语义相似度计算

美化和内容（包括精炼）还未完成

语义相似度计算

目录

概况介绍

抛砖引玉

关键技术

应用场景

概况介绍

文本相似度在不同领域被广泛讨论, 由于应用场景不同, 其内涵有所差异, 故没有统一、公认的定义。

概况介绍

语义相似度是指 句子内在含义之间的相似度，是计算语言学中的一个度量，表示依赖于其层次关系的概念共性。

出自：东南大学学报（自然科学版）

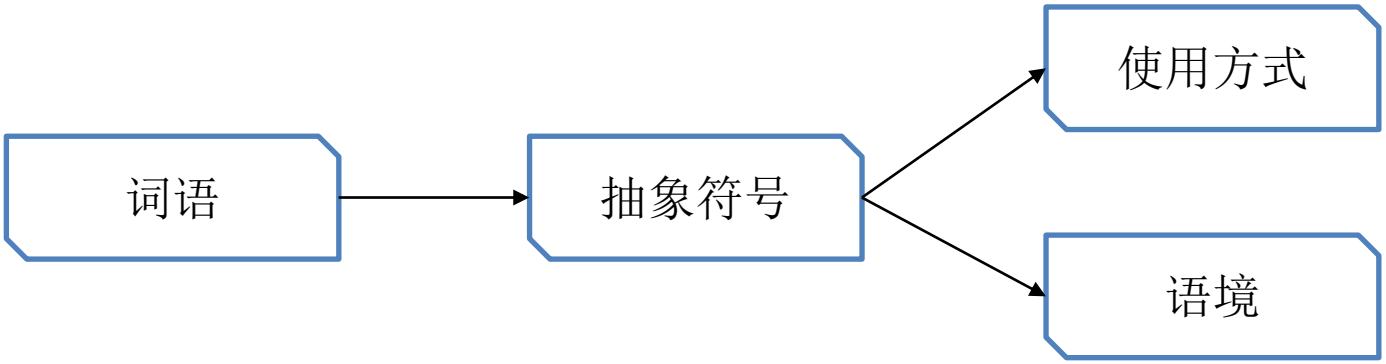
概况介绍

在自然语言语义的研究中，先驱者们把这个道理总结成了一条假设：

“词语的含义，以及词语之间语法关系的含义和这些词语与其他词语之间组合方式的限制有关。”

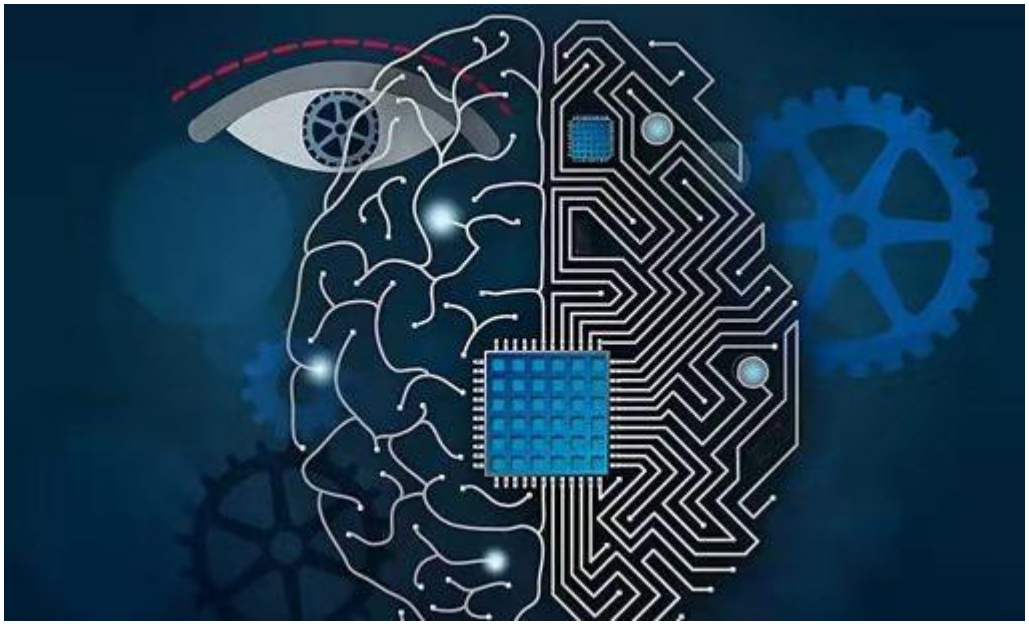
(The meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities.)

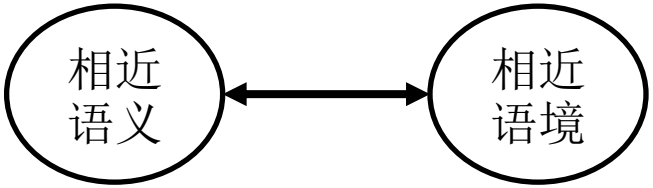
概况介绍



词语与所指的关系是极其复杂的

概况介绍





识别两个词在语义上是否相近
对多种自然语言实践任务都是有帮助的

概况介绍

那么，如何籍由词语关系来衡量一个词(所指)的意义，及其和其他词(所指)之间的互动呢？词语的“类似”依靠什么来衡量呢？在衡量过程中，是否会遇到一些特殊情况需要处理？

概况介绍

将文本相似度计算方法分为 4 大类: 基于字符串的方法、基于语料库的方法、基于世界知识的方法和其他方法。基于字符串的方法也称作“字面相似度方法”, 其中较为典型的方法包括最长公共子串 、编辑距离、Jaccard 等。由于基于字符串的方法没有考虑文本的语义信息, 计算效果受到一定限制。为解决这一问题, 学者们开始对语义相似度方法展开研究, 包括基于字符串的方法、基于语料库的方法、基于世界知识的方法和其他方法。其中其他方法又包括句法分析和混合方法, 句法分析是对句子的语法结构分析, 也属于语义分析的一种, 但其不依赖于某种语料库或世界知识, 所以

