CS 330 MIP – Lecture 05

文本信息处理

Text Information Processing

Jimmy Liu 刘江

2025-03-19

2. DeepSeek LLM

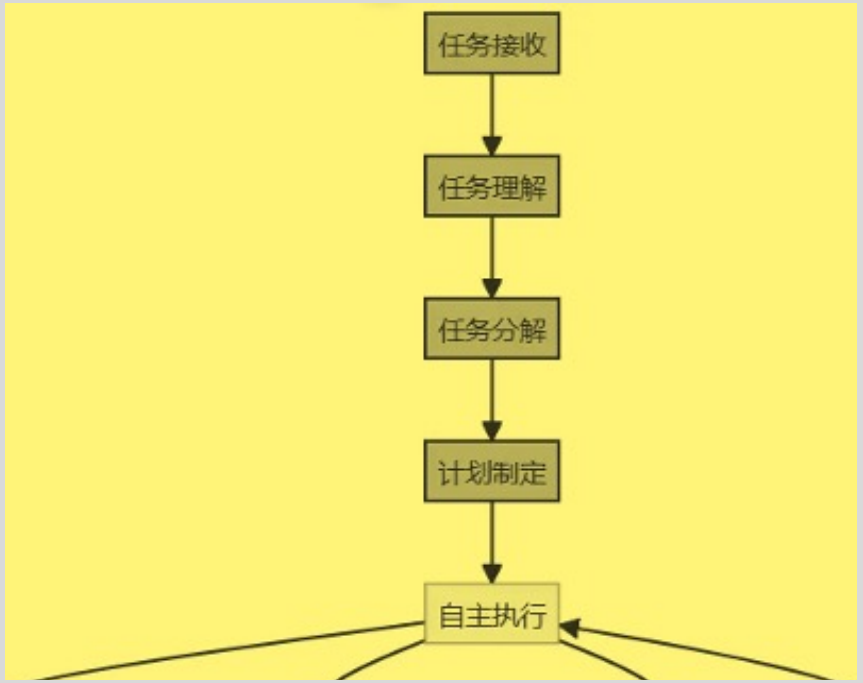
3. AI Agent

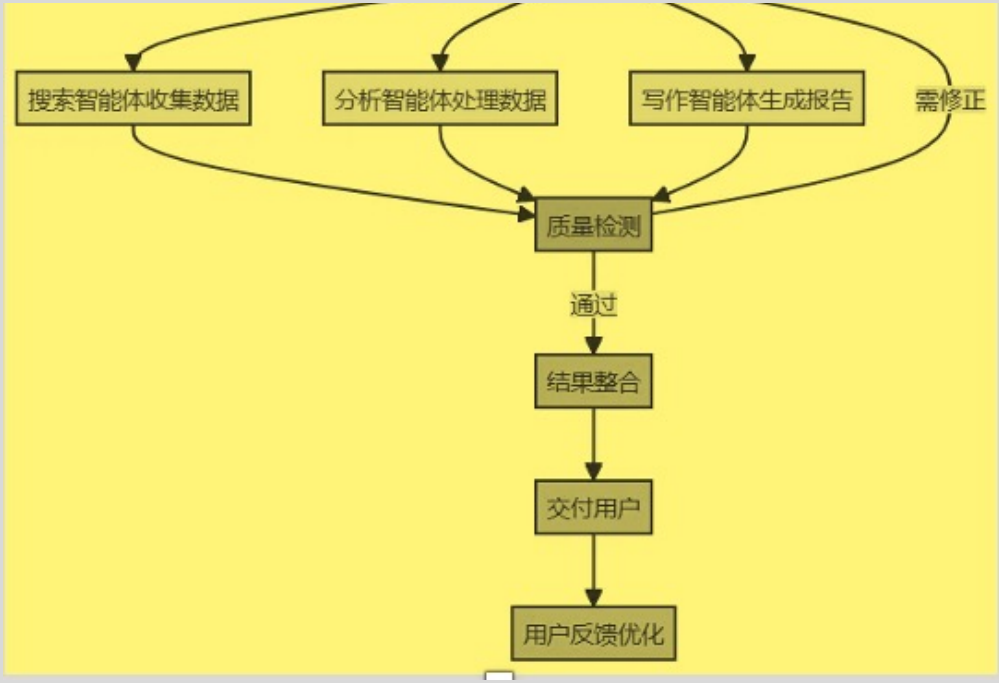
AI Agent（人工智能代理）通常指的是一个软件实体，它可以在某种程度上模拟人类智能的某些方面，执行特定任务或达成目标。这些代理可以感知环境，并在此基础上进行决策并采取行动，从而完成既定的任务或解决问题。

AI Agent工作流的四种设计模式

- 反思（Reflection）：LLM 检查自己的工作，以提出改进方法。
- 工具使用（Tool use）：LLM 拥有网络搜索、代码执行或任何其他功能来帮助其收集信息、采取行动或处理数据。
- 规划（Planning）：LLM 提出并执行一个多步骤计划来实现目标（例如，撰写论文大纲、进行在线研究，然后撰写草稿.....）。
- 多智能体协作（Multi-agent collaboration）：多个 AI 智能代理一起工作，分配任务并讨论和辩论想法，以提出比单个智能体更好的解决方案。

