維基百科

語言模型

维基百科，自由的百科全书

統計式的語言模型是一個機率分佈，给定一个长度为 m 的字詞所組成的字串 $w_1, w_2, \ldots, w_m$，派機率給字串：$P(w_1, \ldots, w_m)$。

语言模型提供上下文来区分听起来相似的单词和短语。例如，短语“再给我两份葱，让我把记忆煎成饼”和“再给我两分钟，让我把记忆结成冰”听起来相似，但意思不同。

語言模型經常使用在許多自然語言處理方面的應用，如語音識別[1]，機器翻譯[2]，詞性標註，句法分析[3]，手写体识别[4]和資訊檢索。由於字詞與句子都是任意組合的長度，因此在訓練過的語言模型中會出現未曾出現的字串(資料稀疏的問題)，也使得在語料庫中估算字串的機率變得很困難，這也是要使用近似的平滑n-元語法(N-gram)模型之原因。

在語音辨識和在資料壓縮的領域中，這種模式試圖捕捉語言的特性，並預測在語音串列中的下一個字。

在语音识别中，声音与单词序列相匹配。当来自语言模型的证据与发音模型和声学模型相结合时，歧义更容易解决。

當用於資訊檢索，語言模型是與文件有關的集合。以查詢字「Q」作為輸入，依據機率將文件作排序，而該機率$P(Q|M_d)$代表該文件的語言模型所產生的語句之機率。

目录

模型类型

单元语法（unigram）

一个单元模型可以看作是几个单状态有限自动机的组合[5]。 它会分开上下文中不同术语的概率, 比如将 $P(t_1 t_2 t_3) = P(t_1)P(t_2 \mid t_1)P(t_3 \mid t_1 t_2)$ 拆分为$P_{\text{uni}}(t_1 t_2 t_3) = P(t_1)P(t_2)P(t_3)$.

在这个模型中，每个单词的概率只取决于该单词在文档中的概率，所以我们只有一个状态有限自动机作为单位。自动机本身在模型的整个词汇表中有一个概率分布，总和为1。下面是一个文档的单元模型。

单词 term	在文档 doc 中的概率
a	0.1
world	0.2
likes	0.05
we	0.05
share	0.3
...	...

$$\sum_{\text{term in doc}} P(\text{term}) = 1$$

为特定查询(query)生成的概率计算如下

$$P(\text{query}) = \prod_{\text{term in query}} P(\text{term})$$

不同的文档有不同的语法模型，其中单词的命中率也不同。不同文档的概率分布用于为每个查询生成命中概率。可以根据概率对查询的文档进行排序。两个文档的单元模型示例:

单词	在Doc1的概率	在Doc2中的概率
a	0.1	0.3
world	0.2	0.1
likes	0.05	0.03
we	0.05	0.02
share	0.3	0.2
...	...	...

在信息检索环境中，通常会对单语法语言模型进行平滑处理，以避免出现P(term)= 0的情况。一种常见的方法是为整个集合生成最大似然模型，并用每个文档的最大似然模型对集合模型进行线性插值来平滑化模型。[6]

n-元语法

在一个 n-元语法模型中，观测到序列 $w_1, \ldots, w_m$ 的概率 $P(w_1, \ldots, w_m)$ 可以被近似为

$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_i \mid w_1, \ldots, w_{i-1}) \approx \prod_{i=1}^{m} P(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1})$$

此处我们引入马尔科夫假设，一个词的出现并不与这个句子前面的所有词关联，只与这个词前的 n 个词关联（n阶马尔科夫性质）。在已观测到 i-1 个词的情况中，观测到第i个词 w_i 的概率，可以被近似为，观测到第i个词前面n个词（第 i-(n-1) 个词到第 i-1 个词）的情况下，观测到第i个词的概率。第 i 个词前 n 个词可以被称为 n-元。

条件概率可以从n-元语法模型频率计数中计算:

$$P(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1}) = \frac{\text{count}(w_{i-(n-1)}, \ldots, w_{i-1}, w_i)}{\text{count}(w_{i-(n-1)}, \ldots, w_{i-1})}$$