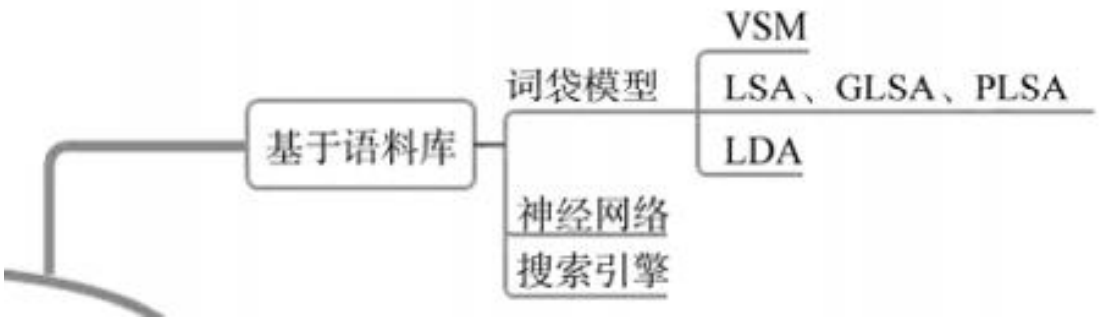
概况介绍

类型	方法	基本思想	类型	特点与不足
基于字符	编辑距离	S_A转换到 S_B需要删除、插入、替换操作的最少次数。	字符组成	计算准确，但费时。
	汉明距离[13]	$1-\left(\sum_{k=1}^{n} x_k \oplus y_k\right)/n$，其中 x_k, y_k 分别表示字符串 S_A、S_B 对应码字第 K 位的分量。	字符组成	采用模 2 加运算，简化长文本计算，效率高。
	LCS	共现且最长的子字符串。	字符顺序	原理简单，针对派生词和短文本有较好效果，但不适用于长文本。
	Jaro-Winkler	$d_j=\frac{1}{3}\left(\frac{m}{\lvert S_A\rvert}+\frac{m}{\lvert S_B\rvert}+\frac{m-t}{m}\right)$，其中 m 是匹配的字符数；t 是换位的数目。相似度计算公式为 $d_j+(lp(1-d_j))$，其中 d_j 是两个字符串的 Jaro 距离，l 是前缀相同的长度，规定最大为 4。Winkler 将 p 定义为 0.1。	字符顺序	考虑了前缀相同的重要性，针对短文本有较好效果，但不适用于长文本。
	N-gram	$\frac{\text{相似的}n\text{元组数量}}{n\text{元组总量}}$	集合思想	n 可调，方法较为灵活，但不适用于长文本。
基于词语	余弦相似度	$\frac{\overrightarrow{S_A}\cdot\overrightarrow{S_B}}{\lVert S_A\rVert\lVert S_B\rVert}$	词语组成	将文本置于向量空间，解释性强，较为常用，但不适用于长文本。
	Dice 系数[14]	$\frac{2\times comm(S_A,S_B)}{leng(S_A)+leng(S_B)}$	词语组成	增强相同部分的作用，有效关注较短的相同文本。
	欧式距离	$\sqrt{S_A^2+S_B^2}$	词语组成	算法简单直接，但效果粗糙，不适用于长文本。
	Jaccard	$\frac{S_A\cap S_B}{S_A\cup S_B}$	集合思想	不适用于长文本。
	Overlap Coefficient	$\frac{S_A\cap S_B}{\min(S_A,S_B)}$	集合思想	当一个字符串是另一个字符串的子字符串时，相似度最大。

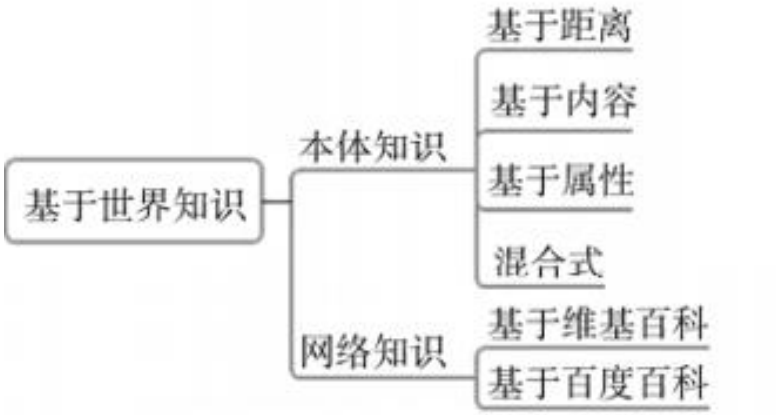
基于字符串

该方法从字符串匹配度出发, 以字符串共现和重复程度为相似度的衡量标准。根据计算粒度不同, 可将方法分为基于字符的方法和基于词语的方法。一类方法单纯从字符或词语的组成考虑相似度算法, 如编辑距离、汉明距离、余弦相似度、Dice 系数、欧式距离; 另一类方法还加入了字符顺序, 即字符组成和字符顺序相同是字符串相似的必要条件 , 如最长公共子串 、Jaro-Winkler; 再一类方法采用集合思想, 将字符串看作由词语构成的集合, 词语共现可用集合的交集计算, 如 N-gram、Jaccard、Overlap Coefficient。基于字符串的方法是在字面层次上的文本比较, 文本表示即为原始文本。该方法原理简单、易于实现, 现已成为其他方法的计算基础。但不足的是将字符或词语作为独立的知识单元, 并未考虑词语本身的含义和词语之间的关系。以同义词为例, 尽管表达不同, 但具有相同的含义, 而这类词语的相似度依靠基于字符串的方法并不能准确计算。

基于语料库



基于语料库的方法利用从语料库中获取的信息计算文本相似度。基于语料库的方法可以分为: 基于词袋模型的方法、基于神经网络的方法和基于搜索引擎的方法。前两种以待比较相似度的文档集合为语料库, 后一种以 Web 为语料库。



3.3 基于世界知识

基于世界知识的方法是指利用具有规范组织体系的知识库计算文本相似度, 一般分为两种: 基于本体知识和基于网络知识。前者一般是利用本体结构体系中概念之间的上下位和同位关系, 如果概念之间是语义相似的, 那么两个概念之间有且仅有一条路径[7,10]。而网络知识中词条呈结构化并词条之间通过超链接形式展现上下位关系, 这种信息组织方式更接近计算机的理解。概念之间的路径或词条之间的链接就成为文本相似度计算的基础。

