大模型开发基础：OpenAI Embedding

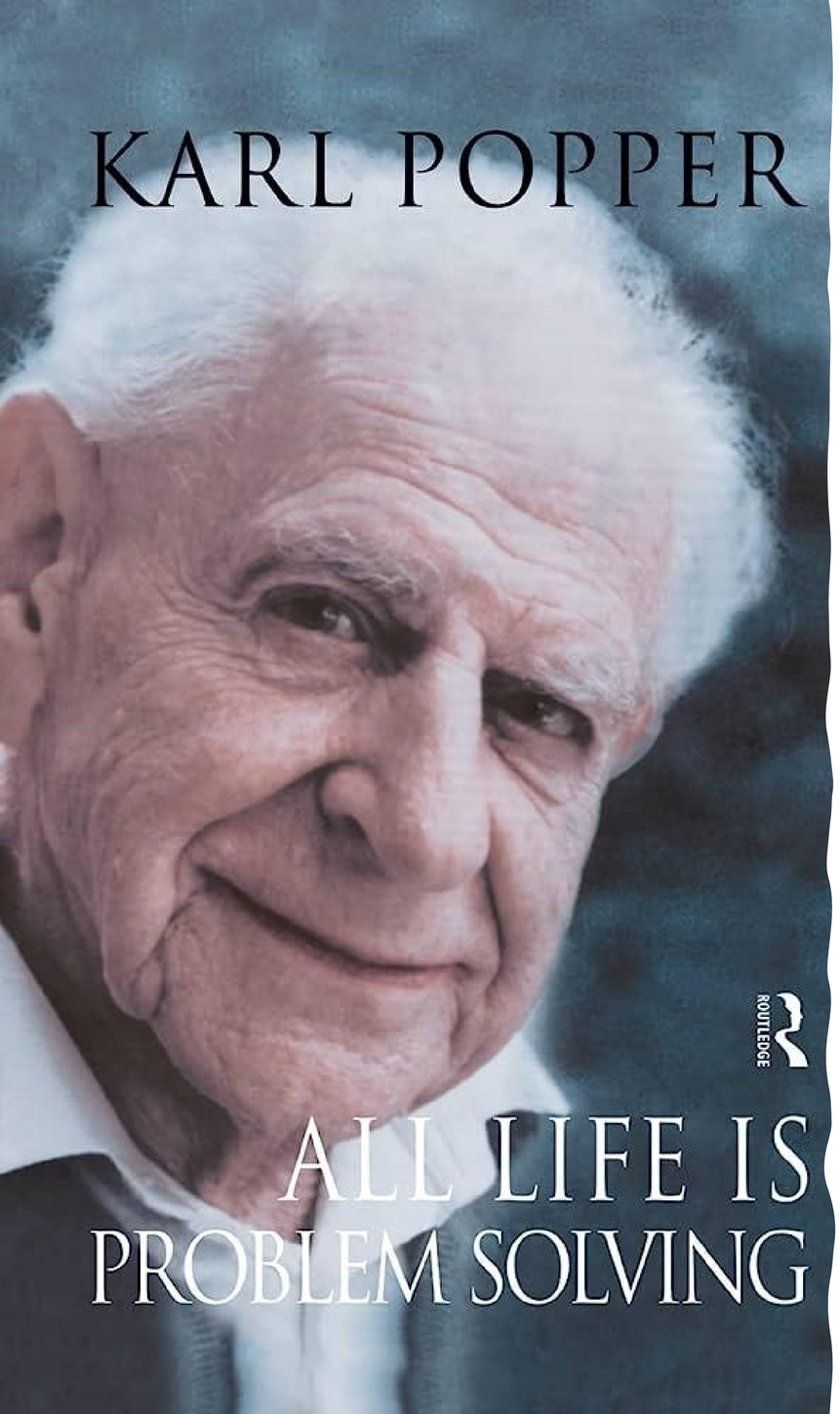
彭靖田

Google Developer Experts
LangChain Developer

目录

- 通用人工智能的前夜
  - “三个世界”与“图灵测试”
  - 计算机的数据表示
  - 表示学习与嵌入
- Embeddings Dev 101
  - 课程项目：GitHub openai-quickstart
  - 快速上手 OpenAI Embeddings
- 快速上手 OpenAI GPT
- 大模型应用最佳实践
- 大模型核心硬件选型指南

通用人工智能的前夜



波普尔的三个世界

- **1978**年， 奥地利哲学家**卡尔·波普尔**提出一种观察、定义现实的方法：
  - 世界一：由物理客体和事件组成的世界， 比如物质、能量和一切有机和无机物质，也就是哲学中的**客观世界**；
  - 世界二：由心灵主体和其感知事件组成的世界。 包括感觉、意识等心理状态和过程，也就是哲学中的**主观世界**；
  - 世界三：**客观知识组成的世界**。 它的内容是思想，思想的内容可以被物质化，成为人造产品和文化产品，比如语言、艺术品、图书、手机、电脑等等，一切含有知识、思想的有形的物质。当然也包括可以用语言表达出来的，人的意识的固定对象，比如问题、猜想、理论、反驳等等

笛卡尔
心物问题

康德
对客观知识的追求

图灵测试(Turing Test)

- 1950年， 英国计算机科学家艾伦·图灵发表了一篇划时代的论文，文中预言了创造出具有真正智能的机器的可能性。
- 由于注意到“智慧”这一概念难以确切定义，他提出了著名的图灵测试：
- 如果一台机器能够与人类展开对话（透过电传设备）而不被辨别出其机器身份，那么称这台机器具有智慧。这一简化使得图灵能够令人信服地说明“思考的机器”是可能的。论文中还回答了对这一假说的各种常见质疑。图灵测试是人工智能哲学方面首个严肃的提案。

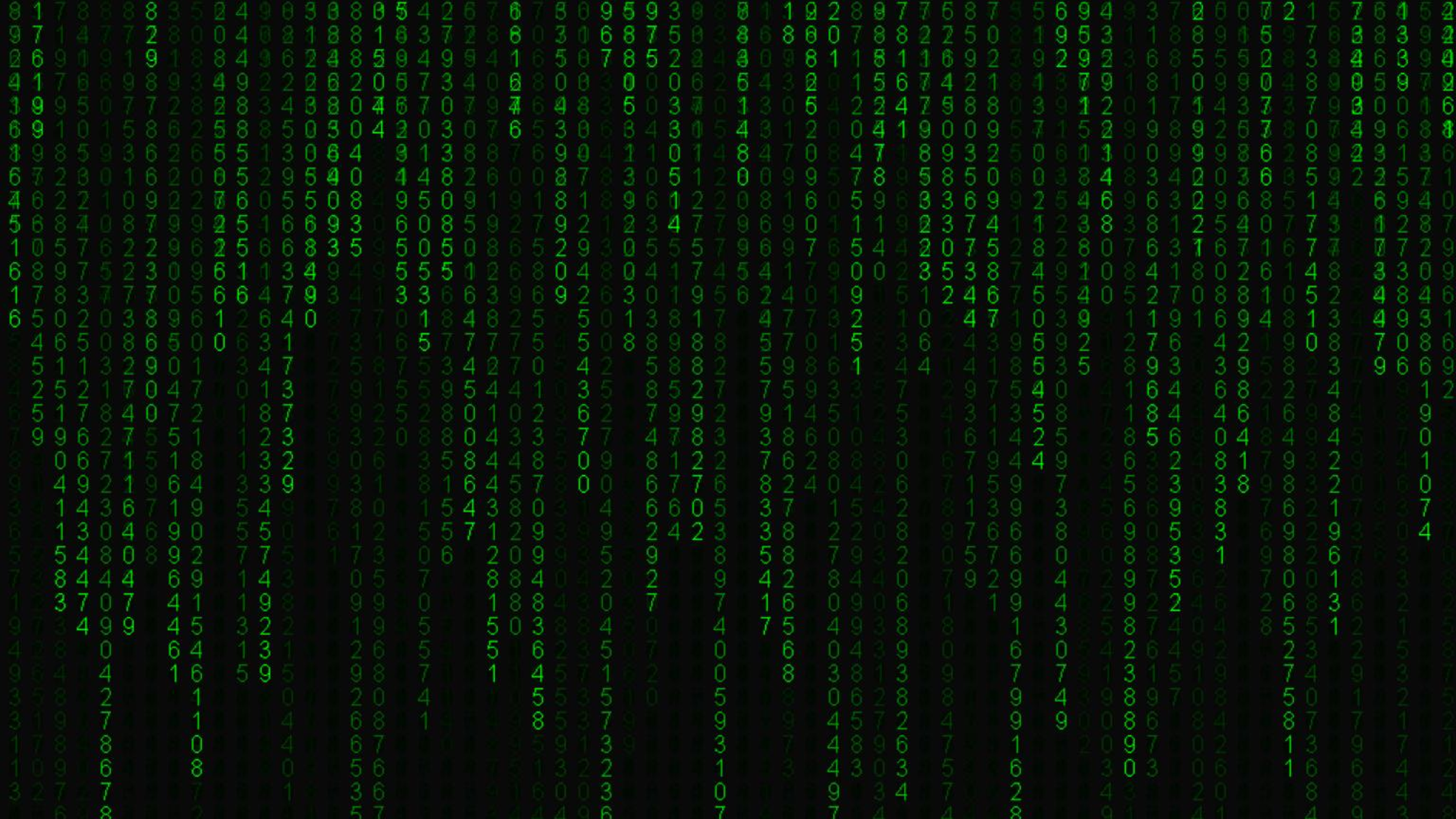
“We can only see a short distance ahead, but we can see plenty there that needs to be done.”