AI Agent 工作流


任务接收
任务理解
任务分解
计划制定
自主执行


搜索智能体收集数据
分析智能体处理数据
写作智能体生成报告
需修正
质量检测
通过
结果整合
交付用户
用户反馈优化

代理式AI Agentic AI或AI智能体 AI Agent

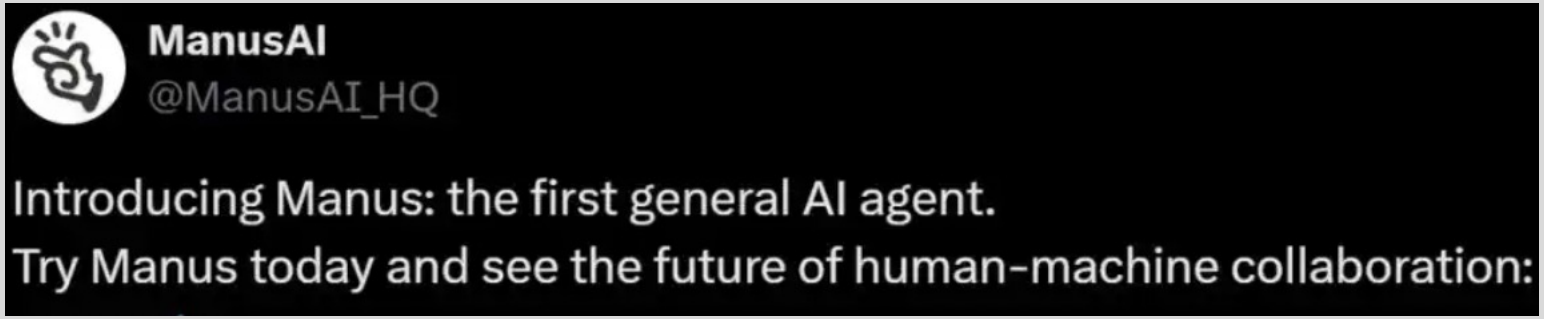
- OpenAI推出名为“Operator”的AI智能体，称可完成一些网购和订票等具体任务。Anthropic和谷歌等企业，也相继发布AI智能体或是用于开发代理式AI的模型套件，并预言智能体可能终会成为公司“员工”之一。
- 亚马逊（Amazon）2月26日宣布为迭代后的AI助理“Alexa”配备代理能力。SAP和ServiceNow等国际软件巨头，也相继投资研发AI智能体工具，并迅速推向市场。

LLM 与 AI Agent

- LLM 的核心，是一个深度思考的大语言模型，它通过逻辑缜密的思考链（COT：Chain of Thoughts），让思考变得更加深入。
- 代替人类作出决策并独立完成复杂任务的AI Agent的核心，就是通过一个个任务清单，详细地拆开任务，一步步执行这个执行链（COA：Chain of Actions）。
- LLM 是个思考助手，而AI Agent是个执行助手。
- 面对LLM，最重要的事情，可能是提问题。而面对AI Agent的时候，最重要的事情，可能是下任务。

通用AI智能体 MANUS

- 2025年3月5日晚间，中国初创公司Butterfly Effect（蝴蝶效应）发布AI Agent（智能体）



文本

- 随着互联网的快速发展，基于自然语言的网页数量已达到惊人的万亿级别。
- 这些资源成为人们获取信息、交流思想、展开社交活动的重要平台。
- 随着文本数据规模的迅速膨胀，人们面临着如何有效利用如此大批量数据的挑战。
- 人工处理如此海量的文本资源显然不切实际，因此，文本信息处理技术应运而生，赋予了机器理解、分析和生成人类语言的能力，成为连接人类智慧与机器智能的纽带。

自然语言

- 语言作为人类智慧的基石，不仅传递逻辑和知识，还承载着交流和思维的核心使命。
- 文本和语音是语言的两种重要表达方式，在人类文明的进程中发挥着关键作用。
  - 文本以书面形式展现语言，通过文字和符号便于阅读理解
  - 语音则以口头形式传达信息，依赖听觉理解。
  - 相比之下，文本更适合作为信息的记录载体，因为它需要更加结构化的组织，而语音则包含语气、语调、重音等额外信息。
- 目前，自然语言是最主流的文本信息之一，自然语言是人类日常交流使用的语言，例如英语、汉语、西班牙语等，通过文字、语音等形式表达信息。
- 在文本形式中，自然语言包括书面语言，如书籍、文章、邮件、社交媒体帖子等。这些文本信息可以包含丰富的语义和情感，是自然语言处理技术的主要处理对象。

文本信息处理/文本智能计算

- 文本信息处理致力于将自然语言文本转换为计算机可理解和处理的形式，包括文本分类、情感分析、命名实体识别、机器翻译等任务。
- 其目标是从文本中提取有用信息，为后续分析和应用奠定基础。
- 文本处理涵盖两个重要的子领域：文本理解和文本生成。
  - 文本理解指计算机通过分析和解释文本输入，提取其中的意义和信息，包括语义理解、语法分析、语境理解等任务。其目标是使计算机准确理解和解释人类语言，以便进行推理、问题回答、任务执行等操作。
  - 文本生成则是计算机根据特定输入数据利用语言模型和规则生成符合语法和语义的文本，包括生成报告、摘要、回答问题、对话等任务。其目标是让计算机以自然的方式生成可理解的文本，实现与用户的自然交互和沟通。

文本信息处理步骤

- 文本信息处理通过文本预处理、特征提取、任务分类、模型训练与优化、模型评估与选择以及应用部署的一系列步骤，实现了对文本数据的高效处理、理解和生成。

- 这些步骤共同构成了现代文本信息分析与应用的基本流程，为各种文本相关任务提供了可靠的技术支持和解决方案。

预处理-文本分词 (Tokenization)

- 文本信息处理中的文本分词 (Tokenization) 是至关重要的预处理步骤，在文本处理中扮演着不可或缺的角色，为文本结构理解和语言处理奠定基础。
- 分词将连续字符序列划分为有意义的词语，有助于计算机更准确地理解文本的意义和语法结构。
- 分词的目标是将较大的文本单元（如句子或文章）切分成更小的单元，这些单元称为“tokens”，通常是单词、短语或其他有意义的元素，作为语言处理的基本单元用于进一步分析和处理。