	基于距离	基于内容	基于属性	混合式
基本原理	用概念之间的路径长度表示语义距离	用概念词共享的信息量化它们之间的语义相似度	用概念词之间的公共属性数量衡量它们之间的相似度	将基于距离、基于内容、基于属性三种方法综合计算概念之间的相似度
代表方法	Shortest Path[38]、Wu 等[39]、Weighted Links[40]、Li 等[41]、刘群等[10]	Lin[42]、Resnik[43]、Lord 等[44]、边振兴[45]	Tversky[46]	葛斌等[47]、王艳娜等[48]、李文清等[49]
特点	在计算方法中加入了节点深度、密度、强度、宽度及分类体系层次等影响因子	计算方法采用不同节点的信息量以及表达信息内容的不同公式	计算效果依赖于本体属性集的完整性	计算方法中权重参数设置大多依赖领域专家

(1) 基于本体

文本相似度计算方法使用的本体不是严格的本体概念, 而指广泛的词典、叙词表、词汇表以及狭义的本体。随着 Berners-Lee 等提出语义网的概念, 本体成为语义网中对知识建模的主要方式, 在其中发挥着重要作用。由于本体能够准确地表示概念含义并能反映出概念间的关系, 所以本体成为文本相似度的研究基础[7]。最常利用的本体是通用词典, 例如 WordNet、《知网》(HowNet)和《同义词词林》等, 除了词典还有一些领域本体, 例如医疗本体、电子商务本体、地理本体、农业本体等。基于本体的方法将文本表示为本体概念以及概念之间的关系, 该方法能够准确反映概念内在语义关系, 是一种重要的语义相似度计算方法, 主要缺点如下: ①本体一般需要专家参与建设, 耗费大量时间和精力, 而已有的通用本体存在更新速度慢、词汇量有限等问题, 不适用于出现的新型词语; ②利用本体计算文本相似度, 首先是在词语层次进行计算, 然后累加词语相似度获得长文本相似度, 相对基于语料库的方法对文本整体处理而言计算效率较低; ③无论是通用本体还是领域本体, 本体之间相互独立将带来本体异构问题, 不利于跨领域的文本相似度计算。

(2) 基于网络知识

由于本体中词语数量的限制, 有些学者开始转向
基于网络知识方法的研究, 原因是后者覆盖范围广
泛、富含丰富的语义信息、更新速度相对较快, 使用
最多的网络知识是维基百科、百度百科。网络知识一
般包括两种结构, 分别是词条页面之间的链接和词条
之间的层次结构。孙琛琛等[50]将其概括为: 文章网络
和分类树(以树为主题的图)。

今后文本相似度计算方法的趋势有以下三个方面:
(1) 基于神经网络的方法研究将更加丰富。由于词向量表示文本, 所表达的文本语义信息更符合人类认知, 所以随着第三次人工智能浪潮的到来, 神经网络算法将得到不断改进, 基于神经网络的文本相似度计算也必将得到更多探索。
(2) 网络资源为文本相似度计算方法研究提供更多支持。Web3.0、移动网络以及未来5G 技术的实现, 网络资源无疑是最大、最丰富的语料库, 与此同时语义网和关联数据进一步发展, 网络文本资源面向结构化与互连化。所以新型的信息组织结构与信息之间的链接方式将应用到文本相似度计算之中。
(3) 针对特定领域以及跨领域文本的相似度计算将成为今后发展的重点。跨学科合作越来越趋于常态化, 领域专家的合作促进跨领域世界知识的集成并为跨领域文本的相似度计算提供便捷的人工参与和建议。

抛砖引玉

抛砖引玉

问题1：如何定义语境？如何衡量两个语境是否类似？

问题2：如果一个词能够指代多个事物，如何区分对应的不同语境？

问题3：两个事物之间几乎不共享语境元素，是否代表它们没有关系？

抛砖引玉

1. 向量空间模型与相似度计算

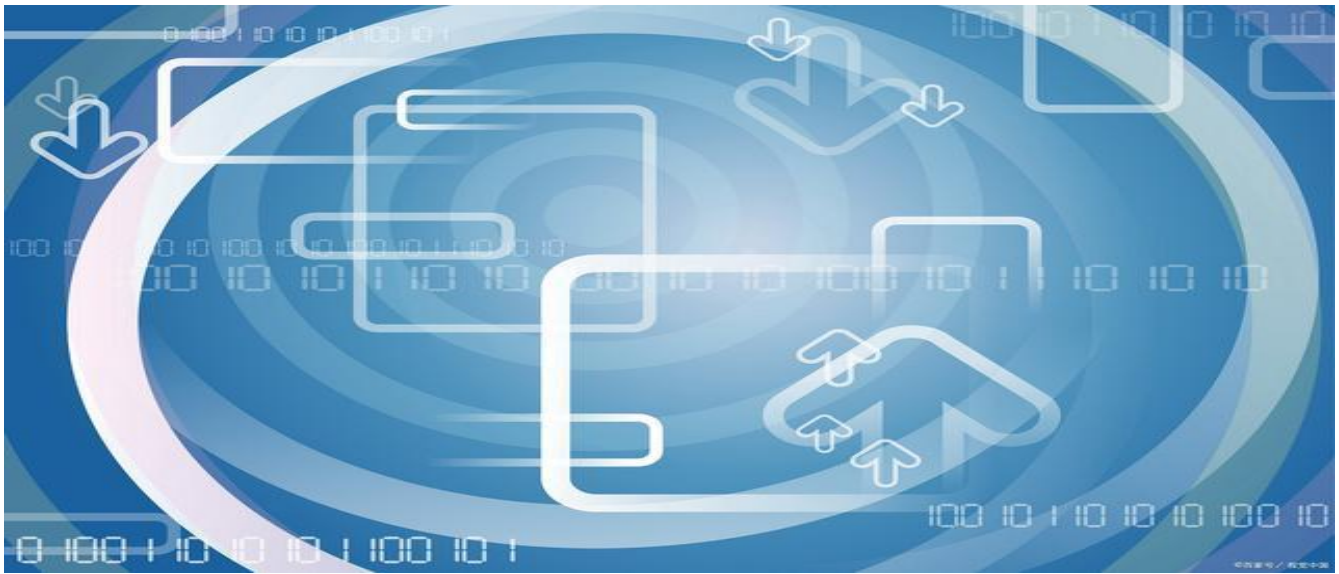
1.1 向量空间模型

这里举一个仅考虑上下文词的例子(已分好词)：

武林高手：1.经常 2.从 3.山川 4.之间 5.顿悟 6，并 7.由 8.山川之形 9.变化 10.出 11.上乘 12.武艺 13。

上下文的选取是要考虑范围的，离目标词太远的词，就可以忽略不计了。如果百度考虑“山川”的上下文，且忽略距离“山川”2个词以上的词，那么“山川”的上下文就可以用下属向量表示：

