Natural Language Processing in Action

Understanding, analyzing, and generating text with Python

Stephen CUI

April 20, 2023

Part I

Wordy machines

Chapter 1

Packets of thought(NLP overview)

1.1 自然语言与编程语言

Definition 1. 自然语言处理是计算机科学和人工智能（*artificial intelligence*，*AI*）的一个研究领域，它关注自然语言（如英语或汉语普通话）的处理。这种处理通常包括将自然语言转换成计算机能够用于理解这个世界的数据（数字）。同时，这种对世界的理解有时被用于生成能够体现这种理解的自然语言文本（即自然语言生成）。

1.2 计算机“眼”中的语言

1.2.1 一个简单的聊天机器人

受计算资源所限，早期的NLP研究人员不得不使用人类大脑的计算能力来设计和手动调整复杂的逻辑规则来从自然语言字符串中提取信息。这称为 **基于模式（pattern）**的NLP方法。这些模式就像正则表达式那样，可以不仅仅是字符序列模式。NLP还经常涉及词序列、词性或其他高级的模式。核心的NLP构建模块（如词干还原工具和分词器）以及复杂的端到端NLP对话引擎（聊天机器人）（如ELIZA）都是通过这种方式，即基于正则表达式和模式匹配来构建的。基于模式匹配NLP方法的艺术技巧在于，使用优雅的模式来获得想要的内容，而不需要太多的正则表达式代码行。

经典心智计算理论

这种经典的NLP模式匹配方法是建立在心智计算理论（computional theory of mind，CTM）的基础上的。CTM假设类人NLP可以通过一系列处理的有限逻辑规则集来完成。在世纪之交，神经科学和NLP的进步导致了心智“连接主义”理论的发展，该理论允许并行流水线同时处理自然语言，就像在人工神经网络中所做的那样。

1.2.2 另一种方法

为数值序列和向量设计的距离度量方法对一些NLP应用程序来说非常有用，如拼写校正器和专有名词识别程序。所以，当这些距离度量方法有意义时，我们使用这些方法。但是，针对那些我们对自然语言的含义比对拼写更感兴趣的NLP应用程序来说，有更好的方法。对于这些NLP应用程序，我们使用自然语言词和文本的向量表示以及这些向量的一些距离度量方法。