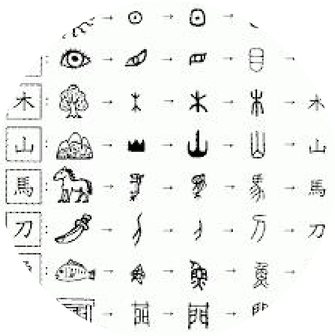
Alan Turing, 1950

人类的客观知识世界

极客时间 训练营

计算机如何表示客观知识的世界？



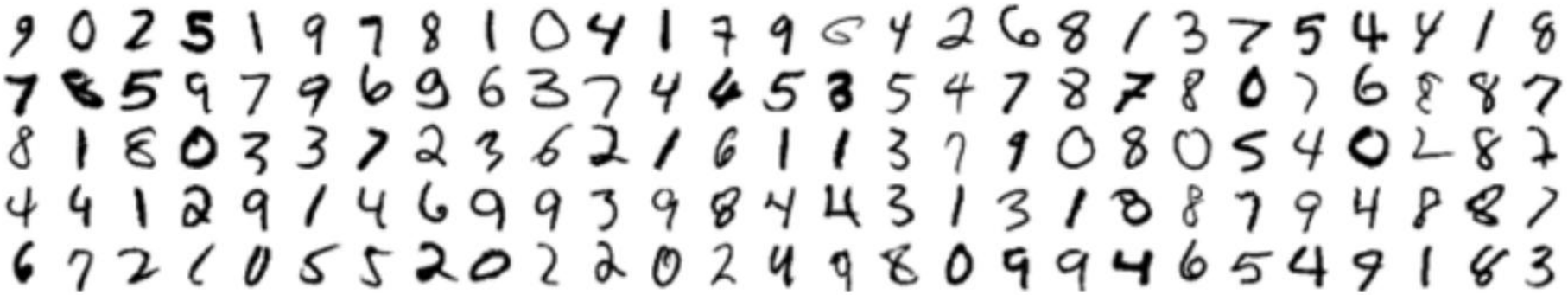


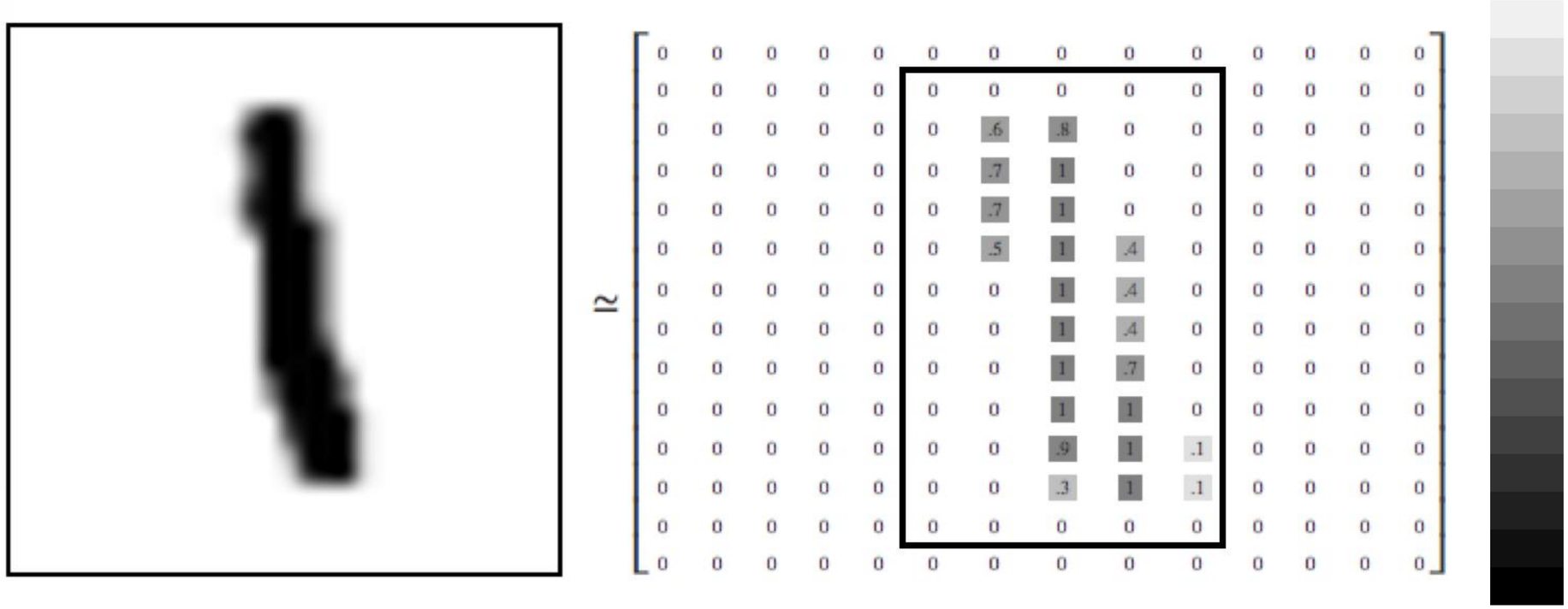
计算机

0	0	0
0	1	0
0	0	0

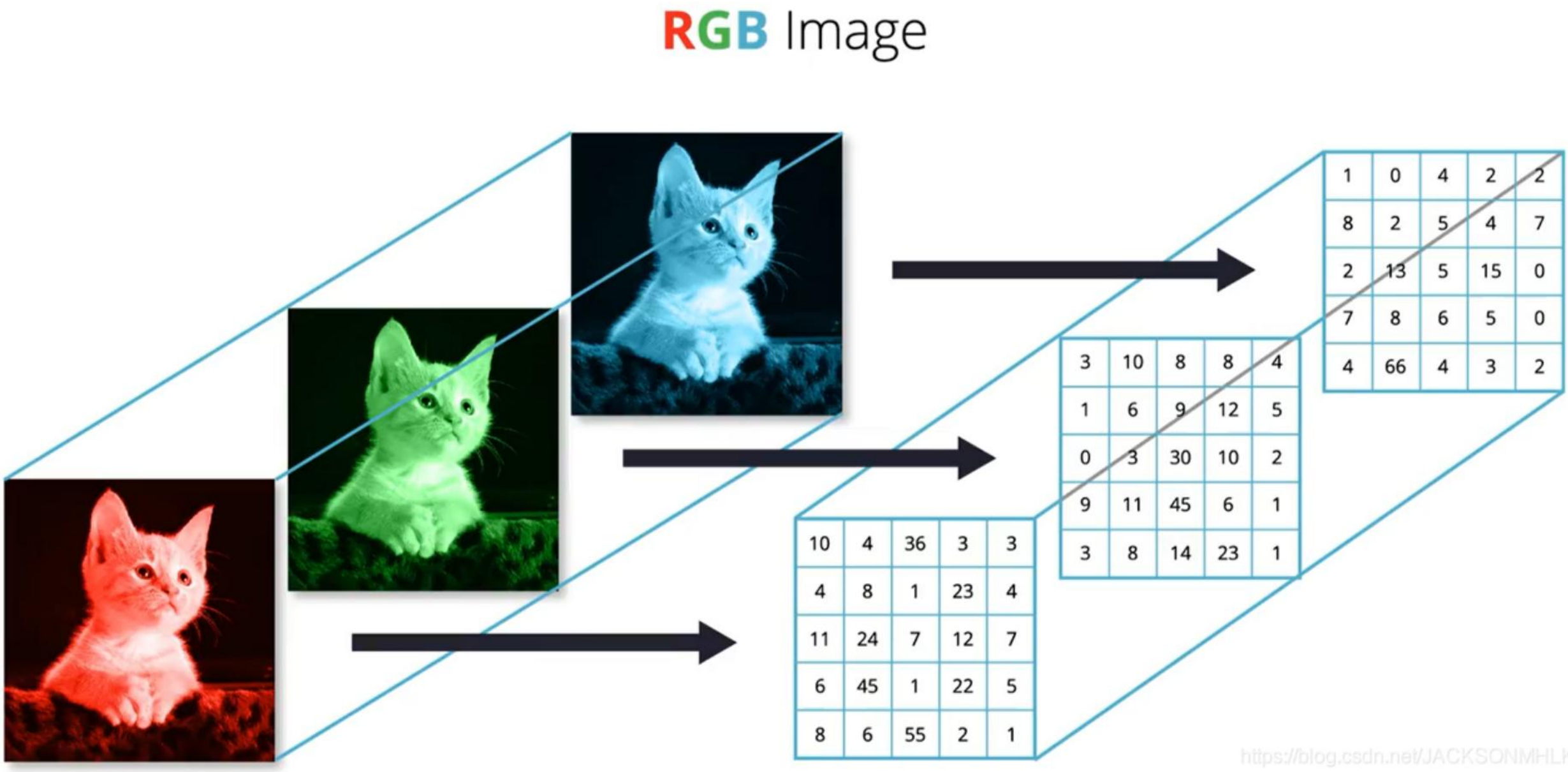
计算机的数据表示

灰度图像表示




≈
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 .6 .8 0 0 0 0 0 0
0 0 0 0 0 0 .7 1 0 0 0 0 0 0
0 0 0 0 0 0 .7 1 0 0 0 0 0 0
0 0 0 0 0 0 .5 1 .4 0 0 0 0 0
0 0 0 0 0 0 0 1 .4 0 0 0 0 0
0 0 0 0 0 0 0 1 .4 0 0 0 0 0
0 0 0 0 0 0 0 1 .7 0 0 0 0 0
0 0 0 0 0 0 0 1 1 0 0 0 0 0
0 0 0 0 0 0 0 .9 1 .1 0 0 0 0
0 0 0 0 0 0 0 .3 1 .1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0

彩色图像表示



https://blog.csdn.net/JACKSONMHLK

西文表示

- **ASCII** (American Standard Code for Information Interchange, 美国信息交换标准代码)是**IEEE** （美国电气和电子工程师协会）**里程碑**之一。1963年发布第一版，最后一次更新在1986年。
- ASCII是基于拉丁字母的一套电脑编码系统。它主要用于显示现代英语，而其扩展版本延伸美国标准信息交换码则可以部分支持其他西欧语言，并等同于国际标准ISO/IEC 646。

ASCII可显示字符（共95个）

二进制	十进制	十六进制	图形	二进制	十进制	十六进制	图形	二进制	十进制	十六进制	图形
0010 0000	32	20	(space)	0100 0000	64	40	@	0110 0000	96	60	`
0010 0001	33	21	!	0100 0001	65	41	A	0110 0001	97	61	a
0010 0010	34	22	"	0100 0010	66	42	B	0110 0010	98	62	b
0010 0011	35	23	#	0100 0011	67	43	C	0110 0011	99	63	c
0010 0100	36	24	$	0100 0100	68	44	D	0110 0100	100	64	d
0010 0101	37	25	%	0100 0101	69	45	E	0110 0101	101	65	e
0010 0110	38	26	&	0100 0110	70	46	F	0110 0110	102	66	f
0010 0111	39	27	'	0100 0111	71	47	G	0110 0111	103	67	g
0010 1000	40	28	(	0100 1000	72	48	H	0110 1000	104	68	h
0010 1001	41	29	)	0100 1001	73	49	I	0110 1001	105	69	i
0010 1010	42	2A	*	0100 1010	74	4A	J	0110 1010	106	6A	j
0010 1011	43	2B	+	0100 1011	75	4B	K	0110 1011	107	6B	k
0010 1100	44	2C	,	0100 1100	76	4C	L	0110 1100	108	6C	l
0010 1101	45	2D	–	0100 1101	77	4D	M	0110 1101	109	6D	m
0010 1110	46	2E	.	0100 1110	78	4E	N	0110 1110	110	6E	n
0010 1111	47	2F	/	0100 1111	79	4F	O	0110 1111	111	6F	o
0011 0000	48	30	0	0101 0000	80	50	P	0111 0000	112	70	p
0011 0001	49	31	1	0101 0001	81	51	Q	0111 0001	113	71	q
0011 0010	50	32	2	0101 0010	82	52	R	0111 0010	114	72	r

ASCII控制字符（共33个）

二进制	十进制	十六进制	缩写	Unicode 表示法	脱出字符 表示法	名称 / 意义
0000 0000	0	00	NUL	␀	^@	空字符（Null）
0000 0001	1	01	SOH	␁	^A	标题开始
0000 0010	2	02	STX	␂	^B	本文开始
0000 0011	3	03	ETX	␃	^C	本文结束
0000 0100	4	04	EOT	␄	^D	传输结束
0000 0101	5	05	ENQ	␅	^E	请求
0000 0110	6	06	ACK	␆	^F	确认回应
0000 0111	7	07	BEL	␇	^G	响铃
0000 1000	8	08	BS	␈	^H	退格
0000 1001	9	09	HT	␉	^I	水平定位符号
0000 1010	10	0A	LF	␊	^J	换行键
0000 1011	11	0B	VT	␋	^K	垂直定位符号
0000 1100	12	0C	FF	␌	^L	换页键
0000 1101	13	0D	CR	␍	^M	CR (字符)
0000 1110	14	0E	SO	␎	^N	取消变换（Shift out）
0000 1111	15	0F	SI	␏	^O	启用变换（Shift in）
0001 0000	16	10	DLE	␐	^P	跳出数据通讯
0001 0001	17	11	DC1	␑	^Q	设备控制一（XON 激活软件速度控制）
0001 0010	18	12	DC2	␒	^R	设备控制二
0001 0011	19	13	DC3	␓	^S	设备控制三（XOFF 停用软件速度控制）
0001 0100	20	14	DC4	␔	^T	设备控制四