(0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0)



抛砖引玉

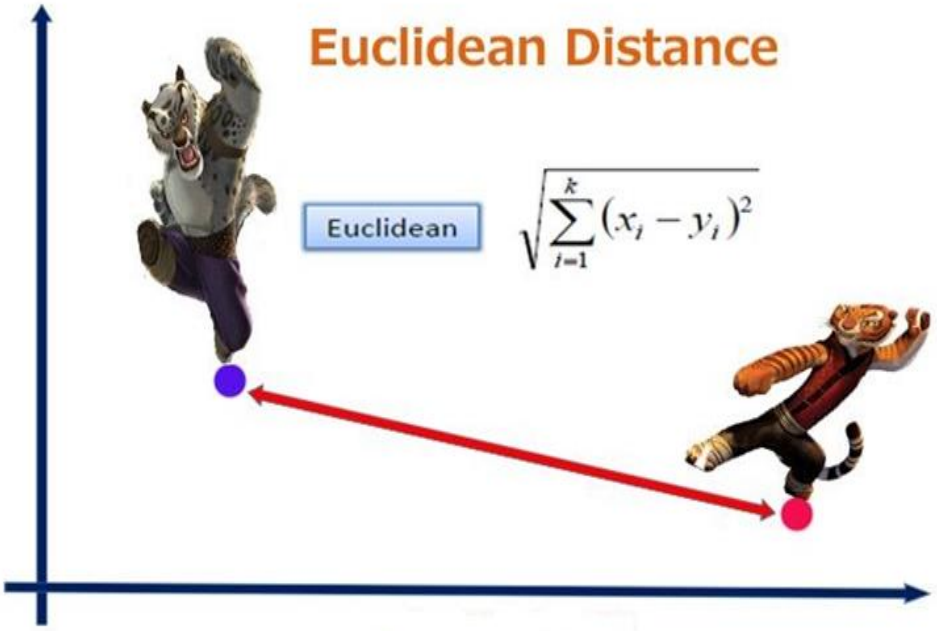
1.2 基本语义相似度计算

词u在不同的地方出现过n次，进而形成了n个上下向量U1…Un,那么我们就可以用集合Su={undefinedU1…Un}来表示w的语境。
而词v的语境可以用集合Sv={undefinedV1…Vm}来表示。

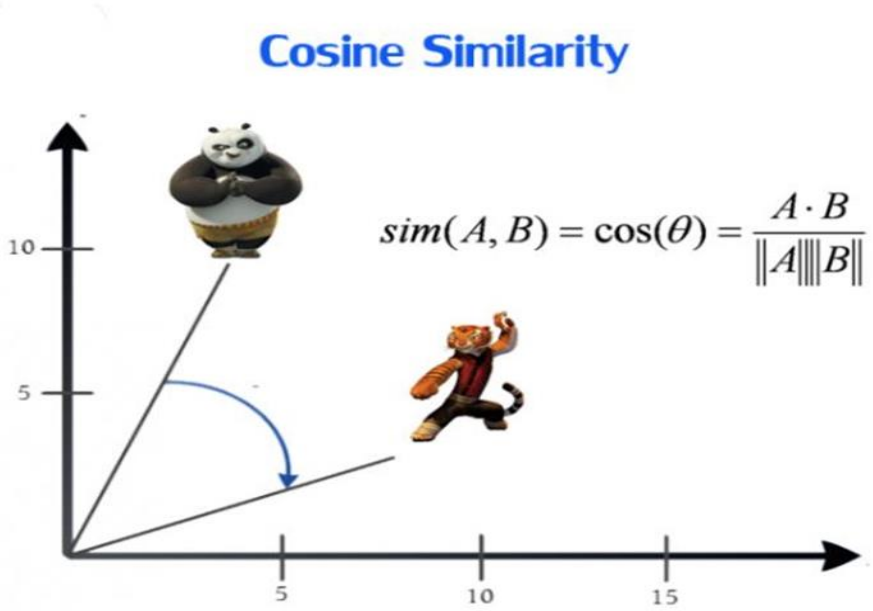
接下来，如何判断u和v在语境上，进而在语义上是否相似呢？

抛砖引玉

在这个问题上，我们面临着两种选择，也对应着对向量空间的两种不同的理解：



a.欧氏向量空间



b.余弦相似度

抛砖引玉

现在，我们用了一组向量来表示一个词的语境，那么计算相似度的时候到底要用其中哪些向量，还是全部使用呢？

这个问题就引出了语义相似度计算中的两种重要实践：

a) 原型(prototype)方法
对于Su={undefinedU1…Un}，原型方法将所有向量取平均，形成一个向量使用。

b) 范例方法
对于Su={undefinedU1…Un}和Sv={undefinedV1…Vm}两组向量范例方法保留所有向量，两个集合Su、Sv间的相似度采用向量两两间相似度的均值，或者最小值/最大值表示。