术语 二元语法(**bigram**) 和三元语法(**trigram**) 语言模型表示 n = 2 和 n = 3 的 *n*-元 [7]。

典型地，n-元语法模型概率不是直接从频率计数中导出的，因为以这种方式导出的模型在面对任何之前没有明确看到的n-元时会有严重的问题。相反，某种形式的平滑是必要的，将一些总概率质量分配给看不见的单词或n-元。使用了各种方法，从简单的“加一”平滑(将计数1分配给看不见的n-元，作为一个无信息的先验)到更复杂的模型，例如Good-Turing discounting或 back-off 模型。

例子

在二元语法模型中 ($n = 2$) , *I saw the red house* 这个句子的概率可以被估计为

$$\begin{aligned} & P(\mathrm{I, saw, the, red, house}) \\ \approx\ & P(\mathrm{I} \mid \langle s \rangle) P(\mathrm{saw} \mid \mathrm{I}) P(\mathrm{the} \mid \mathrm{saw}) P(\mathrm{red} \mid \mathrm{the}) P(\mathrm{house} \mid \mathrm{red}) P(\langle /s \rangle \mid \mathrm{house}) \end{aligned}$$

而在三元语法模型中，这个句子的概率估计为

$$\begin{aligned} & P(\mathrm{I, saw, the, red, house}) \\ \approx\ & P(\mathrm{I} \mid \langle s \rangle, \langle s \rangle) P(\mathrm{saw} \mid \langle s \rangle, I) P(\mathrm{the} \mid \mathrm{I, saw}) P(\mathrm{red} \mid \mathrm{saw, the}) P(\mathrm{house} \mid \mathrm{the, red}) P(\langle /s \rangle \mid \mathrm{red, house}) \end{aligned}$$

注意前 n-1 个词的 n-元会用句首符号 <s> 填充。

指数型

最大熵语言模型用特征函数编码了词和n-元的关系。

$$P(w_m|w_1,\ldots,w_{m-1}) = \frac{1}{Z(w_1,\ldots,w_{m-1})} \exp(a^T f(w_1,\ldots,w_m))$$

其中 $Z(w_1,\ldots,w_{m-1})$ 是分区函数, a 是参数向量， $f(w_1,\ldots,w_m)$ 是特征函数。

在最简单的情况下，特征函数只是某个n-gram存在的指示器。使用先验的 a 或者使用一些正则化的手段是很有用的。

对数双线性模型是指数型语言模型的另一个例子。

外部链接

- LMSharp (https://lmsharp.codeplex.com/) （页面存档备份 (https://web.archive.org/web/20171227064458/https://lmsharp.codeplex.com/)，存于互联网档案馆） - 开源统计语言模型工具包，支持n-gram模型（Kneser-Ney平滑），以及反馈神经网络模型（recurrent neural network model）

1. Kuhn, Roland, and Renato De Mori. "A cache-based natural language model for speech recognition (https://www.researchgate.net/profile/Roland_Kuhn2/publication/3191800_Cache-based_natural_language_model_for_speech_recognition/links/004635184ee5b2c24f000000.pdf)." IEEE transactions on pattern analysis and machine intelligence 12.6 (1990): 570-583.
2. Andreas, Jacob, Andreas Vlachos, and Stephen Clark. "Semantic parsing as machine translation (https://www.aclweb.org/anthology/P13-2009) （页面存档备份 (https://web.archive.org/web/20200815080932/https://www.aclweb.org/anthology/P13-2009)，存于互联网档案馆）." Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2013.
3. Andreas, Jacob, Andreas Vlachos, and Stephen Clark. "Semantic parsing as machine translation (https://www.aclweb.org/anthology/P13-2009) （页面存档备份 (https://web.archive.org/web/20200815080932/https://www.aclweb.org/anthology/P13-2009)，存于互联网档案馆）." Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2013.

4. Pham, Vu, et al. "Dropout improves recurrent neural networks for handwriting recognition (https://arxiv.org/pdf/1312.4569)." 2014 14th International Conference on Frontiers in Handwriting Recognition. IEEE, 2014.
5. Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze: An Introduction to Information Retrieval, pages 237–240. Cambridge University Press, 2009
6. Buttcher, Clarke, and Cormack. Information Retrieval: Implementing and Evaluating Search Engines. pg. 289–291. MIT Press.
7. Craig Trim, *What is Language Modeling?* (http://trimc-nlp.blogspot.com/2013/04/language-modeling.html)（页面存档备份 (https://web.archive.org/web/20201205054905/http://trimc-nlp.blogspot.com/2013/04/language-modeling.html)，存于互联网档案馆）, April 26th, 2013.