中文表示

1. **ASCII扩展：** 在早期的计算机系统中，大多采用ASCII码来表示字符。ASCII只能表示128个字符，主要用于表示英文字符。为了表示汉字，各地开始开发基于ASCII的扩展，如GB2312（中国大陆）、Big5（台湾）、EUC-JP（日本）等。这些编码将ASCII扩展到了256个或更多的字符，使得它们可以表示汉字或者其他非拉丁字母的字符。但这些编码往往只能表示一种语言，且在多种语言混合使用时会出现问题。

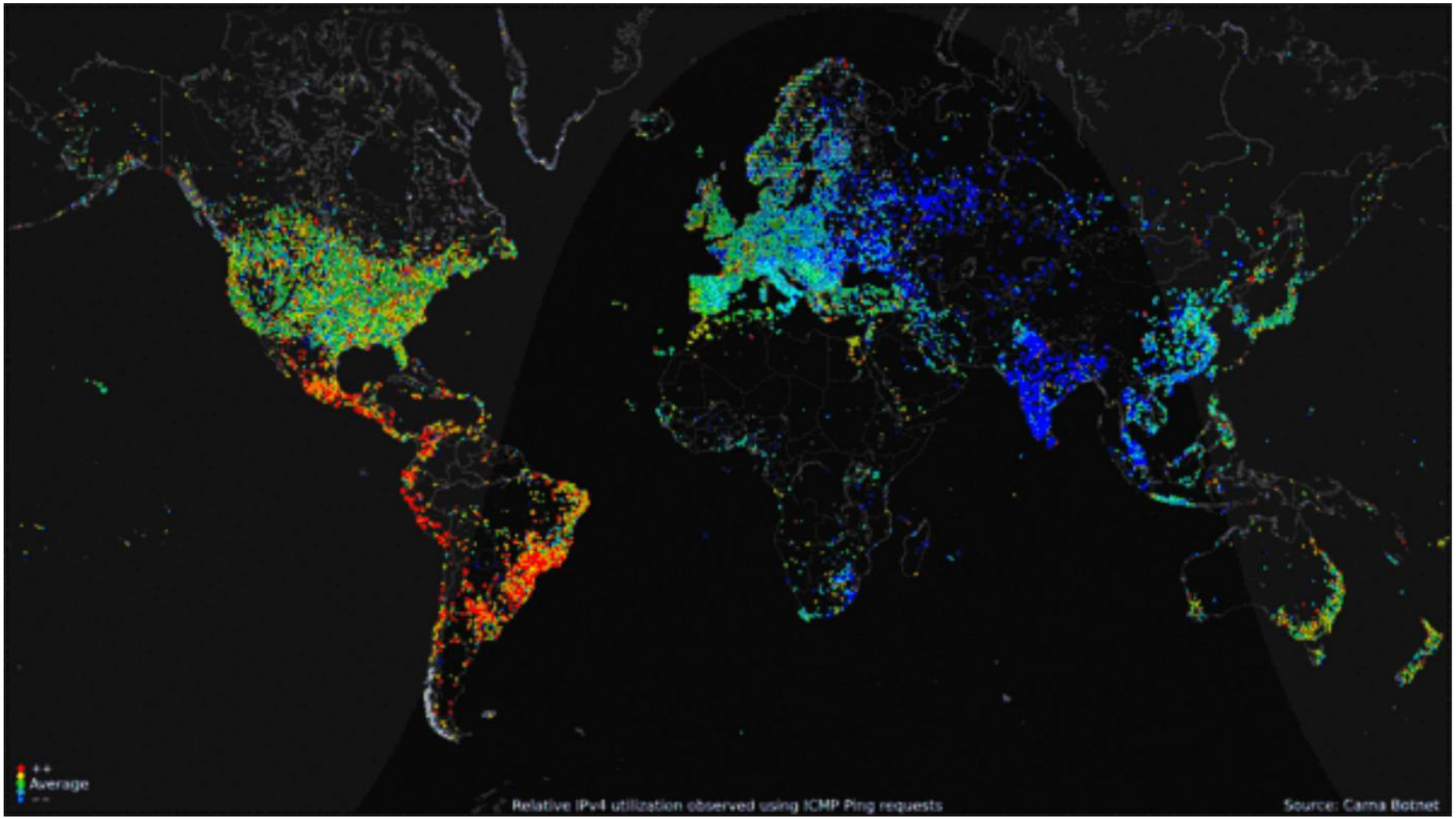
2. **Unicode和UTF-8：** 为了解决多种语言混合使用时的问题，Unicode应运而生。Unicode是一个包含全世界所有字符的编码系统，它将每个字符都映射到一个唯一的数值。UTF-8是Unicode的一种实现方式，它是一种可变长度的编码方式，可以使用1到4个字节来表示一个字符。

3. **One-hot编码（词汇表）：** 在机器学习和深度学习的应用中，文本数据通常需要转换为数值型数据。One-hot编码是一种常用的转换方式。在One-hot编码中，每个汉字都被表示为一个只有一个元素为1，其余元素全为0的向量。向量的长度等于汉字的总数量，1的位置表示该汉字的索引。One-hot编码的一个主要优点是它简单直观，但当汉字的数量非常大时，One-hot编码会占用大量的内存。

4. **基于深度学习的语言模型：** 在最近的发展中，一种基于深度学习的自然语言处理技术被用于表示文字，包括汉字。**这种技术将每个字或词映射到一个高维向量（称为词嵌入），这个向量可以捕捉到字或词的语义信息。**

表示学习与嵌入

World map of 24-hour relative average utilization of IPv4 addresses



Observed using ICMP ping requests as part of the Internet Census of 2012 (Carna Botnet), June – October 2012.[26] Key: from red (high), to yellow, green (average), light blue, and dark blue (low).

Wikipedia contributors (2023) “Global Internet usage,” *Wikipedia* [Preprint]. Available at: https://en.wikipedia.org/wiki/Global_Internet_usage.

APR 2023

OVERVIEW OF INTERNET USE

ESSENTIAL INDICATORS OF INTERNET ADOPTION AND USE

GLOBAL OVERVIEW

INDIVIDUALS USING THE INTERNET	INDIVIDUALS USING THE INTERNET AS A PERCENTAGE OF TOTAL POPULATION	YEAR-ON-YEAR CHANGE IN THE NUMBER OF INDIVIDUALS USING THE INTERNET	PERCENTAGE OF THE TOTAL FEMALE POPULATION THAT USES THE INTERNET	PERCENTAGE OF THE TOTAL MALE POPULATION THAT USES THE INTERNET
5.18 BILLION	64.6% YOY: +2.0% (+129 BPS)	+2.9% +147 MILLION	61.8%	67.4%

AVERAGE DAILY TIME SPENT USING THE INTERNET BY EACH INTERNET USER	PERCENTAGE OF USERS ACCESSING THE INTERNET VIA MOBILE DEVICES	PERCENTAGE OF USERS ACCESSING THE INTERNET VIA COMPUTERS AND TABLETS	PERCENTAGE OF THE TOTAL URBAN POPULATION THAT USES THE INTERNET	PERCENTAGE OF THE TOTAL RURAL POPULATION THAT USES THE INTERNET
6H 35M YOY: -4.4% (-18 MINS)	95.0% YOY: +2.8% (+260 BPS)	62.6% YOY: -7.9% (-540 BPS)	78.5%	45.9%

SOURCES: KEPIOS ANALYSIS; ITU; GSMA INTELLIGENCE; EUROSTAT; WORLD BANK; GOOGLE'S ADVERTISING RESOURCES; CIA WORLD FACTBOOK; CNNIC; NIELSEN; LOCAL GOVERNMENT AUTHORITIES; UNITED NATIONS. TIME SPENT AND MOBILE SHARE DATA FROM GWI (Q4 2022). SEE GWI.COM FOR MORE DETAILS. **NOTES:** GENDER DATA ARE ONLY AVAILABLE FOR "FEMALE" AND "MALE". PERCENTAGE CHANGE FIGURES IN THE BOTTOM ROWS OF DATA SHOW RELATIVE YEAR-ON-YEAR CHANGE. "BPS" FIGURES REPRESENT BASIS POINTS, AND SHOW ABSOLUTE YEAR-ON-YEAR CHANGE. **COMPARABILITY:** SOURCE AND BASE CHANGES. ALL FIGURES USE THE LATEST AVAILABLE DATA, BUT SOME SOURCE DATA MAY NOT HAVE BEEN UPDATED IN THE PAST YEAR. SEE NOTES ON DATA FOR DETAILS.

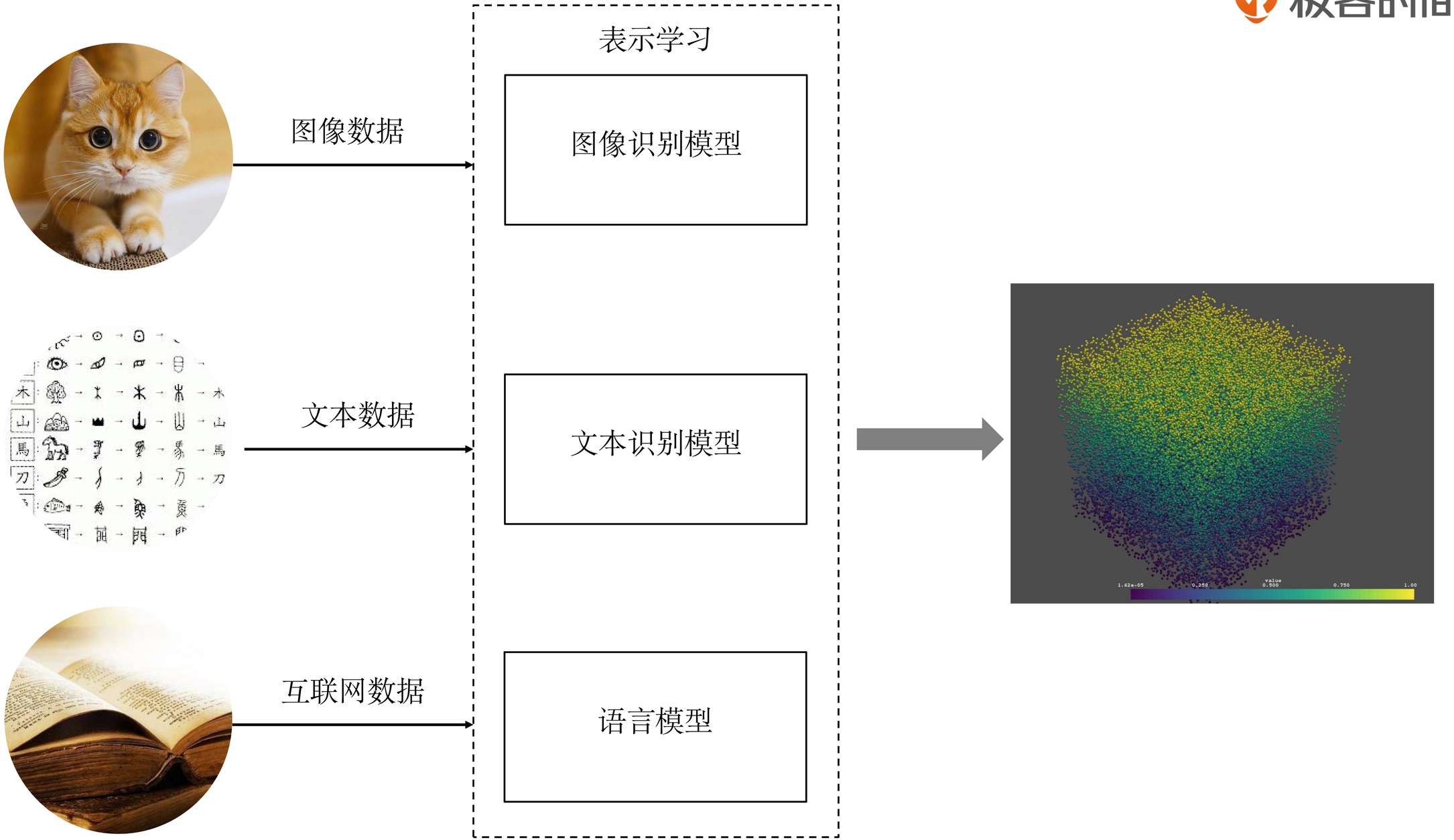
表示学习 vs 嵌入

表示学习（Representation Learning）和嵌入（Embedding）是密切相关的概念，它们可以被视为在不同领域中对同一概念的不同命名或描述。

- **表示学习**：通过学习算法**自动地**从原始数据中学习到一种**表示形式或特征表示**，该表示形式能够更好地表达数据的重要特征和结构。表示学习的目标是将输入数据转换为具有良好表示能力的特征空间，使得在该空间中的数据具有更好的可分性、可解释性或推理能力。