抛砖引玉

2. 带约束的语义相似度计算

何为约束——

关键词差异：手机　牛仔裤

选择集合S中和关键词k相似度大于某个阈值的向量使用

对目标词v的向量V和关键词k的向量K执行某种混合运算

只从关键词的周围选上下文

这引出了下一个问题：

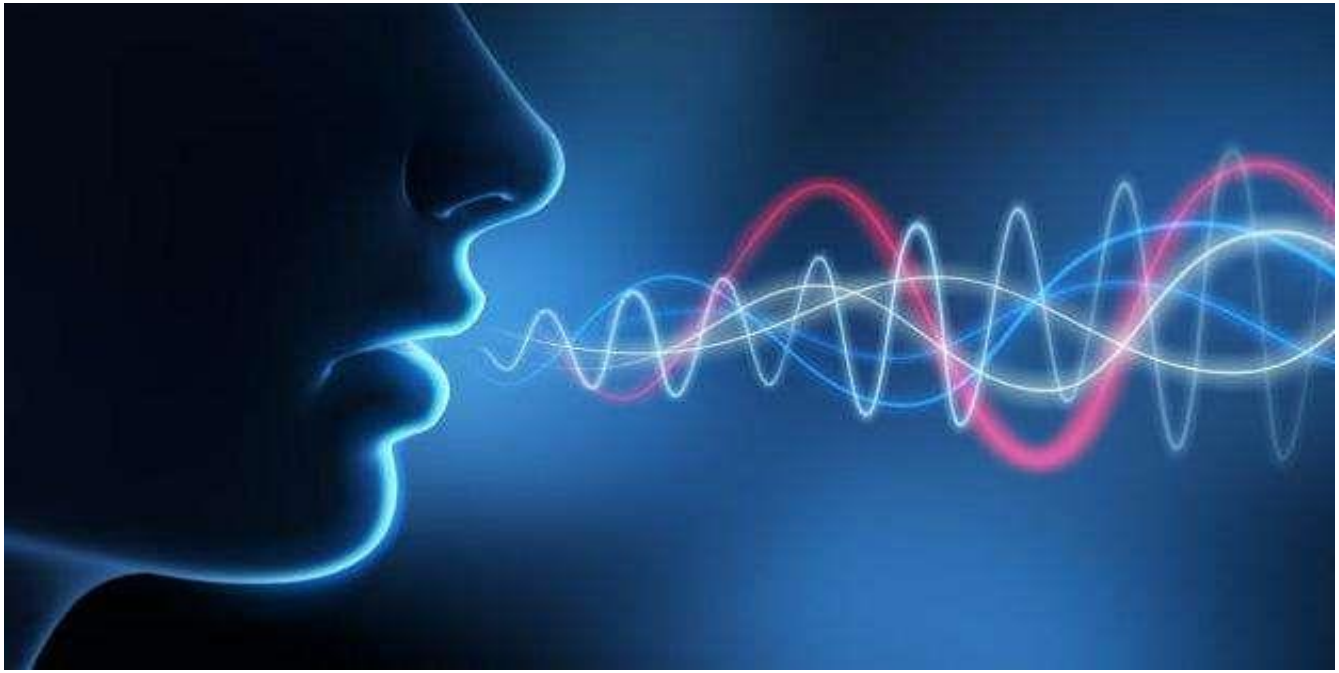
求交后向量变得稀疏，使得很多本来相似的词由于共享分量太少，变得不相似了，这个时候应该怎么办呢？

抛砖引玉

3. 平滑的语义相似度计算

在原形方法中，所谓平滑，是指用一个词的相似词来代表它本身 。

平滑的计算结果在排序上要略好于非平滑结果(噪音排下去了)。



技术简介

一、 基于拓扑相似

1. Leacock & Chodorow [1]提出

$$Sim_{lch} = -\log \frac{length}{2 * D}$$

其中length是至两个概念之间最短节点计数距离，D是整个结构最大深度。

2. Wu & Palmer[2]提出

$$Sim_{wup} = \frac{2 * depth(LCS)}{depth(concept_1) + depth(concept_2)}$$

3. Resnik[3]提出

$$Sim_{res} = IC(LCS)$$

LCS跟上同义，IC 为information content由下面公式计算得出

$$IC(c) = -\log P(c)$$

其中P(c)是指在知识拓扑中出现概念c的实例的概率。

4. Lin提出[4]

$$Sim_{lin} = \frac{2 * IC(LCS)}{IC(concept_1) + IC(concept_2)}$$

基于Resnik方法。

二、 基于统计相似

这种相似计算需要有一个语料库，以下是几个典型算法：

1. Normalized Google Distance (NGD)[5]

$$\mathrm{NGD}(x,y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log N - \min\{\log f(x), \log f(y)\}}$$

2. Pointwise Mutual Information (PMI) [6]

$$PMI\text{-}IR(w_1, w_2) = log_2 \frac{p(w_1 \& w_2)}{p(w_1) * p(w_2)}$$

3. Latent semantic analysis (LSA)[7]

例子：待完成

基于深度学习的短文本语义相似度计算模型

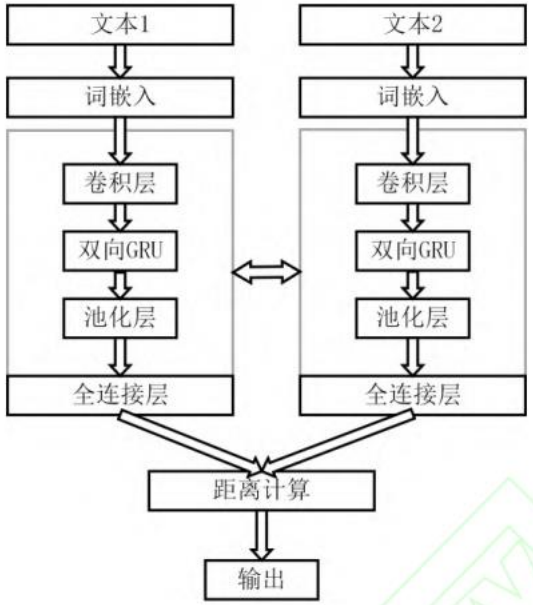


图 1 语义相似度计算模型

Fig. 1 Semantic similarity computing model

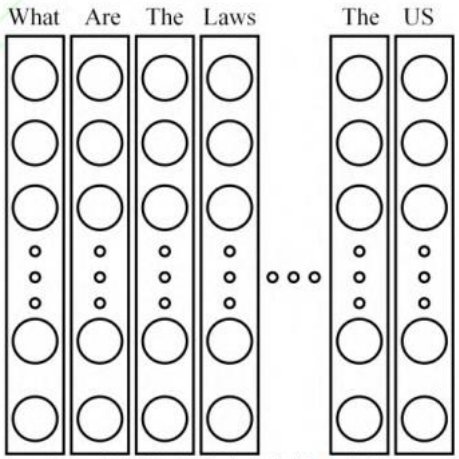


图 2 词嵌入后文本的向量化表示

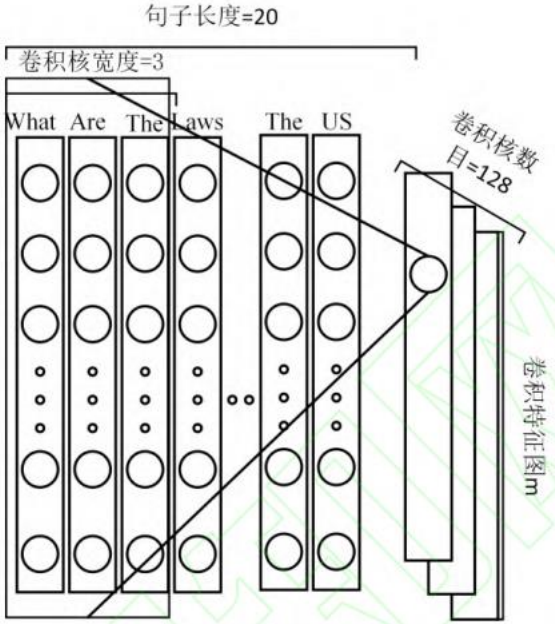


图 3 卷积层示意

Fig. 3 Schematic diagram of convolutional layer

应用场景

OBIE[7]是基于 DBpedia 和 FreeBase 的 Twitter 命名实体识别算法，算法利用 Gate 中的 ANNIE 词典以及抽取自维基百科和 DBpedia 的词典，基于规则模板，使用模式匹配的方式来定位 Tweets 中的命名实体指称，利用源自 FreeBase API 的流行度得分[8]和基于依存关系树（Dependency Tree） 的句法相似