分词过程

- 分词过程与具体任务密切相关，例如：
  - 处理英语文本时，通常按空格和标点符号切分，因为英语词汇以空格分隔；
  - 处理中文、日语等语言时，分词变得更加复杂，因为这些语言不使用空格分隔词汇，需要识别连续汉字之间的词汇边界，通常依赖词典、统计模型或深度学习模型来实现。
  - 随着分词技术的不断发展，从最初的基于规则和词典的方法，到统计机器学习方法，再到现代基于深度学习的方法，处理复杂文本的能力不断提升。
  - 在实际应用中，选择合适的分词工具和方法取决于具体语言、任务需求和可用资源。

中文分词Chinese Word Segmentation

- 由于中文文本中的词语是连续书写的，没有明显的分隔符(如空格)来表示词汇边界，因此分词是理解中文文本的第一步。
- 虽然单个字能传达一定的语义信息，但是不明确的词汇边界容易引入歧义，如“中文”和“中心”两词中的“中”字传达了不同的语义信息。
- 中文分词(Chinese Word Segmentation)也是如句法分析、语义理解等下游任务中的重要前置步骤。除此之外，对中文文本分词，可以提高信息检索与数据分析的准确性。例如：句子“今天是晴天。”的分词结果为“今天/是/晴天/。”，其中“/”为词之间的边界
- 目前，中文分词技术主要分为：基于规则的分词、基于统计的分词和基于深度学习的分词。其中基于深度学习的分词通过在大规模语料上的预训练,获得强大的语义理解能力，并在此基础上，通过微调或其他技术，进一步提高分词的准确率。

子词分词Subword Tokenization

- 子词分词(Subword Tokenization)的目的是将单词切分成子词，主要运用在预训练模型中。
- 例如：句子“I am eating.”对应的子词分词结果可为“I/am/eat/#ing/.”，其中“#ing”即为子词。
- 常用的子词分词算法有Byte-Pair Encoding (BPE)算法、WordPiece算法以及SentencePiece算法等，这些算法各有特点，可适用不同场景。

文本预处理1-Tokenization 分词

- 在处理文本前，通常需要对句子/文档分词。
- 在预训练语言模型（如BERT）之前：
  - 中文分词：我/正在/工作/。 （以词为单位）
  - 英文分词：I/am/working/. （以单词为单位）
- 在预训练模型数据量大后（Byte-Pair Encoding, BPE为例）：
  - 中文分词：今/正/在/工/作/。（以字为单位）
  - 英文分词：I/am/work/#ing/. （以子词 Subword为单位）

分词算法

- BPE分词算法可以灵活调整词表的大小，适配不同的语言。同时，由于BPE算法的词表一般包含该语言的最小字符单位，因此通过将单词切分为字词，BPE分词算法能够很好地处理未登录词(Out-of-Vocabulary Words, OOV) 的情况
- WordPiece算法可以通过分词保持较小的词汇表同时覆盖大部分词汇，且其能有效处理稀有词汇和未见词汇，增强模型的鲁棒性
- SentencePiece算法不依赖于特定的语言学特性或分词规则，可以处理任何语言，包括没有明确分词规则的语言（如中文、日文等）

分词算法BPE

BPE分词算法为例介绍子词分词的过程：

(1) 统计字符频率：首先统计文本数据中每个单个字符的出现频率。

(2) 初始化词表：以单个字符作为初始的词表条目。

(3) 迭代合并：在词表中寻找出现频率最高的相邻字符对，将其合并成一个新的字符,并加入到词表中。

(4) 更新频率：在合并字符后,更新各个字符的出现频率。

(5) 重复合并过程：重复步骤 (3) 和 (4)，直到达到预设的词表大小或合并次数上限。

文本预处理2-词性标注（Part-of-Speech Tagging）

- 词性标注是确定文本中每个单词的词性的过程。例如，一个单词可以是名词、动词、形容词等。
- 在预处理阶段进行词性标注有助于后续任务，如句法分析、命名实体识别和信息抽取。
- 通过词性标注，模型可以更好地理解句子的结构，从而提高其在各种NLP任务中的性能。

2.POS 词性/词类

句子是由单词构成，英语单词根据词义及在句子中的作用划分为以下十二类：

动词 Verbs (v.) 表示动作或状态等。例词：work,know.

名词 Nouns(n.) 表示人或事物的名称。例词：boy，flower.

形容词 Adjectives(a./adj.) 用来修饰名词或代词。例词：pretty,useful.

副词 Adverbs(ad./adv.) 用来修饰动词、形容词或副词。例词：slowly，very.

介词 Prepositions（prep.）用在名词、代词等前面，表示与别的词的关系。例词：for，from.

代词 Pronouns (pron.) 用来代替名词或数词等。例词：they,some.

数词 Numeral(num.) 用来表示数量或顺序。

量词 Quantifier（quant.） 通常用来表示人、事物或动作的数量单位。

连词 Conjunction（conj.） 用来连接词与词或句与句。例词：but，if.

疑问词 Interrogative (int.)

感叹词 Interjection (interj.) 表示说话时的感情或口气。例词：oh，ah.

冠词 Article(art.) 用在名词前帮助说明其词义。例词：a，the.