- **嵌入**：表示学习的一种形式，通常用于将**高维数据映射到低维空间中**的表示形式。嵌入可以是词嵌入、图像嵌入、图嵌入等。例如，在自然语言处理中，词嵌入将词语映射到低维向量空间，以捕捉词语之间的语义和句法关系。在图像处理中，图像嵌入将图像映射到低维向量空间，以表示图像的视觉特征。

因此，嵌入可以被视为一种表示学习的特定形式，旨在将高维数据转换为低维向量表示。表示学习可以涉及更广泛的概念和方法，包括嵌入在内，以实现对数据的更好理解和表达。


表示学习
图像数据
图像识别模型
文本数据
文本识别模型
互联网数据
语言模型

极客时间 训练营

表示学习怎么学?

- 向量数据
- 高维空间
- 特征学习
- 语义语法
- 上下文信息
- 大语言模型
- 非结构化数据

表示学习：让计算机更好地理解世界

极客时间 训练营

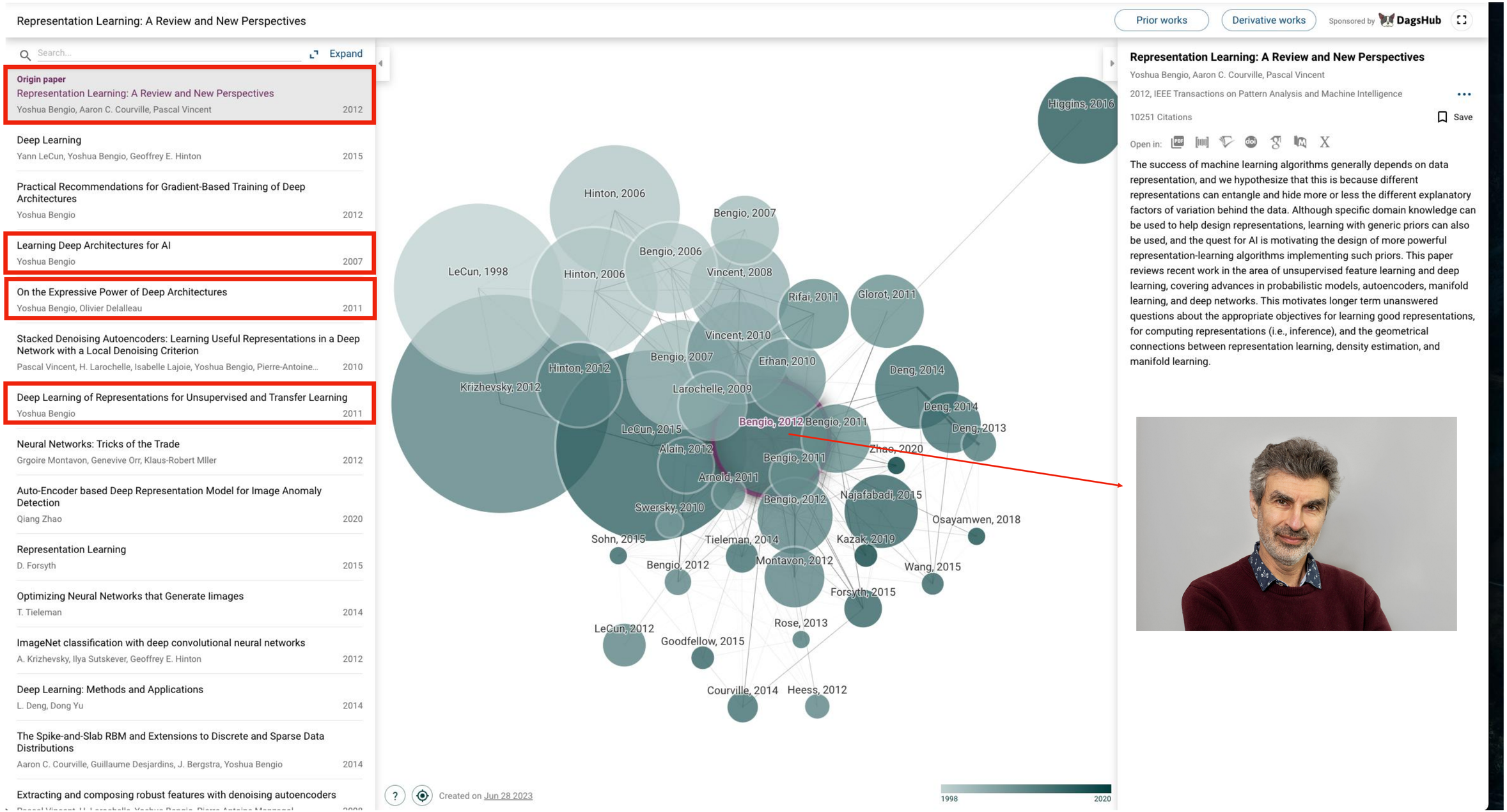
表示学习（Representation Learning）是指通过学习算法**自动地**从原始数据中学习到一种表示形式或特征表示，该表示形式能够更好地表达数据的重要特征和结构。表示学习的目标是将输入数据转换为具有良好表示能力的**特征空间**，使得在该空间中的数据具有更好的可分性、可解释性或推理能力。

- **可分性**：表示学习的目标之一是将输入数据转换为具有更好可分性的特征空间。这意味着在学习到的表示中，**不同类别或不同概念之间的样本应该有明显的边界或区别**。例如，在图像分类任务中，表示学习的目标是使来自不同类别的图像在特征空间中更容易区分。这样的特征表示使得机器学习算法可以更轻松地进行分类、聚类或其他数据分析任务。

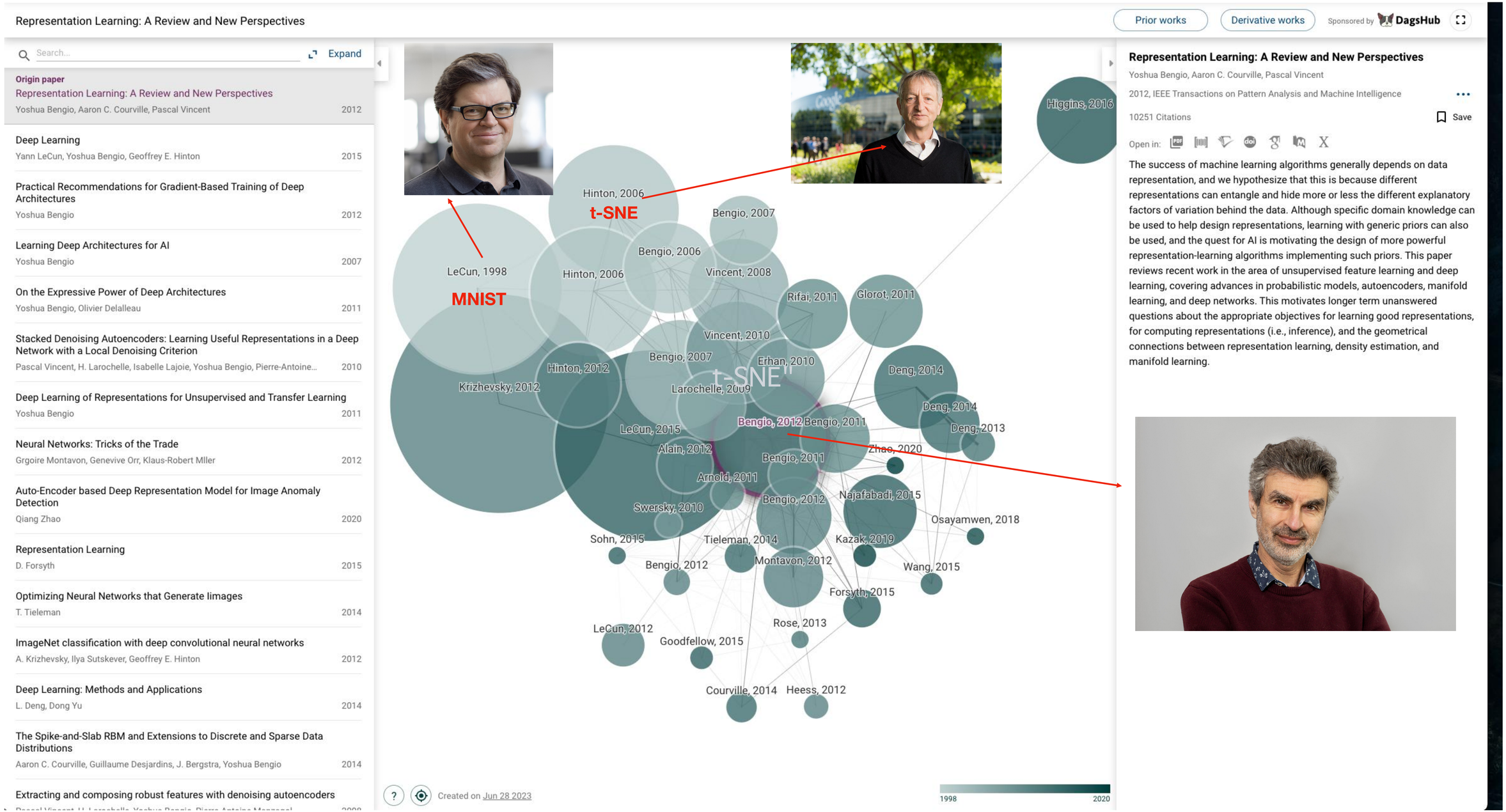
- **可解释性**：另一个表示学习的目标是生成可解释性的特征表示。这意味着学习到的特征应该具有对应于原始数据中的可理解概念或语义的含义。例如，在自然语言处理中，词嵌入模型学习到的词向量应该能够捕捉到**词语之间的语义关系**，使得相似的词在向量空间中更接近。这样的表示不仅有助于模型的解释性，也可以在语义分析和文本生成等任务中提供更好的性能。