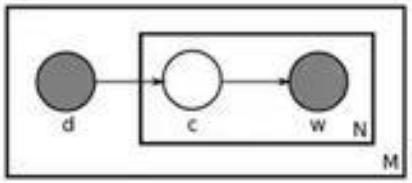
度对候选资源进行排序，返回相似度最高的实体及其在 DBpedia 本体中的分类。通过对 BBC 新闻、纽约时报和时代周刊三个 Twitter 账户下 115 条 Tweets 在人名、地名和组织名三个类别上进行识别测试，最终结果显示，相比于无消歧的命名实体识别方法，有消歧算法的 F 值得到了显著提升。

Damljanovic 等[9] 利 用 Gate 中 的 ANNIE 和 LKB（Large Knowledge Gazetteer） 组件定位文本中的命名实体指称，通过 rdfs:label 和 foaf:name 属性进行匹配获取实体指称对应的候选资源，通过基于编辑距离 （Levenshtein Distance） 的字符串相似度、基于实体共现的结构相似度和基于随机索引 （Random Indexing） 的上下文相似度计算命名实体指称与候选资源之间的相似度，返回相似度最高的候选实体。通过测试发现，ANNIE+LKB+消歧的算法相比于 ANNIE 和 ANNIE+LKB，识别的准确性和 F 值显著提升，而查全率略有降低。

应用场景

Topic Model 的语义表示技术：文档降维、文档映射。
通过主题进行语义表示

百度早期语义表示技术 ：Topic Model

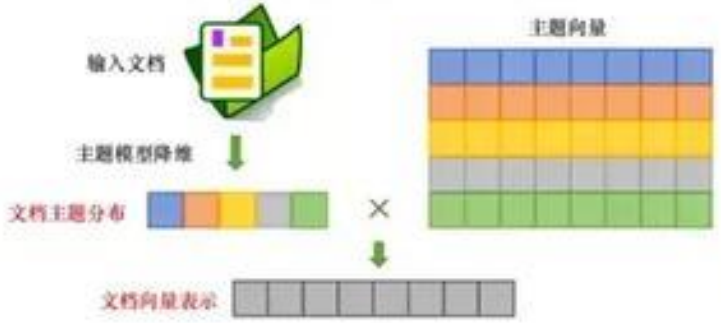


$$P(w,d)=\sum_c P(c)P(d|c)P(w|c)=P(d)\sum_c P(c|d)P(w|c)$$

技术难点：超大规模语料训练
Online EM
MPI 并行化

应用场景：
广告召回、相关性计算
情感分析
文本分类
……

在篇章语义表示上效果明显



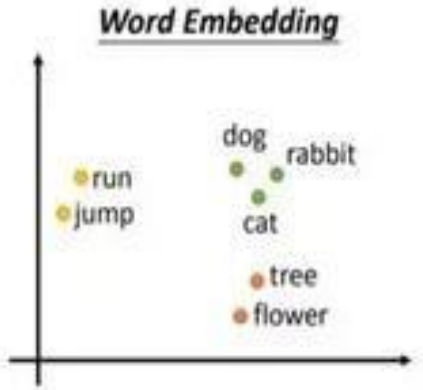
百度NLP主题模型工具包

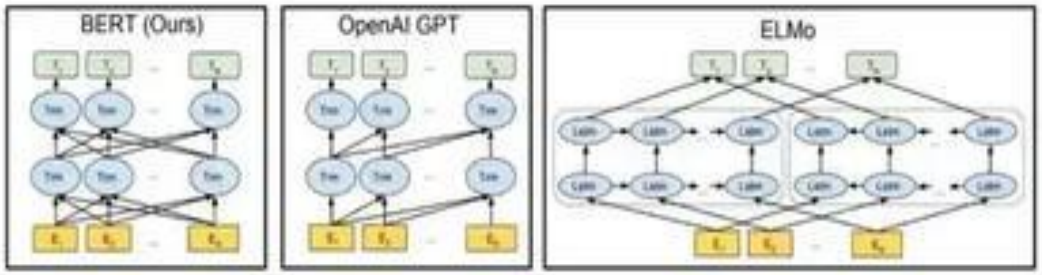
FAMILIA

应用场景

基于DNN的语义表示技术兴起: word embedding

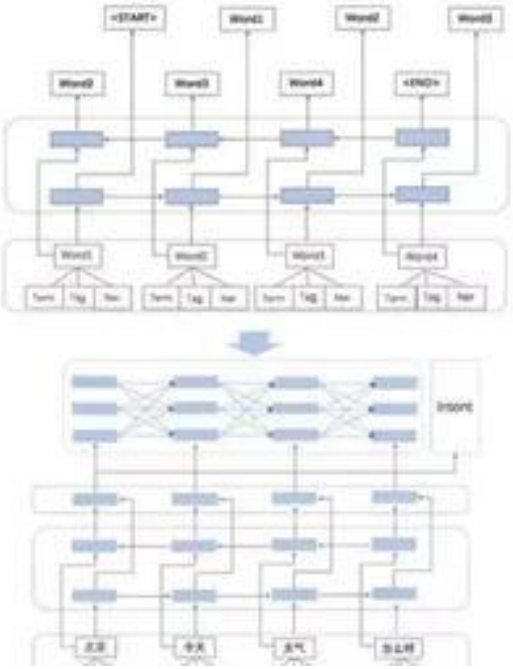
- Context-independent Word Embedding
  - NNLM (2003)
  - C&W (2008)
  - Word2Vec (2013)
  - GloVe (2014)
- Context-aware Word Embedding
  - Cove (2017.12)
  - ELMo (2018)
  - GPT (2018)
  - BERT(2018)