汉语的词主要可以分为两个大类，14小类。

大类即实词和虚词，

名词、动词、形容词、数词、量词、代词、区别词共7小类是实词

副词、介词、连词、助词、叹词、语气词、拟声词等7小类是虚词。

文本预处理3-语义消歧 (Word Sense Disambiguation)

- 语义消歧旨在解决一词多义的问题。在自然语言中，同一个单词可能有多个含义，具体含义取决于其在句子或上下文中的位置。
- 语义消歧的目标是确定给定上下文中单词的具体含义。
- 通过消除语义歧义，模型能够更准确地理解文本的含义，从而提高文本分类、情感分析等任务的准确性。

文本预处理3-Stemming词干提取

- 词干提取（Stemming）是自然语言处理中的一个过程，用于将单词缩减到其基本的词形或词干形式。这个过程通常通过去除单词的词缀（如复数标记、时态标记等）来实现，以便更好地进行文本分析、比较或索引。
- 词干提取的目标是减少词汇的多样性，使不同形态但意义相近的单词能够归并到同一个基本形式。这对于很多NLP任务来说是有益的，因为它能够简化词汇，减少数据稀疏性，并有助于提升模型的泛化能力。

Original Words
…
consign
consigned
consigning
consignment
consist
consisted
consistency
consistent
consistently
consisting
consists
organization
policy
execute
arm

Stemmed Words
…
consign
consign
consign
consign
consist
consist
consist
consist
consist
consist
consist
Not Organ
Not Police
Not Executive
Not Army

文本预处理4- Parsing Tree语法树

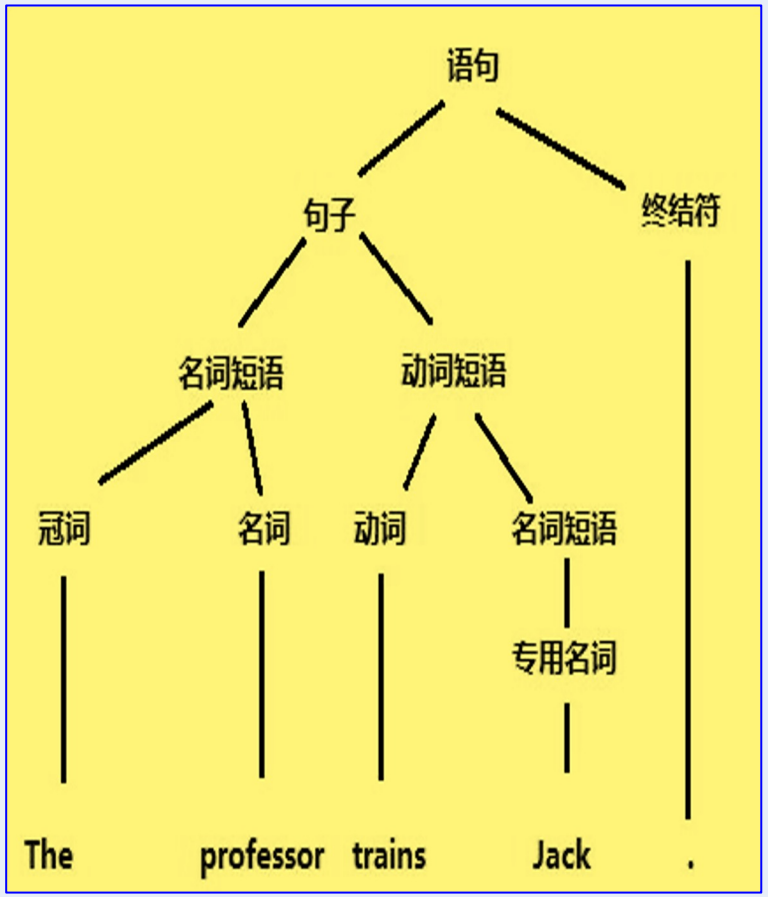
- 语法树（Syntactic Tree），也被称为解析树（Parse Tree），是句子语法结构的直观表示，它反映了句子中词语的句法关系。语法树将句子的组成成分按照它们之间的语法关系组合成树状的结构，以显示词语或词组如何彼此关联并组合在一起形成有意义的句子。

- 在语法树中，每个节点都代表句子中的一个词语或短语，而边则表示这些词语或短语之间的语法关系。例如，主语、谓语、宾语、定语、状语等都会作为节点出现在语法树中，而它们之间的关系（如主谓关系、动宾关系等）则由边来表示。