- **推理能力**：另一个重要的目标是使得学习到的特征表示在推理任务中更具能力。这意味着在特征空间中，我们可以执行类似于**推理、类比或关联**的操作。例如，通过在词嵌入空间中执行向量运算，如 "国王" - "男人" + "女人"，我们可以得到与 "皇后" 非常接近的结果。这种推理能力使得我们能够进行关联和类比推理，从而在自然语言处理、推荐系统和知识图谱等领域中实现更高级的语义推理和理解。

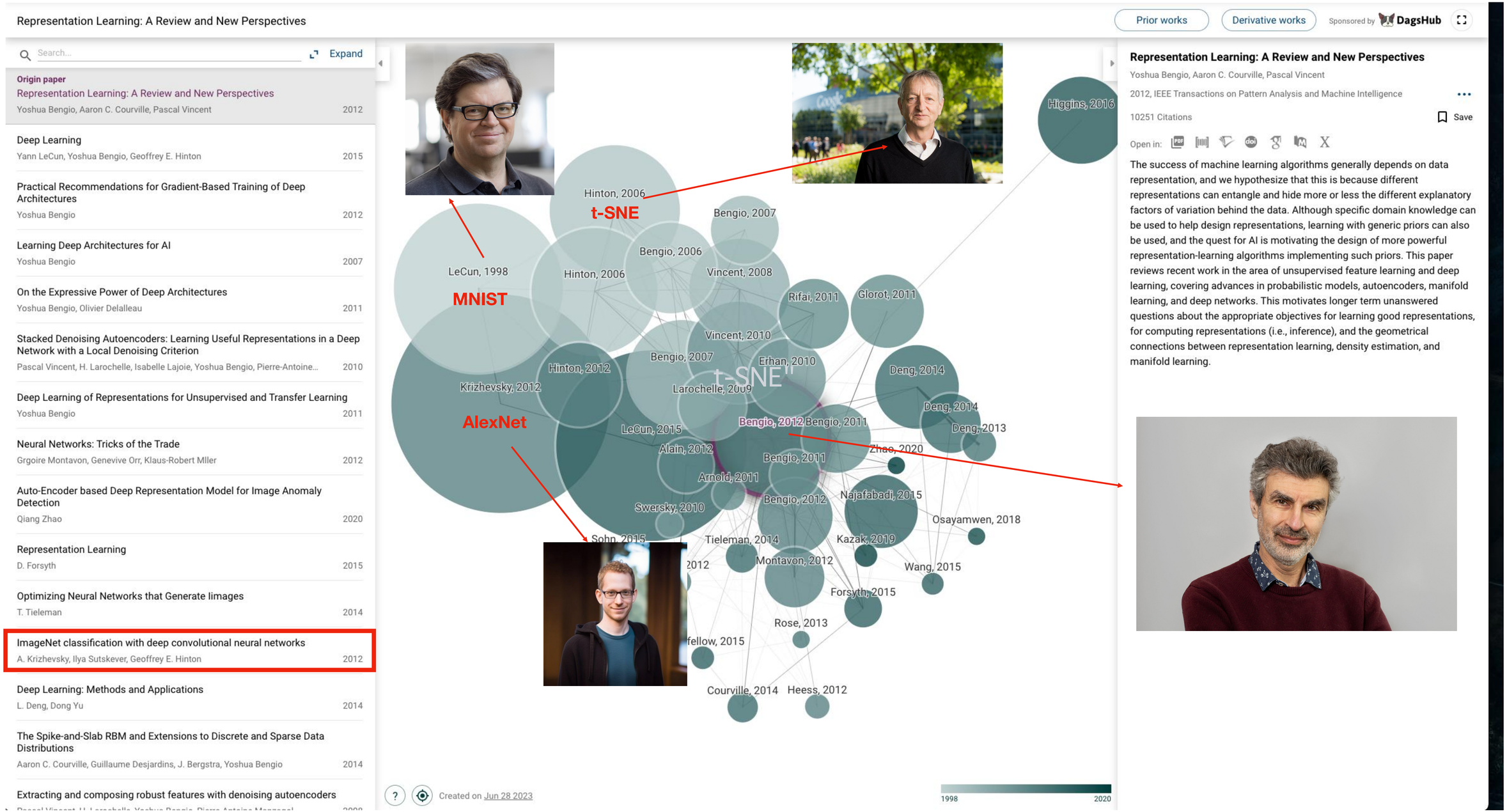
深度神经网络(DNN)：一种高效学习数据抽象特征表示的方法



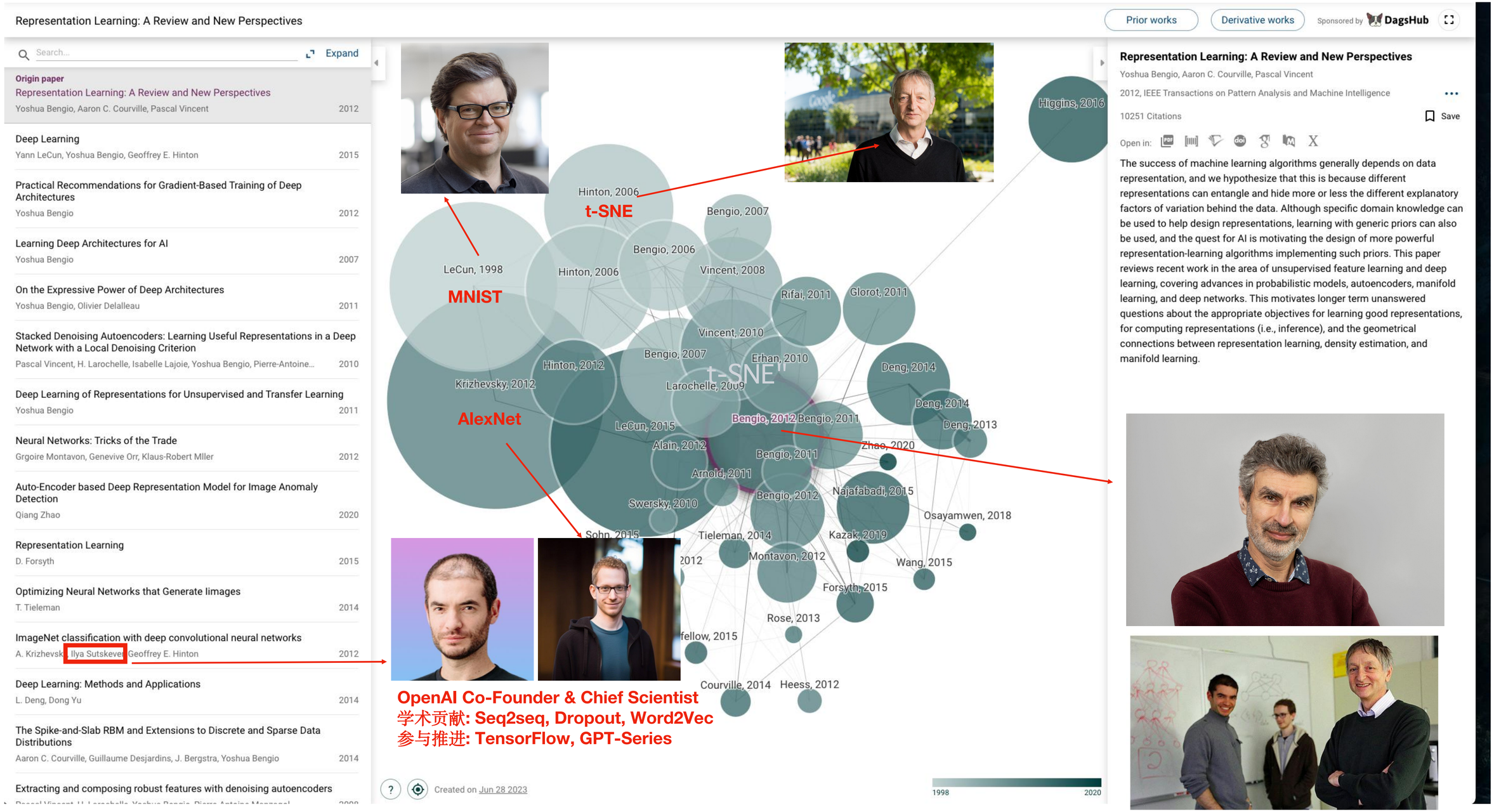
深度神经网络(DNN)：一种高效学习数据抽象特征表示的方法



深度神经网络(DNN)：一种高效学习数据抽象特征表示的方法

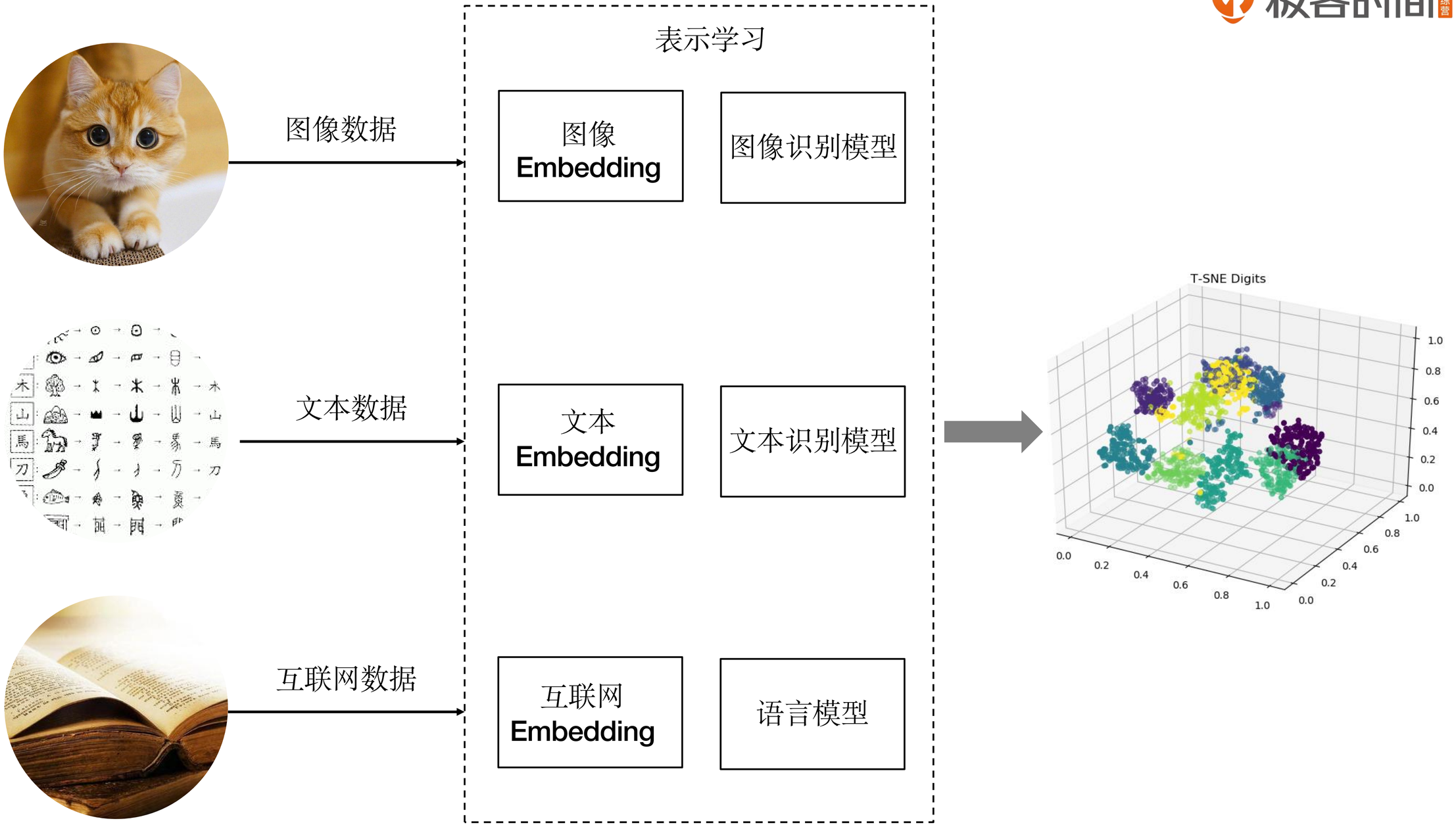


深度神经网络(DNN)：一种高效学习数据抽象特征表示的方法



维度太高，怎么知道学会了？

四六级词汇书
就记得abandon


表示学习
图像数据
图像
Embedding
图像识别模型
文本数据
文本
Embedding
文本识别模型
互联网数据
互联网
Embedding
语言模型
T-SNE Digits
0.0
0.2
0.4
0.6
0.8
1.0