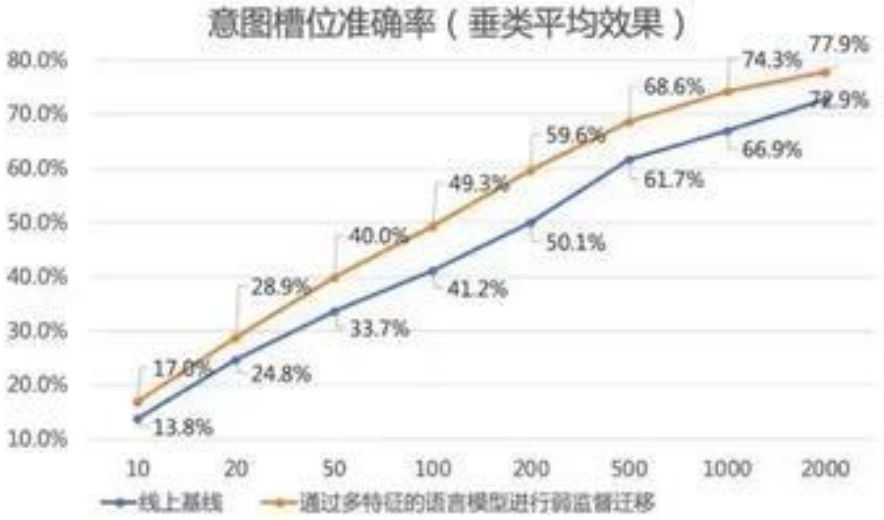




基于大规模表示迁移的探索：多特征融合的表示模型 (2017年)

基于20亿搜索Query构建多特征语义表示提升SLU的Intent&Slot效果





同结构模型迁移，小数据任务，通用性验证不充分

应用场景

最新研究——知识增强的语义表示模型

- **BERT 基于基本语言单元语义建模**
  - 词汇/实体中局部语言规律，使得模型很容易推测出掩码的字信息
  - 缺乏显式的语义概念（哈尔滨、黑龙江），以及对应语义关系（省会）的建模

- **ERNIE基于知识增强语义建模**
  - 保持基于字特征输入基础上，显式建模语义单元（词、实体）的语义知识，保持字特征语义组合的灵活性
  - 无监督学习自然本文中的真实世界知识

应用场景

实验效果

ERNIE中文效果全面领先BERT

NLP-TASK (中文)	SoTA	ERNIE	BERT
NER (MSRA-NER)	93.2%	93.8%(+1.2)	92.6%
Inference (XNLI)	68.3%	78.4%(+1.2)	77.2%
QA (DBQA)	-	82.7%(+1.9)	80.8%
Sentiment Classification (ChnSentiCorp)	-	95.4%(+1.1)	94.3%
Semantic Similarity (LCQMC)	83.4%	87.4%(+0.4)	87.0%

在中文上做了 ERNIE(1.0)实验，找了五个典型的中文公开数据集做对比。不管是词法分析 NER、推理、自动问答、情感分析、相似度计算，ERNIE(1.0)都能够显著提升效果。

应用场景

实验效果

ERNIE英文效果全面领先BERT（GLUE & SQuAd）

NLP-TASK (English)	Base		Large	
	ERNIE	BERT	ERNIE	BERT
CoLA(The Corpus of Linguistic Acceptability)	54.8 (+2.7)	52.1	62.1 (+1.6)	60.5
SST(The Stanford Sentiment Treebank)	94.0 (+0.5)	93.5	94.8 (-0.1)	94.9
MRPC(Microsoft Research Paraphrase Corpus)	88.6/84.3 (-0.3/-0.5)	88.9/84.8	89.0/85.0 (-0.3/-0.4)	89.3/85.4
STSB(Semantic Textual Similarity Benchmark)	87.2/86.0 (+0.1/+0.2)	87.1/85.8	88.6/87.7 (+1/+1.2)	87.6/86.5
QQP(Quora Question Pairs)	72.4/89.6 (+1.2/+0.4)	71.2/89.2	72.4/89.7 (+0.3/+0.4)	72.1/89.3
MNLI(Matched)	85.2 (+0.6)	84.6	87.1 (+0.4)	86.7
MNLI(Mismatched)	84.2 (+0.8)	83.4	86.4 (+0.5)	85.9
QNLI(Question NLI)	92.0 (+1.5)	90.5	93.7 (+1.0)	92.7
RTE(Recognizing Textual Entailment)	70.8 (+4.4)	66.4	75.8 (+5.7)	70.1
WNLI(Winograd NLI)	65.1	65.1	65.1	65.1
Squad 1.1 - **Dev**	83.5/90.3	80.8/88.5	86.6/92.8	84.1/91.8