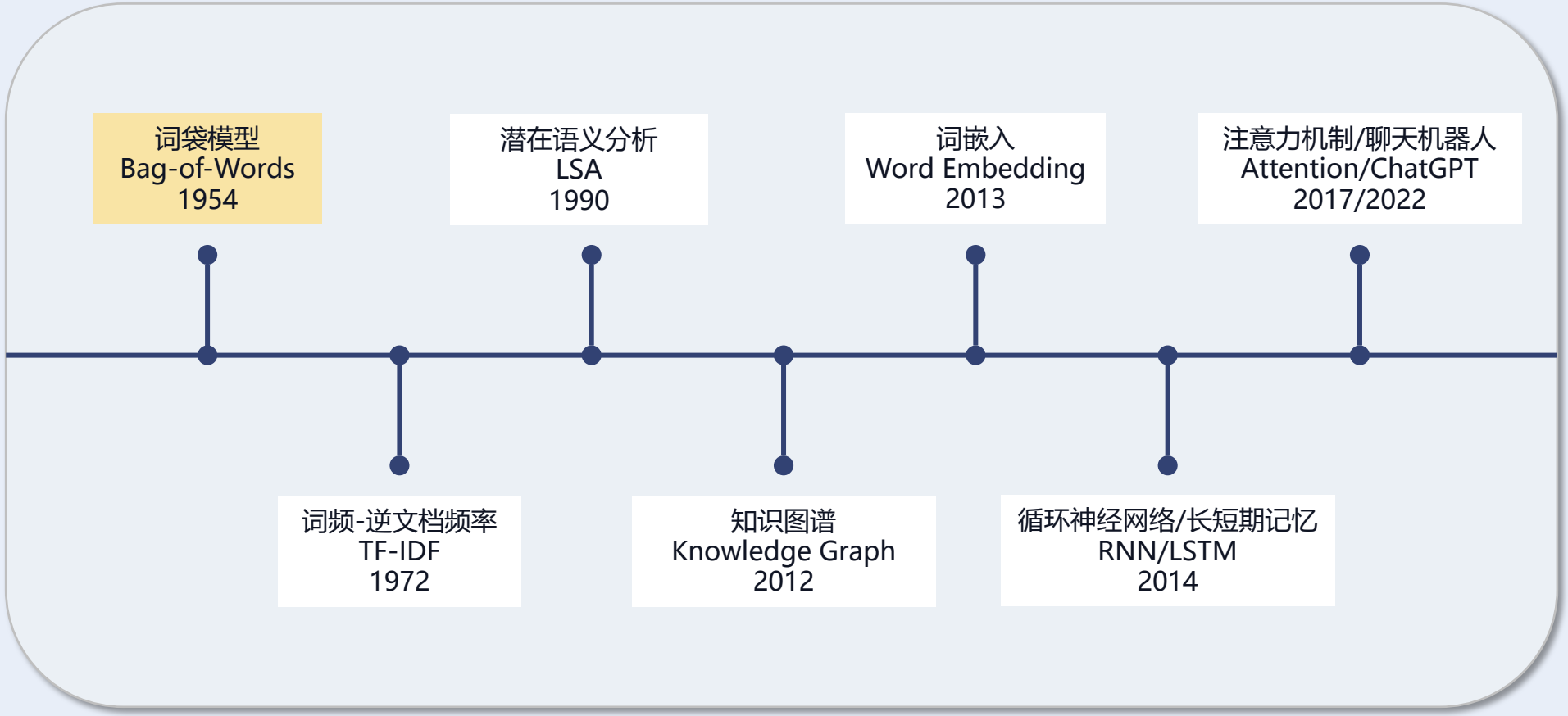
- 语法树的根节点通常代表整个句子，而子节点则代表句子中的各个成分。通过语法树，我们可以清晰地看到句子中各个成分之间的层次关系和依赖关系，这对于理解句子的结构和意义非常有帮助。

文本预处理4- Parsing Tree语法树

- 给定文法G=(Vn,Vt,P,S)，对于G的任何句型都能构造与之关联的语法树(推导树)。
- 树中的每一个节点都有一个标记，此标记是V=VnUVt中的一个符号。语法树是句子结构的图形表示，它代表了句子的推导结果，有利于理解句子语法结构的层次。
- 简单说，语法树就是按照某一规则进行推导时所形成的树
- 语法树包括了一个句型的所有可能的推导过程。



文本处理发展里程碑1


词袋模型
Bag-of-Words
1954
词频-逆文档频率
TF-IDF
1972
潜在语义分析
LSA
1990
知识图谱
Knowledge Graph
2012
词嵌入
Word Embedding
2013
循环神经网络/长短期记忆
RNN/LSTM
2014
注意力机制/聊天机器人
Attention/ChatGPT
2017/2022

里程碑1: 词袋模型 (Bag-of-Words)

- 1950年艾伦·图灵 (Alan Turing) 提出图灵测试 (Turing Test) 作为判断智能的条件，从此拉开了人工智能的序幕。
- 1954年泽里格·哈里斯 (Zellig Harris) 提出词袋 (Bag-of-Words) 的概念。
- 词袋模型常用语自然语言处理 (Natural Language Processing, NLP) 和信息检索 (Information Retrieval, IR) 领域中。除此以外，词袋模型也被用在计算机视觉 (Computer Vision, CV) 领域中。
- 词袋模型以向量的形式表示一个文档，其中每一维表示文档中的一个词。由于词袋模型并不考虑词在文档中的顺序，因此被形象地称为“词袋”。

1：BoW 词袋模型定义

- 词袋模型(Bag-of-Words, BoW)是文本信息智能计算中最基础和常用的算法，广泛应用于自然语言处理和信息检索等领域，在计算机视觉中也有应用
- 词袋模型以向量的形式表示一个文档，其中每一维表示文档中的一个词。该模型不考虑词在文档中的顺序，被形象地称为“词袋”
- 该算法的核心思想是将文档表示为各个词语出现频率的向量，忽略了词语在文档中的顺序和语法结构。词袋模型的优点在于简单高效，易于实现和应用于大规模文本处理

BoW 局限

- BoW存在一些局限性：
  - 首先，词袋模型忽略了词语顺序信息，导致丢失文本的上下文语义关系。
  - 其次，它无法处理单词的多义性和歧义性，可能将相似的词视为不同的特征。
  - 另外，词袋模型也无法捕捉词序对结果的影响，而对于某些任务如情感分析和机器翻译，词序是非常重要的。

