问题，即只考虑临近的若干字词。
>
> 本文中提出一种基于反馈神经网络和最大熵模型的语言模型，可以降低计算复杂性，同时达到很好的效果。
>
> 所使用的反馈神经网络结构如下



> 其中，输入层由 N 取 1 表示的前一个单词w(t)和上一次隐藏层的状态s(t − 1)连接而成。隐藏层的s(t)的神经元使用 Sigmoid 函数作为激励函数。输出层的大小和w(t)一致，在训练之后表示的是基于当前隐藏层状态和前一个单词所给出的下一个单词的概率分布。分类层c(t)可以用来减少计算复杂性。

> 训练神经网络使用的是标准的**随机梯度下降**(stochastic gradient descent)算法，反馈神经网络的训练使用**随时间反向传播**(backpropagation through time algorithm)算法

上述文章提供的算法实现在 RNNLM 工具包中，因此可以直接调用。

使用 N-gram 模型提高语言模型精度

单纯的使用神经网络效果尚可，但是对于常用搭配而言，N-gram 的优势还是非常明显，尤其是双字的搭配（例如「天子」「诸侯」「国家」等等）。为此，我希望在反馈神经网络语言模型之外，加入 N-gram 模型用以辅助提升精度。

具体的做法的精要在于语言模型可以线性叠加。因此假设神经网络得到的分数为$score_a$，N-gram 得到的分数为$score_b$，则可使用参数λ做如下叠加

$$score = \lambda score_a + (1-\lambda) score_b$$

实践上 N-gram 模型使用了 SRILM(SRILM - The SRI Language Modeling Toolkit)的 ngram 工具。

使用 Logit 模型选取最终答案

得出分数（即似然度）之后，即可选择最大分数对应选项作为答案。但是我想进一步的分析比较每一个选项，因而在此计算每一个选项符合的概率。

假设选项$i(1 \le i \le n)$的分数为s_i，则其被选择的概率应为

$$P_i = \frac{e^{s_i}}{\sum_{j=1}^{n} e^{s_j}}$$

因此在「选择概率最大者作为答案」之外，还能够比较各个选项之间的关系。

实验与分析

实验结果

实验数据使用**测试集**中的句子作为数据。

对于语言模型在测试集上的表现，我使用 PPL 值作为评价标准，PPL 值越低越好。对于测试集，语言模型的 PPL 值为 92.2064。

以下对一些具体的例子进行分析。方式是从测试集中摘出句子，人工选取空位，构造选项，并加以解读。这些例子的选取各有其着重点。例如，若摘出的句子选取空位为「和」字，则构造选项「和」「平」「而」「與」。选项的四个字分别与「和」在不同的意义下可以互相替代，因此若语言模型能够选出正确的「和」字，则说明模型本身由足够的能力「理解」语言。

测试结果

正解	題目及結果	選項概率

不	玄成[不]得已受爵	**不(74%)** 無(4%) 未(14%) 而(12%)
不	羅亦覺之[不]敢發	**不(56%)** 無(12%) 未(14%) 而(18%)
不	雖驟立[不]過五矣	**不(82%)** 無(8%) 未(3%) 而(8%)
不	[不]然父子俱屠無為也	**不(51%)** 無(16%) 未(17%) 而(16%)
之	願王之知[之]	**之(46%)** 此(22%) 其(27%) 是(4%)
之	三年吳王歸[之]	**之(71%)** 此(10%) 其(10%) 是(9%)
之	失之豪釐差[之]千里	**之(90%)** 此(4%) 其(4%) 是(2%)
之	如是而死何恨[之]有	**之(54%)** 此(6%) 其(30%) 是(10%)
以	吾所[以]亡者其何哉	**以(94%)** 而(3%) 或(1%) 其(2%)
以	嘗[以]諸侯與之接矣	**以(63%)** 而(22%) 或(3%) 其(13%)
以	守柔弱日[而]強大也	以(43%) **而(47%)** 或(3%) 其(6%)
以	水可載舟亦[以]覆舟	**以(68%)** 而(11%) 或(10%) 其(11%)
而	刑一而正百殺一[以]慎萬	而(26%) 此(15%) **以(37%)** 其(22%)
而	陽名成功故九會[而]終	**而(54%)** 此(7%) 以(19%) 其(20%)
而	即取[而]歸之於諸侯	**而(62%)** 此(9%) 以(10%) 其(19%)
而	夫戰[而]忘勇非孝也	**而(100%)** 此(0%) 以(0%) 其(0%)
以	子曰君子博學於文約之[以]禮亦可以弗畔矣夫	**以(96%)** 而(1%) 或(1%) 其(2%)

就结果而言，正确答案多数对应高概率，说明语言模型的表现是很好的。以下分析两个与原典相异之处。

- 「守柔弱日以強大也」，此时两个选项「以」和「而」的概率非常接近。两个选项中，「以」对应的是「守/柔弱/日以/強大/也」，「而」对应的是「守/柔弱日/而/強大/也」。
  缘由推定如下：本模型用以文言的「字」对应西文的词，因此文言的常用搭配（例如「柔弱」）对应的是西文的短语。本模型应用于西文之时即未考虑短语，这是尚缺之处。
- 「刑一而正百殺一而慎萬」，此处的「以」和「而」均可，语义两者皆通顺，原典选用「而」目的在于互文对仗。因此此相异可以接受。

改进

在做 Project 之中，我也曾尝试加入字的向量表示(vector representation)作为每个字的单独特征，但是终究没有能够将其整合入现有模型。这是一个可以探究的方向。

还有一个可以改进的地方是语料的分布。经典和史书由于叙述风格略有不同，因此在语料中的分布也应该加以探究，而非如同现在这般全部等权值列入语料库中。

感想

本次的语言模型能够做到这样的结果远远超出我的预期。当自己构造的语言模型能够「理解」文言文的时候，我很高兴。这次的 Project 避免了许多例如编码、语料下载的问题，没有在细枝末节上浪费时间，也是因为之前在研究院诸多项目上已经受过非常多挫折，积累了不多不少的一些 NLP 的经验。本次的 Project 过程之中，在选题、模型、技巧等诸多方面与课内课外不少同学有所交流，获益良多，在此一并表达谢意。

参考文献与所用工具

- RNNLM
  - Toolkit: http://www.fit.vutbr.cz/~imikolov/rnnlm/
  - Paper: http://research.microsoft.com/pubs/175562/ASRU-Demo-2011.pdf
- SRILM
  - Toolkit: http://www.speech.sri.com/projects/srilm/