嵌入（Embedding）的价值

1. **降维**： 在许多实际问题中，原始数据的维度往往非常高。例如，在自然语言处理中，如果使用One-hot编码来表示词汇，其维度等于词汇表的大小，可能达到数十万甚至更高。通过Embedding，我们可以将这些高维数据映射到一个低维空间，大大减少了模型的复杂度。

2. **捕捉语义信息**： Embedding不仅仅是降维，更重要的是，它能够捕捉到数据的语义信息。例如，在词嵌入中，语义上相近的词在向量空间中也会相近。这意味着Embedding可以保留并利用原始数据的一些重要信息。

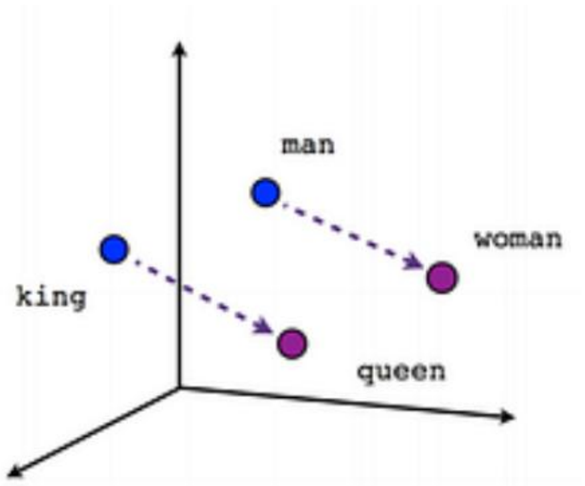
3. **适应性**： 与一些传统的特征提取方法相比，Embedding是通过数据驱动的方式学习的。这意味着它能够自动适应数据的特性，而无需人工设计特征。

4. **泛化能力**： 在实际问题中，我们经常需要处理一些在训练数据中没有出现过的数据。由于Embedding能够捕捉到数据的一些内在规律，因此对于这些未见过的数据，Embedding仍然能够给出合理的表示。

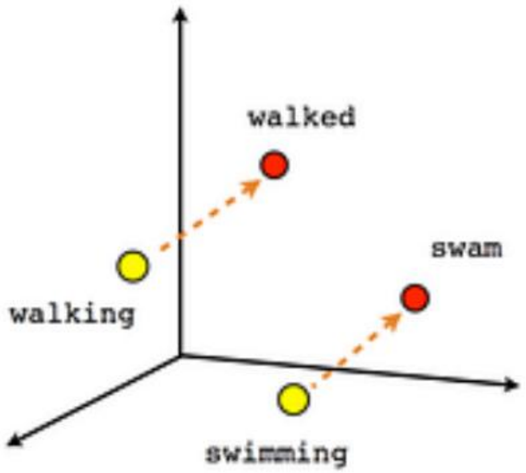
5. **可解释性**： 尽管Embedding是高维的，但我们可以通过一些可视化工具（如t-SNE）来观察和理解Embedding的结构。这对于理解模型的行为，以及发现数据的一些潜在规律是非常有用的。

Word Embedding

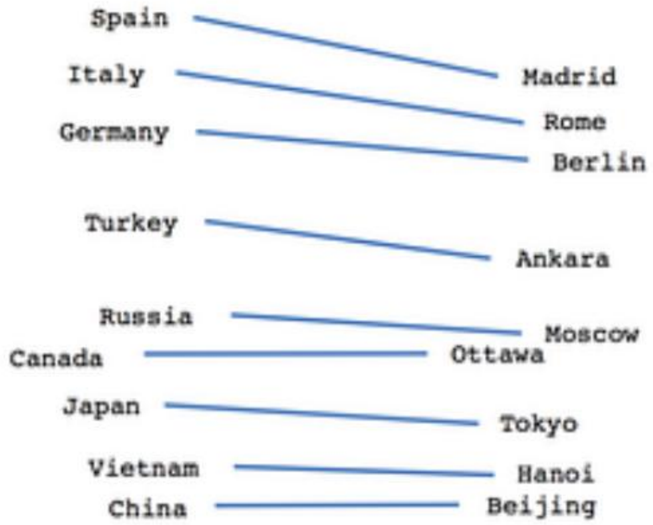
King – Man + Woman = Queen
Paris – France + Germany = Berlin



Male-Female



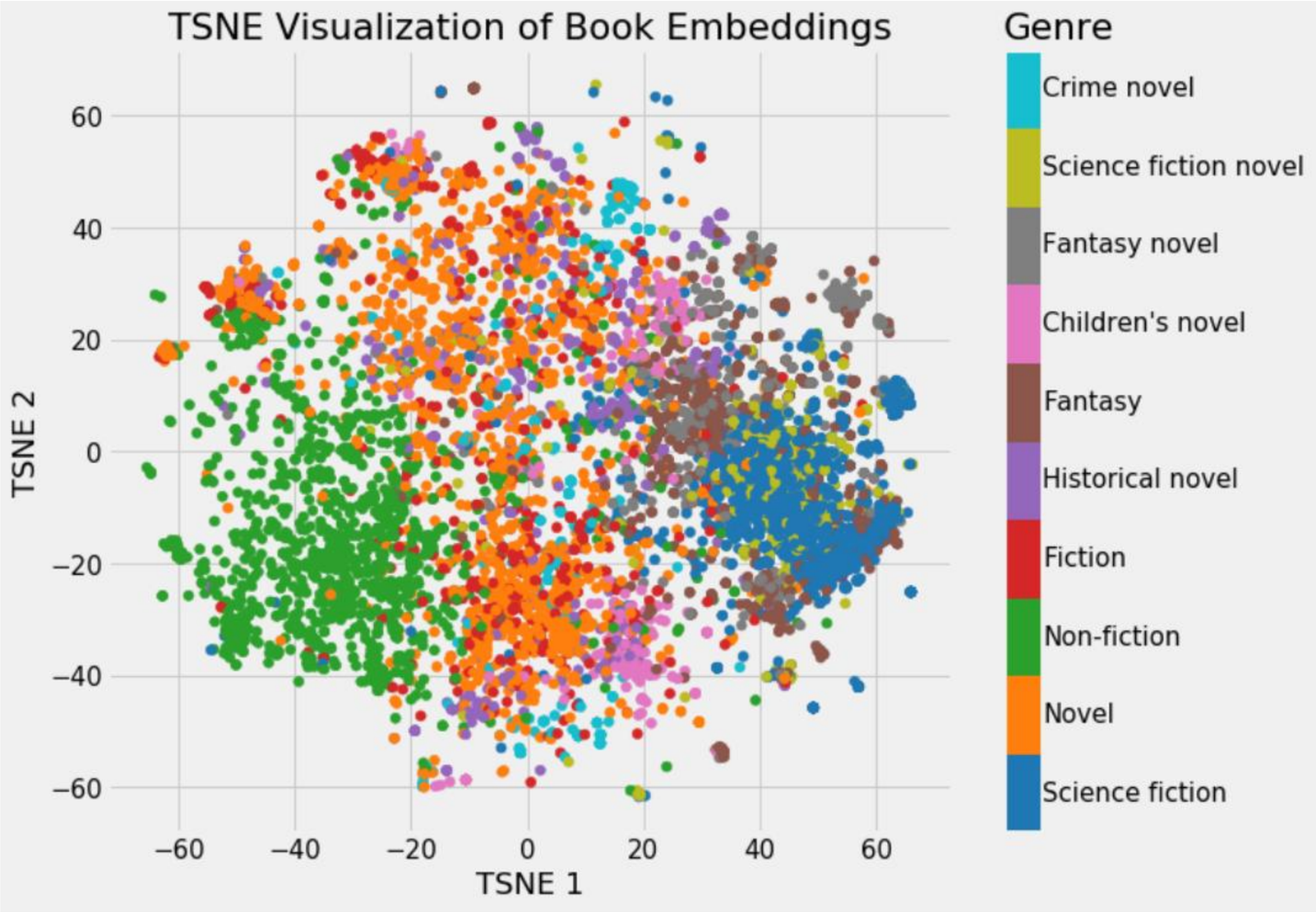
Verb tense



Country-Capital

Word embeddings capture useful relationships

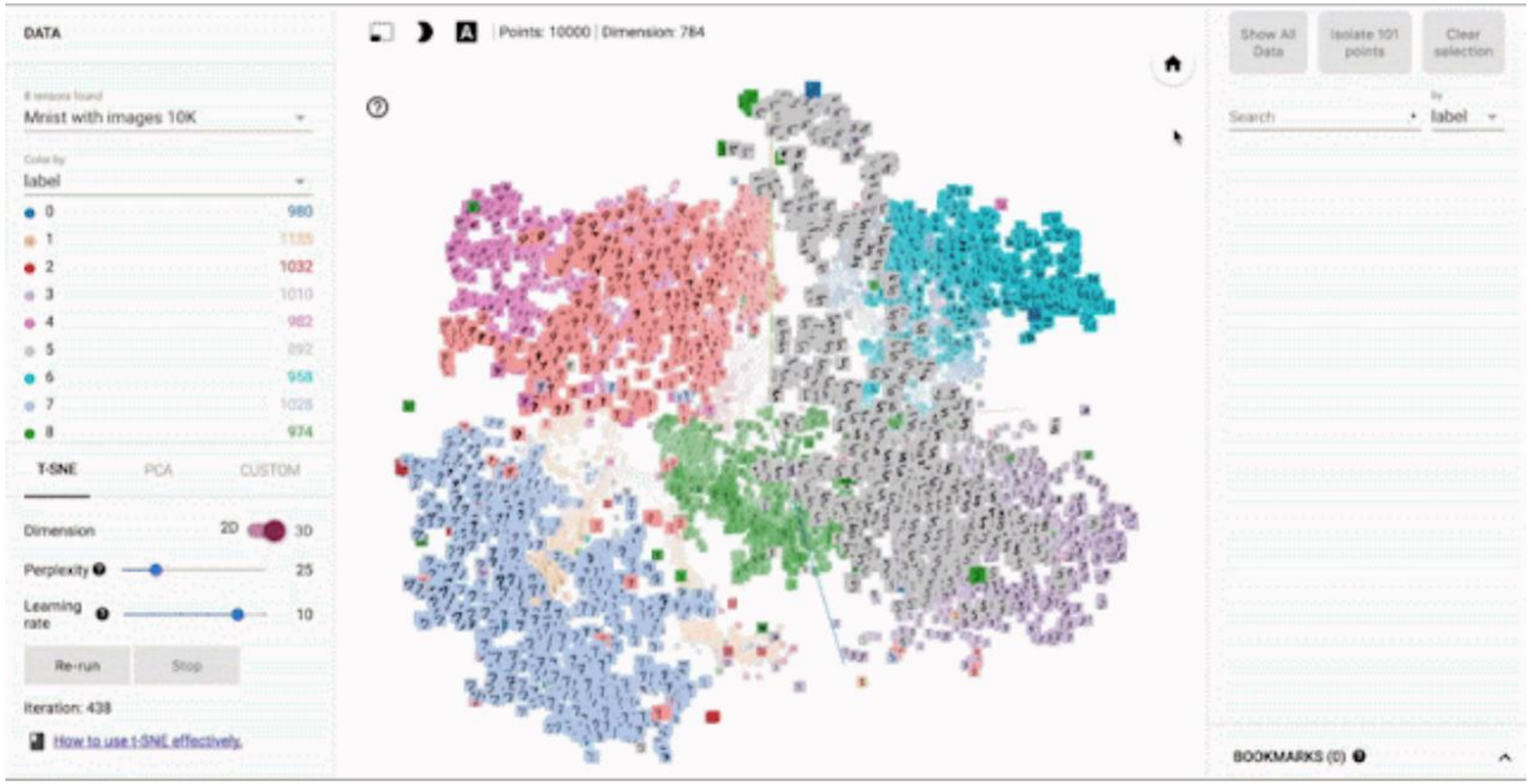
Book Embedding



Books on Wikipedia clustered by genre in just two dimensions

Image Embedding

t-SNE（t-Distributed Stochastic Neighbor Embedding）是一种统计算法，用于在低维空间（通常是2D或3D，便于可视化）里展示高维数据的结构。其目标是保持原有高维空间中近邻点的距离关系，使得距离近的点在低维空间中仍然距离近，而距离远的点在低维空间中仍然距离远。