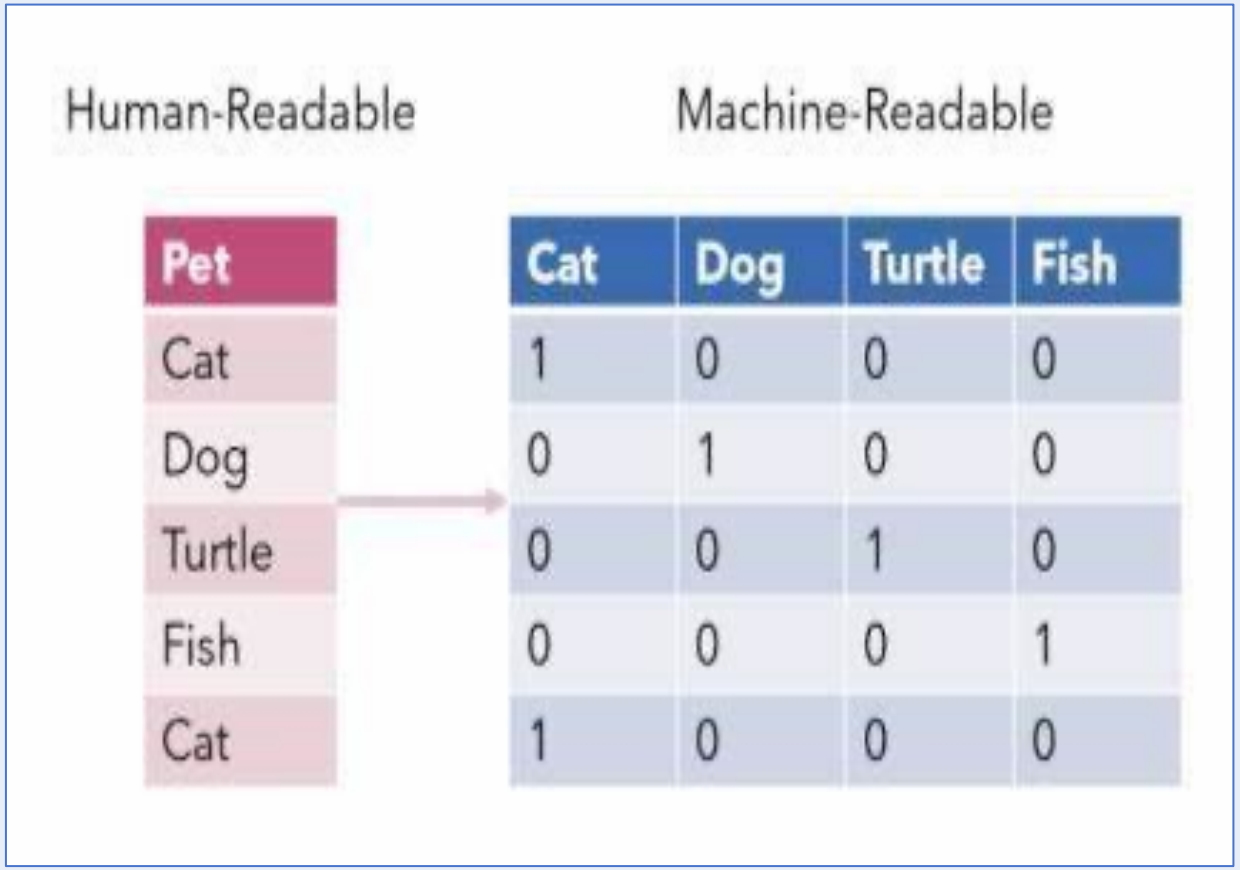
2: BoW 应用

- 词袋模型（Bag of Words, BoW）在文本处理中是一个常见且重要的概念，它假设文本（段落或文档）可以被视为一个无序的词汇集合，其中每个单词的出现都是独立的，不依赖于其他词是否出现。在这种模型下，每个文本可以表示为一个向量，向量的每个维度对应一个词汇，其取值为该词汇在文本中出现的次数或者频率。这种表示方法使得文本数据更易于进行数学运算和机器学习算法的处理。

- 词袋模型在多个NLP领域有广泛的应用，其中包括搜索、文档分类和主题模型等。

BoW 向量表示编码-One-hot 独热编码

- One-hot 编码，将类别变量转换为机器学习算法易于利用的格式的方法。搜索引擎可以快速地计算文档与查询之间的相似度
- One-hot 编码的基本思想是为每个可能的类别创建一个新的二进制特征。对于给定的类别，其对应的特征值为 1，而其他所有特征的值都为 0。这样，每个类别都被表示为一个只有一个 1 的二进制向量。



BoW 文档分类

- 直觉上，如果文档的内容相似，那么这些文档就是相似的。此外，仅仅从内容本身，我们就可以了解文档的含义。

- 词袋模型可以简单也可以复杂，复杂程度取决于：
  - 如何设计已知词汇（或标记）的词汇表，
  - 如何为已知词汇的出现打分

3: BoW 词袋模型构建步骤

- 词袋模型构建包含构建词汇表、编码文本、创建文档向量等步骤。
  - 首先，构建词汇表的目的是为了创建一个包含语料库中所有唯一词汇的列表。
  - 然后，编码文本是将文档中的词映射到词汇表中的词汇，并记录其出现的次数或权重。
  - 最后，创建文档向量则是将每个文档表示为一个向量，向量的维度等于词汇表的大小，而每个元素则对应词汇表中的一个词汇，并表示该词汇在文档中的权重。
- 在文本编码过程中，有时会遇到词汇表中没有收录的词语，称为未登录词汇 (Out of Vocabulary, OOV)。词袋模型往往会先移除这些OOV词语，然后再对文档进行编码。

BoW 词袋模型例子：构建文档分词

- 对于语料库$\mathcal{D} = \{d_1, d_2, d_3\}$，其中每个文档分别为：
  - d_1=“眼睛/是/心灵/的/窗户/。”
  - d_2=“眼睛/是/一种/人体/器官/。
  - d_3=“白内障/是/一种/眼睛/疾病/。”

首先利用分词算法对每个文档分词，然后收集词汇，收集得到的词汇表为$\mathcal{V}$ ={“眼睛”，“是”，“心灵”，“的”，“窗户”，“一种”，“人体”，“器官”，“疾病”，“白内障”，“。”}