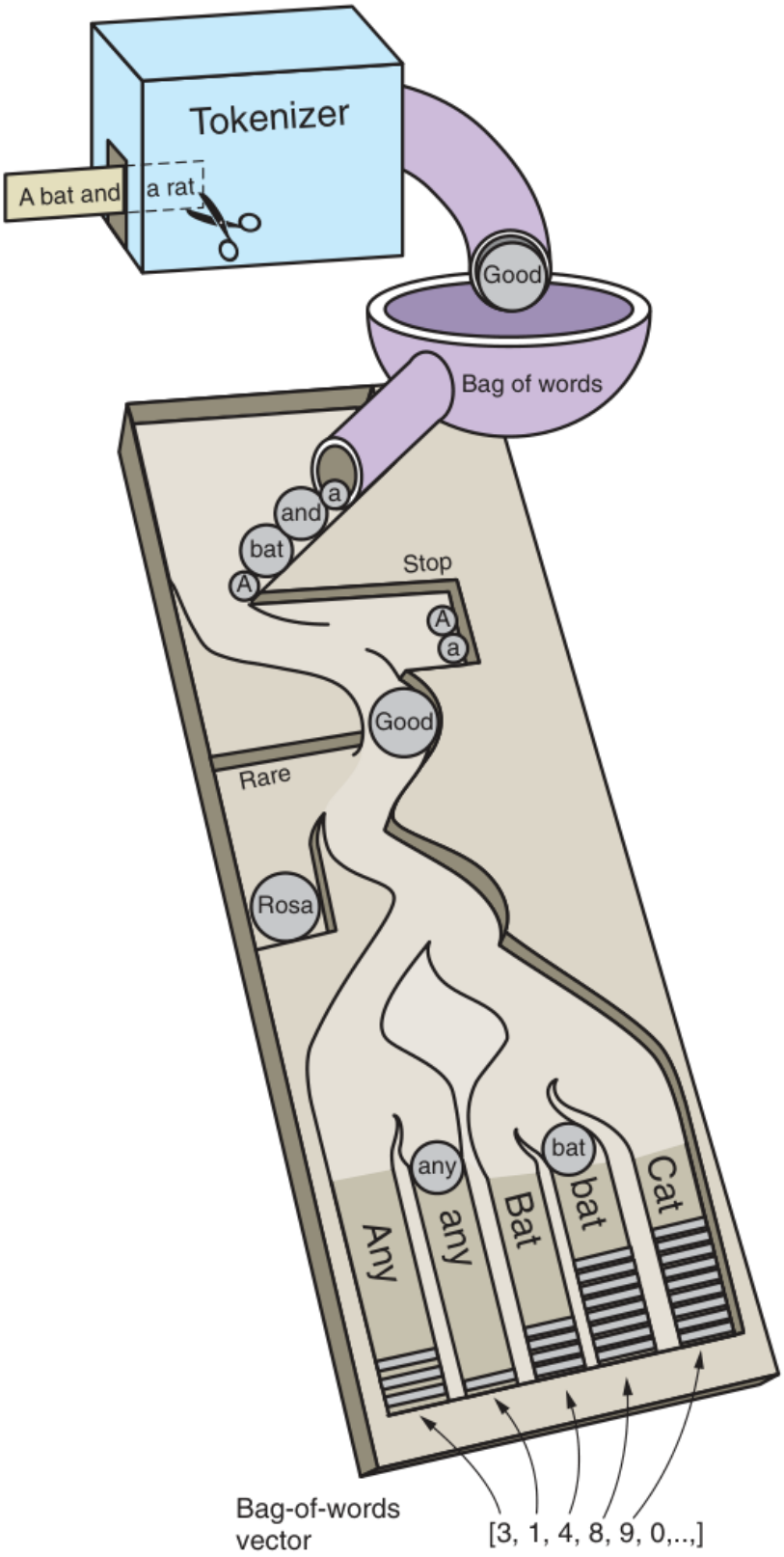

Figure 1.1: 词条分拣托盘

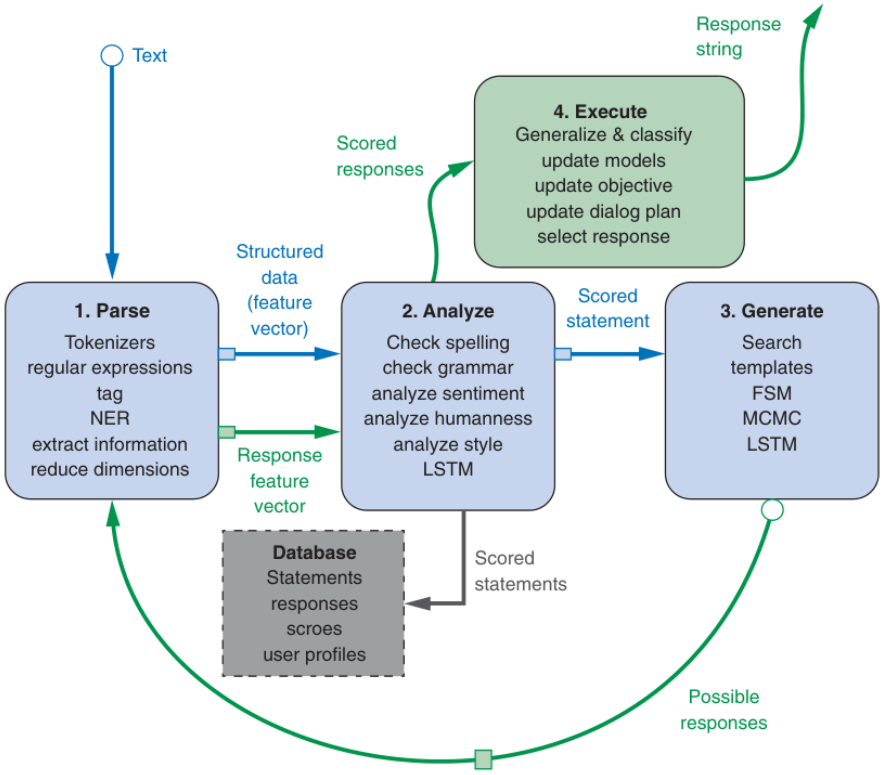

Figure 1.2: Chatbot recirculating (recurrent) pipeline

1.3 聊天机器人的自然语言流水线

聊天机器人需要4个处理阶段和一个数据库来维护过去语句和回复的记录。这4个处理阶段中的每个阶段都可以包含一个或多个并行或串行工作的处理算法（如 Figure 1.2 所示）。

1. 解析：从自然语言文本中提取特征、结构化数值数据。
2. 分析：通过对文本的情感、语法合法度及语义打分，生成和组合特征。
3. 生成：使用模板、搜索或语言模型生成可能的回复。
4. 执行：根据对话历史和目标，规划相应语句，并选择下一条回复。

Chapter 2

构建自己的词汇表-分词

2.1 挑战（词干还原预览）

所谓词干还原，指的是将某个词的不同屈折变化形式统统“打包”到同一个“桶”或者类别中。

2.2 利用分词器构建词汇表

2.2.1 点积

点积也称为内积（inner product），这是因为两个向量（每个向量中的元素个数）或矩阵（第一个矩阵的行数和第二个矩阵的列数）的“内部”维度必须一样，这种情况下才能相乘。这和关系数据库表的内连接（inner join）操作很类似。

点积也称为标积（scalar product），因为其输出结果是个单独的标量值。这使其有别于叉积（cross product）这个概念，后者的输出结果是一个向量。

> 点积和矩阵乘积（matrix product）计算是等价的，后者可以在numpy中使用np.matmul()函数或者@操作符来实现。由于所有的向量都可以转换成 $N \times 1$或者$1 \times N$的矩阵，因此可以使用这种短记号作用于两个列向量（$N \times 1$），其中第一个向量必须要转置，这样才可以相乘。

2.2.2

2.2.3

2.2.4