Embedding

嵌入（**Embedding**）是表示学习的一种特定形式，旨在将高维数据映射到低维空间中的向量表示。

- **词嵌入（Word Embedding）**：在自然语言处理中，词嵌入将词语映射到低维向量空间，以捕捉词语之间的语义和句法关系。 通过学习词嵌入，可以将词语表示为连续的向量，其中相似的词语在向量空间中彼此靠近。它在自然语言处理任务中广泛应用，包括词语相似度计算、文本分类、命名实体识别等。词嵌入可以通过Word2Vec、GloVe 等方法进行学习。

- **图像嵌入（Image Embedding）**：在图像处理中，图像嵌入将图像映射到低维向量空间，以表示图像的视觉特征。 这种嵌入方法通常通过使用卷积神经网络（Convolutional Neural Networks, CNN）等深度学习技术来提取图像的特征表示。

- **图嵌入（Graph Embedding）**：是用于学习图结构的表示学习方法，将图中的节点和边映射到低维向量空间中。通过学习图嵌入，可以将复杂的图结构转化为向量表示，以捕捉节点之间的结构和关联关系。这些方法可以通过DeepWalk、Node2Vec、GraphSAGE等算法来实现。图嵌入在图分析、社交网络分析、推荐系统等领域中广泛应用，用于发现社区结构、节点相似性、信息传播等图属性。

包括嵌入在内，表示学习涉及更广泛的概念和方法，以实现对数据的更好理解和表达。

聚焦 Word Embedding

Word Embedding

Word Embedding 为自然语言处理任务提供了更好的单词表示方法，它的应用主要有：

- **语义表示和语义相似度**：Word Embedding可以捕捉到单词之间的语义关系，使得相似含义的单词在向量空间中彼此靠近。这使得我们可以通过计算向量之间的距离或相似度来度量单词的语义相似度。这对于词义消歧、语义搜索、语义推理等任务非常有用。

- **词语关系和类比推理**：Word Embedding可以捕捉到单词之间的关系，如同义词、反义词、上下位关系等。通过在向量空间中进行向量运算，例如求解向量之间的差异或类比推理，我们可以发现词语之间的有趣关系。例如，对于词语之间的类比关系"king - man + woman = queen"，通过Word Embedding可以得到类似的结果。

- **上下文理解**：Word Embedding可以帮助理解单词的上下文信息。通过学习单词在其周围上下文中的嵌入表示，我们可以捕捉到单词的语境信息，从而帮助解决语义歧义、词语消歧和指代消解等问题。

- **文本分类和情感分析**：Word Embedding可以提供丰富的词语特征表示，从而改善文本分类和情感分析任务的性能。通过将文本中的词语映射为嵌入向量，并将这些向量作为输入特征，我们可以在分类器或情感分析模型中进行训练和预测。

- **机器翻译和生成模型**：Word Embedding对于机器翻译和生成模型也是至关重要的。通过将源语言和目标语言的单词都映射为嵌入向量，可以提高翻译的准确性和生成模型的质量。嵌入向量可以捕捉到单词之间的语义和句法关系，帮助模型更好地理解和生成自然语言。

“Embedding这么厉害，还要语言模型干嘛？”

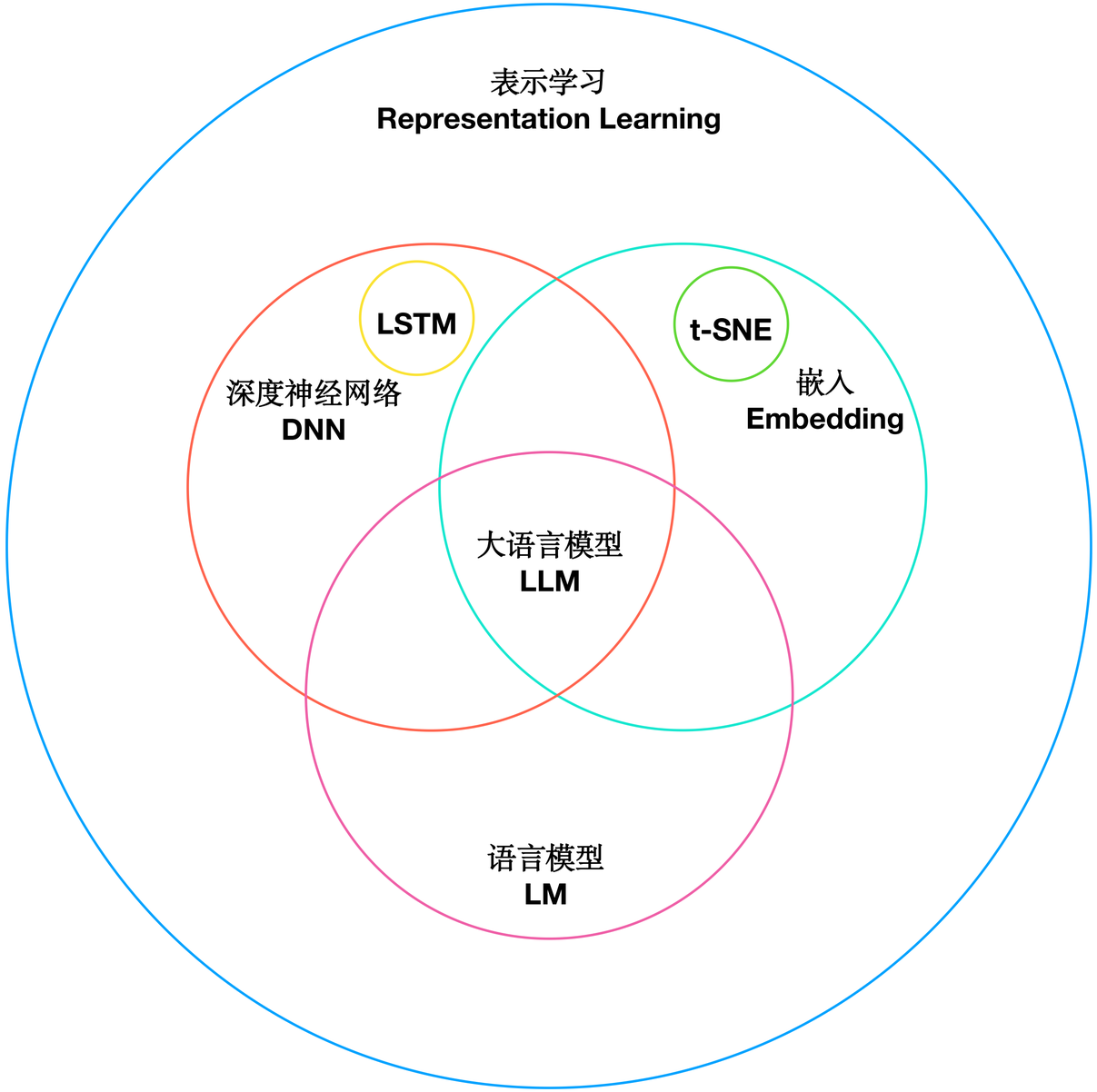
Word Embedding vs Language Model

- **Word Embedding**：词嵌入通常被用来生成词的向量表示，这个过程通常是静态的，即一旦训练完成，每个词的向量表示就确定了。词嵌入的主要目标是**捕获单词或短语的语义和语法信息**，并将这些信息以向量形式表示出来。词嵌入的一个重要特性是，语义上相近的词在嵌入空间中的距离也比较近。然而，词嵌入并不能理解上下文信息，即相同的词在不同的上下文中可能有不同的含义，但词嵌入无法区分这些含义。

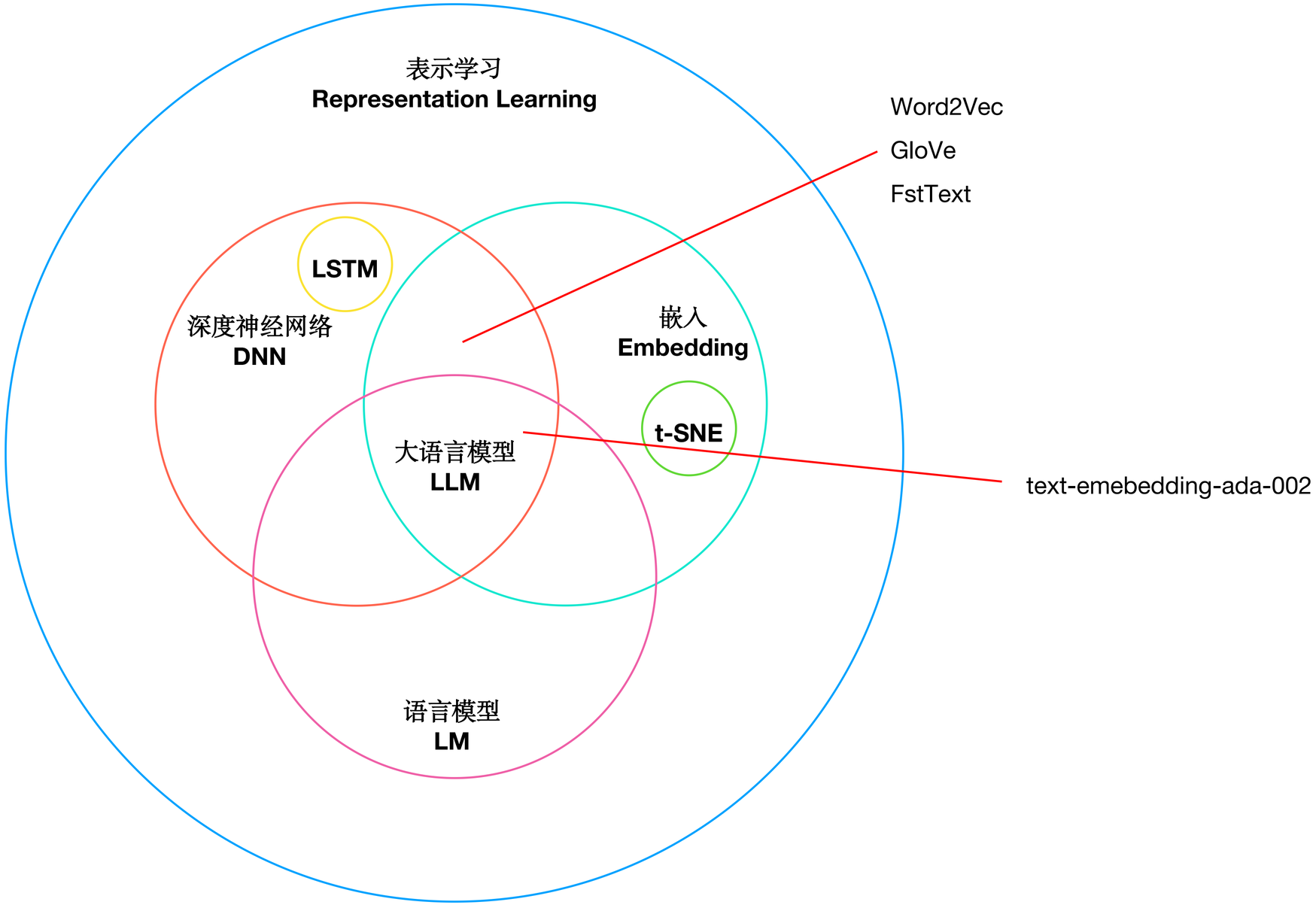
- **Language Model**：语言模型则是**预测词序列的概率模型**，这个过程通常是动态的，会根据输入的上下文进行变化。语言模型的主要目标是**理解和生成文本**。这包括对上下文的理解，词的预测，句子的生成等等。语言模型会用到词嵌入，但同时也会对上下文进行建模，这样可以处理词在不同上下文中的不同含义。