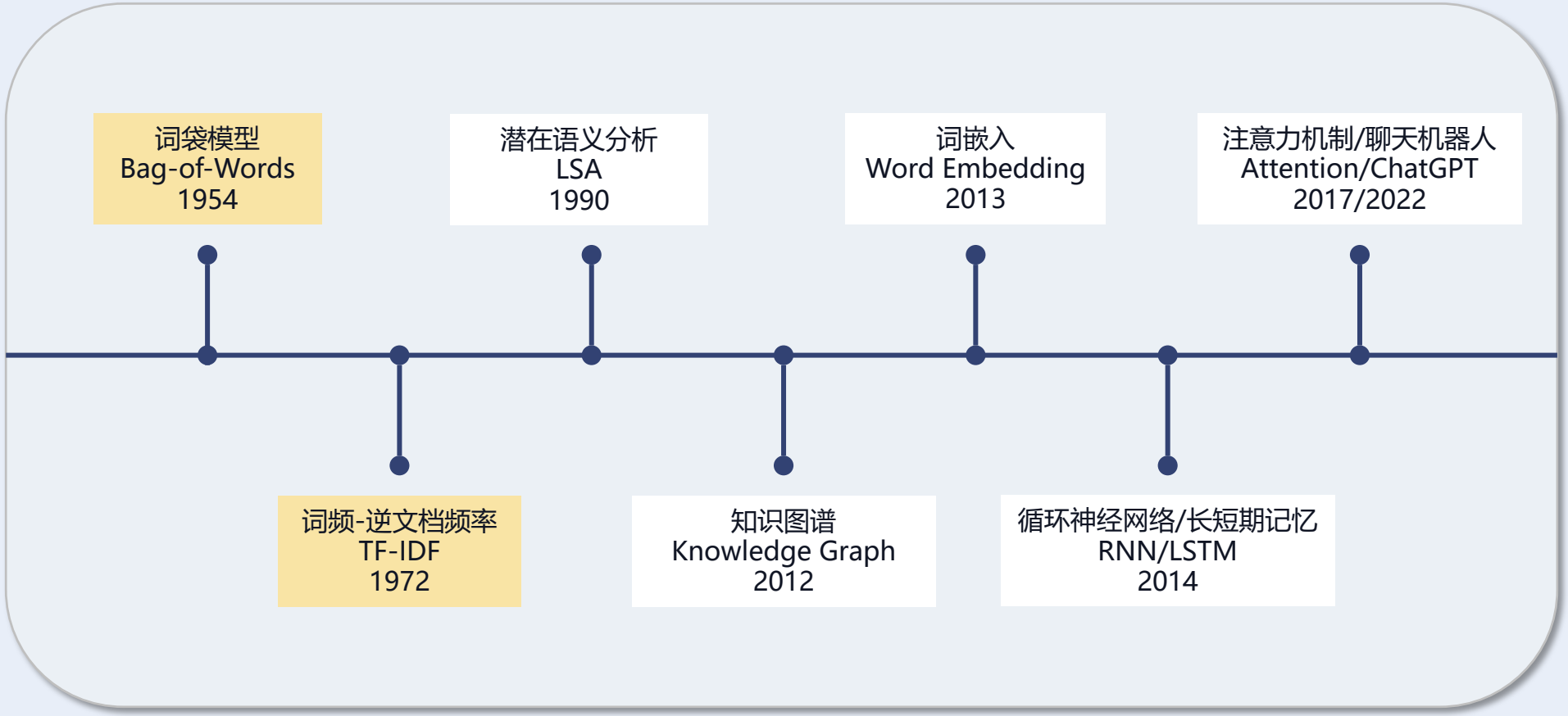
BoW 词袋模型例子：各文档对应的词袋模型向量

	眼睛	是	心灵	的	窗户	一种	人体	器官	疾病	白内障	。
d_1	1	1	1	1	1	0	0	0	0	0	1
d_2	1	1	0	0	0	1	1	1	0	0	1
d_3	1	1	0	0	0	1	0	0	1	1	1

其中1表示单词出现在文档中，而0则表示没有出现。得到文档向量后，我们可以利用余弦相似度(Cosine Similarity)可以计算两个文档之间的相似度，其中文档1与文档2的相似度为：

$\cos(d_1, d_2) = \frac{d_1 \cdot d_2}{\|d_1\| \|d_2\|}$。

文本处理发展里程碑2


词袋模型
Bag-of-Words
1954
词频-逆文档频率
TF-IDF
1972
潜在语义分析
LSA
1990
知识图谱
Knowledge Graph
2012
词嵌入
Word Embedding
2013
循环神经网络/长短期记忆
RNN/LSTM
2014
注意力机制/聊天机器人
Attention/ChatGPT
2017/2022

TF-IDF (词频-逆文档频率)

- 词频-逆文档频率(Term Frequency-Inverse Document Frequency,TF-IDF）是一种衡量文档中词的重要性的文本信息智能计算算法，常用于信息检索、文本挖掘等任务中。
- 词频-逆文档频率这项技术最早可以追溯到1972年，是一种改进的词袋模型，其考虑了每个词在文档以及语料库中的重要性，该算法融合了两种统计特征，即词频 (TF) 和逆文档频率 (IDF)。

词频与逆文档频率

- 其中词频用于衡量一个词在某个文档中出现的频率。一个词在特定文档中出现频率越高，通常意味着它与该文档的主题相关性越强，具有较高的重要性。
- 而逆文档频率衡量一个词在整个语料库中出现的频率。如果一个词出现在大部分文档中,则说明它是一个常用词,相对于整个语料库而言,它的重要性较低。IDF的设计就是为了降低这种常用词的影响。