在某种程度上，你可以将词嵌入看作是语言模型的一部分或者输入，语言模型使用词嵌入捕捉的信息，来进行更深层次的语义理解和文本生成。当然，现在有一些更先进的模型，比如BERT，GPT等，它们生成的是上下文相关的词嵌入，即词的嵌入会根据上下文变化，这样一定程度上弥补了传统词嵌入模型的不足。

核心概念 维恩图


表示学习
Representation Learning
LSTM
t-SNE
深度神经网络
DNN
嵌入
Embedding
大语言模型
LLM
语言模型
LM

核心概念 维恩图


表示学习
Representation Learning
LSTM
深度神经网络
DNN
嵌入
Embedding
t-SNE
大语言模型
LLM
语言模型
LM
Word2Vec
GloVe
FstText
text-emebedding-ada-002

偏见问题

Identify *she* - *he* axis in word vector space,
project words onto this axis

Extreme *she* occupations

1. homemaker	2. nurse	3. receptionist
4. librarian	5. socialite	6. hairdresser
7. nanny	8. bookkeeper	9. stylist
10. housekeeper	11. interior designer	12. guidance counselor

Extreme *he* occupations

1. maestro	2. skipper	3. protege
4. philosopher	5. captain	6. architect
7. financier	8. warrior	9. broadcaster
10. magician	11. figher pilot	12. boss

Bolukbasi et al. (2016)

Nearest neighbor of (b -a + c)

Racial Analogies	
black → homeless	caucasian → servicemen
caucasian → hillbilly	asian → suburban
asian → laborer	black → landowner
Religious Analogies	
jew → greedy	muslim → powerless
christian → familial	muslim → warzone
muslim → uneducated	christian → intellectually

Manzini et al. (2019)

Embeddings Dev 101

开发环境

- Python 版本：3.10+
- 交互式开发：Jupyter Lab
- IDE：VS Code, PyCharm
- 大模型：OpenAI API-KEY

课程项目： GitHub openai-quickstart

极客时间 训练营

DjangoPeng / openai-quickstart

Type / to search

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

openai-quickstart Public

Unpin | Unwatch 1 | Fork 0 | Star 0

main | 1 branch | 0 tags | Go to file | Add file | Code

DjangoPeng Update README.md 57f60ba 2 days ago 2 commits

.gitignore	Initial commit	2 days ago
LICENSE	Initial commit	2 days ago
README.md	Update README.md	2 days ago

README.md

OpenAI Quickstart

This project is designed as a one-stop learning resource for anyone interested in large language models and their application in Artificial Intelligence Governance and Control (AIGC) scenarios. By providing theoretical foundations, development basics, and hands-on examples, this project offers comprehensive guidance on these cutting-edge topics.

Features

- **Theory and Development Basics of Large Language Models**: Deep dive into the inner workings of large language models like GPT-4, including their architecture, training methods, applications, and more.
- **AIGC Application Development with LangChain**: Hands-on examples and tutorials using LangChain to develop AIGC applications, demonstrating the practical application of large language models.

About

A comprehensive guide to understanding and implementing large language models with hands-on examples using LangChain for AIGC applications.

Readme
Apache-2.0 license
Activity
0 stars
1 watching
0 forks

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

https://github.com/DjangoPeng/openai-quickstart

OpenAI Embeddings 开发入门

OpenAI Embeddings

GET STARTED

Introduction
Quickstart
Libraries
Models
Overview
Model updates
GPT-4
GPT-3.5
DALL·E
Whisper
Embeddings
Moderation
GPT-3
How we use your data
Endpoint compatibility
Deprecations
Tutorials
Policies

GUIDES

GPT
GPT best practices
Image generation

Models

Overview

The OpenAI API is powered by a diverse set of models with different capabilities and price points. You can also make limited customizations to our original base models for your specific use case with fine-tuning.

MODELS	DESCRIPTION
GPT-4	A set of models that improve on GPT-3.5 and can understand as well as generate natural language or code
GPT-3.5	A set of models that improve on GPT-3 and can understand as well as generate natural language or code
DALL·E	A model that can generate and edit images given a natural language prompt
Whisper	A model that can convert audio into text
Embeddings	A set of models that can convert text into a numerical form
Moderation	A fine-tuned model that can detect whether text may be sensitive or unsafe
GPT-3 Legacy	A set of models that can understand and generate natural language
Deprecated	A full list of models that have been deprecated

We have also published open source models including Point-E, Whisper, Jukebox, and CLIP.

Visit our model index for researchers to learn more about which models have been featured in our research papers and the differences between model series like InstructGPT and GPT-3.5.

OpenAI Embeddings

Embedding models

OpenAI offers one second-generation embedding model (denoted by `-002` in the model ID) and 16 first-generation models (denoted by `-001` in the model ID).

We recommend using text-embedding-ada-002 for nearly all use cases. It's better, cheaper, and simpler to use. Read the blog post announcement.

MODEL GENERATION	TOKENIZER	MAX INPUT TOKENS	KNOWLEDGE CUTOFF
V2	cl100k_base	8191	Sep 2021
V1	GPT-2/GPT-3	2046	Aug 2020

Usage is priced per input token, at a rate of $0.0004 per 1000 tokens, or about ~3,000 pages per US dollar (assuming ~800 tokens per page):

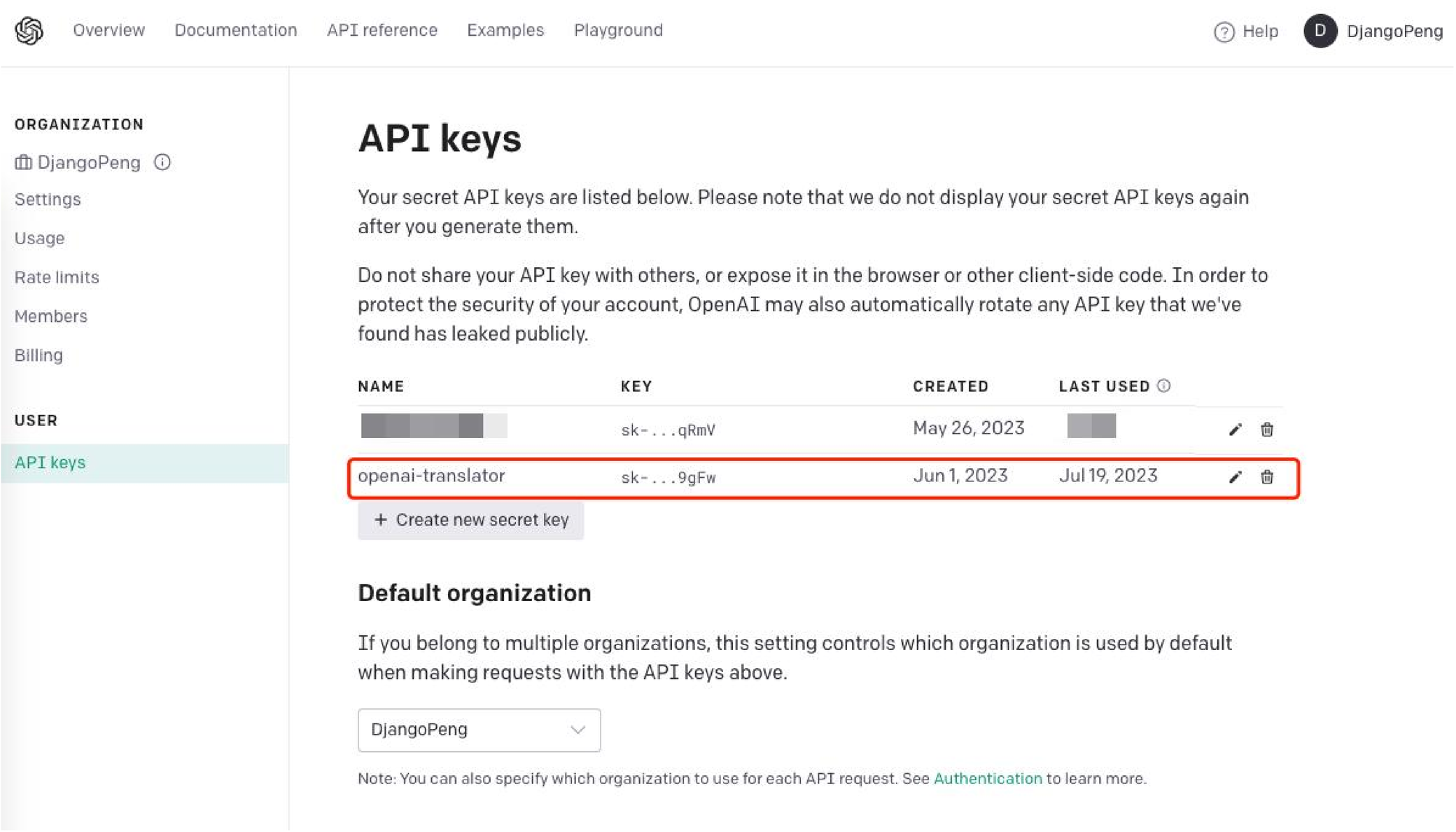
MODEL	ROUGH PAGES PER DOLLAR	EXAMPLE PERFORMANCE ON BEIR SEARCH EVAL
text-embedding-ada-002	3000	53.9
-davinci--001	6	52.8
-curie--001	60	50.9
-babbage--001	240	50.4
-ada--001	300	49.0

Second-generation models

MODEL NAME	TOKENIZER	MAX INPUT TOKENS	OUTPUT DIMENSIONS
text-embedding-ada-002	cl100k_base	8191	1536

https://platform.openai.com/docs/guides/embeddings/embedding-models

OpenAI API Key



https://platform.openai.com/account/api-keys

DEMO 实战