TF-IDF 计算公式 - TF

- TF-IDF算法将这两个因素相乘,得到一个综合评分。具体来说，一个词语的TF-IDF值等于它的词频乘以它的逆文档频率。这样既考虑了词语在某个文档中的重要性，也考虑了它在整个语料库中的重要性。TF-IDF算法的计算由词频 (TF) 和逆文档频率 (IDF) 两部分组成。对于文档 d 中的词 t，$\mathrm{TF}(t,d)$ 表示该词在文档中的频率.具体计算方法如下：

$$\mathrm{TF}(t,d) = \frac{\mathrm{COUNT}(t,d)}{\sum_{t' \in v} \mathrm{COUNT}(t',d)}$$

其中，COUNT(t,d) 表示文档d中出现词t的次数，分母为文档中的总词数。

TF-IDF 计算公式 - IDF

- 对于语料库$\mathcal{D}$，词t的逆文档频率 IDF(t) 表示为词占全体文档比例的对数反比关系,其中，$|\mathcal{D}|$ 表示语料库中的文档数量，分母则为包含词 t 的文档数。具体计算方法如下：

  - $$\mathbf{IDF}(t) = \log \frac{|\mathcal{D}|}{\sum_{d \in \mathcal{D}} 1 \cdot [t \in \mathcal{D}]}$$

TF-IDF

- 词 t 在文档d中的 TF-IDF 表示为词频 TF(t,d) 与逆文档频率 IDF(t) 的乘积, 具体计算方法如下:

$$\mathbf{TF-IDF}(t,d) = \mathbf{TF}(t,d) \cdot \mathbf{IDF}(t)$$

- TF-IDF算法将文档 d 表示为 $|\mathcal{V}|$ 维的向量，其中每一个维度表示一个词t，其数值的意义为词 t 在文档 d 中的重要性。TF-IDF算法对文档中高频出现，但是在语料库中出现频率较低的词赋与了较高的重要性，因此可以筛选文档中的高频词，并过滤掉其中的常见词，保留反映主要内容的关键词。

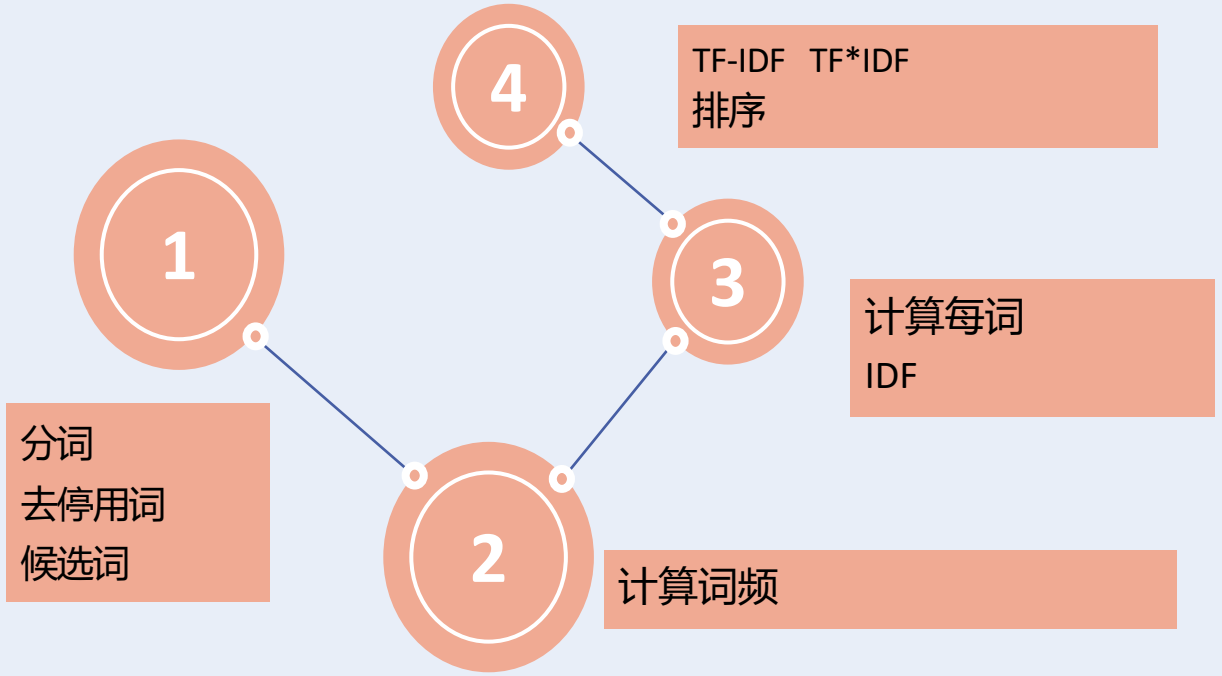
TF-IDF 的可能实现方式

$$词频(TF) = \frac{某个词在文章中的出现次数}{文章的总词数}$$

$$词频(TF) = \frac{某个词在文章中的出现次数}{该文出现次数最多的词的出现次数}$$

$$逆文档频率(IDF) = \log(\frac{语料库的文档总数}{包含该词的文档数+1})$$

$$TF-IDF = 词频(TF) \times 逆文档频率(IDF)$$



作业 05 (PPT)

1

1. Suppose we have an English text document containing 4375 words. In the document, the words dog and cat appear 115 times and 10 times, please compute TF (term frequency) for dog and cat respectively?
2. Assume we have one hundred thousand English text documents, and dog and cat appear 300 and 26500 of these documents. please compute IDF (inverse document frequency) for terms: dog and cat, respectively?
3. please compute term frequency–inverse document frequency (TF-IDF) of terms: dog and cat based on Question 1 and Question 2?

CS 330 MIP – Lecture 05

文本信息处理

Text Information Processing

Jimmy Liu 刘江

2025-03-19