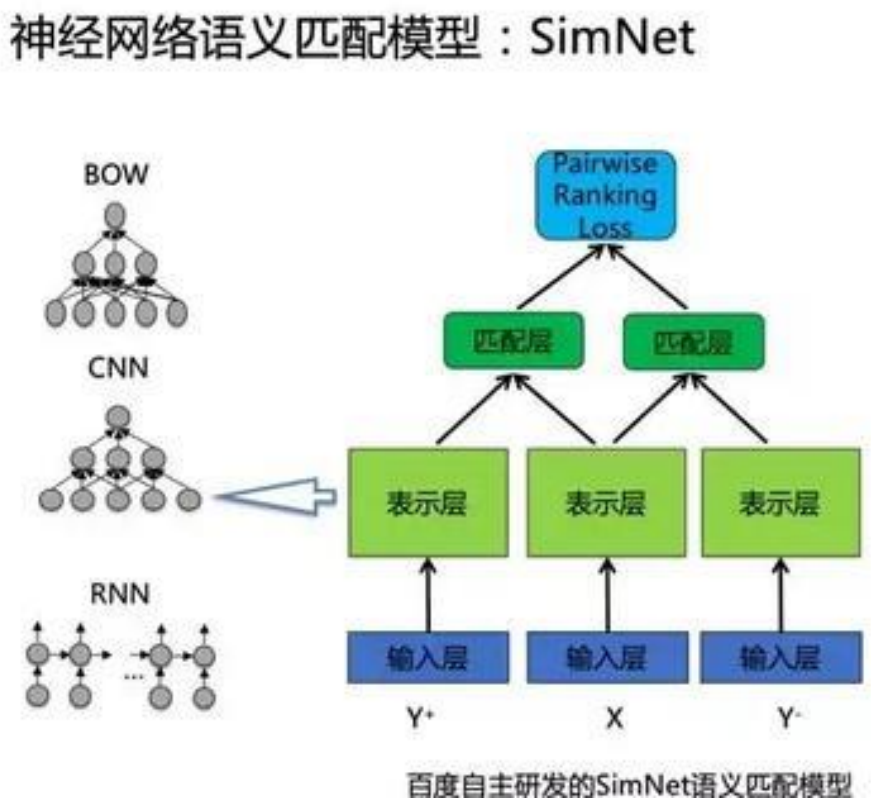
应用场景

语义匹配

1. 文本语义匹配及挑战

应用场景

2. 神经网络语义匹配模型：SimNet

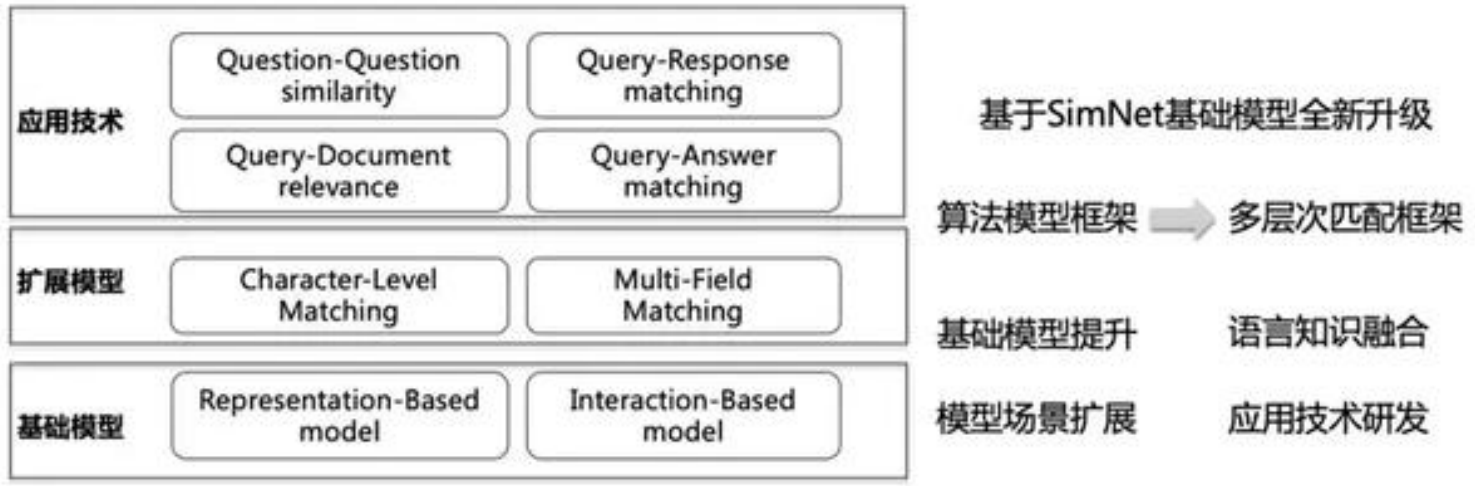


- SimNet
  - 百度NLP于2013年设计研发
  - 沿袭Word Embedding语义表示
  - 有监督的End-to-End语义匹配框架
  - DNN、CNN、RNN模型建模句子不同特性
  - Pairwise训练框架
- 近年来相关工作
  - Microsoft：DSSM、DUET
  - 华为Noah's Ark Lab：Arc-I、Arc-II
  - 中科院：MV-LSTM、MatchPyramid
  - CMU：K-NRM、Conv-KNRM

应用场景

这几年，百度整体上从语义匹配的框架上做了升级，抽象了三个层次

SimNet2.0框架

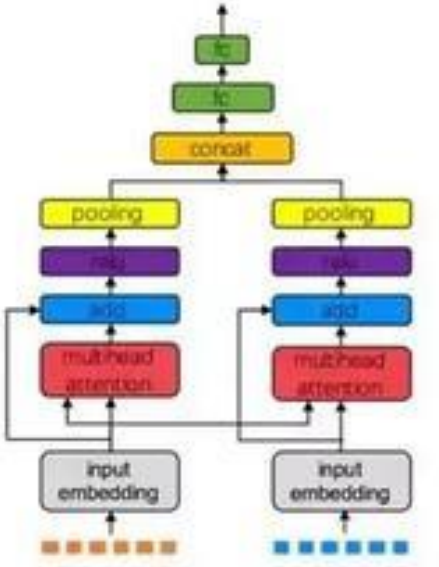


SimNet2.0框架

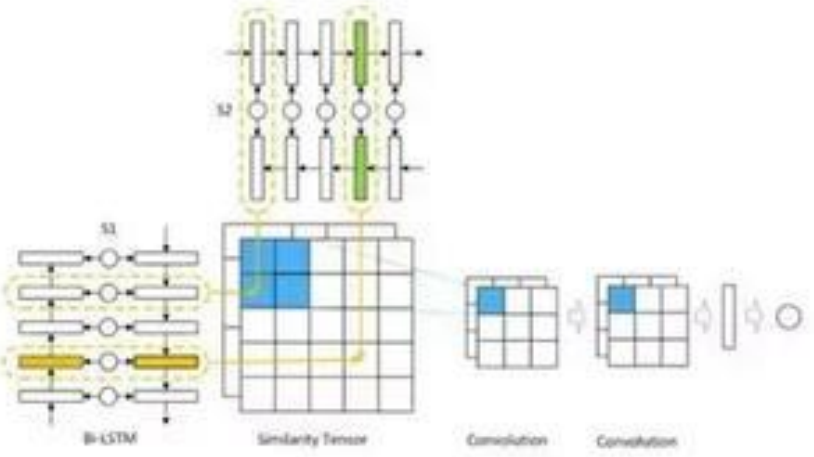
应用场景

基础模型研发

- Attention Matching Model
  - 增强版representation-based model
  - 适合匹配文本较长的匹配任务



- Matching Matrix Model
  - 新匹配范式interaction-based model
  - 匹配更加充分、精细



应用场景

3. SimNet的应用

SimNet 技术在百度应用非常广泛，包括搜索、资讯推荐、广告、对话平台都在使用。

应用场景

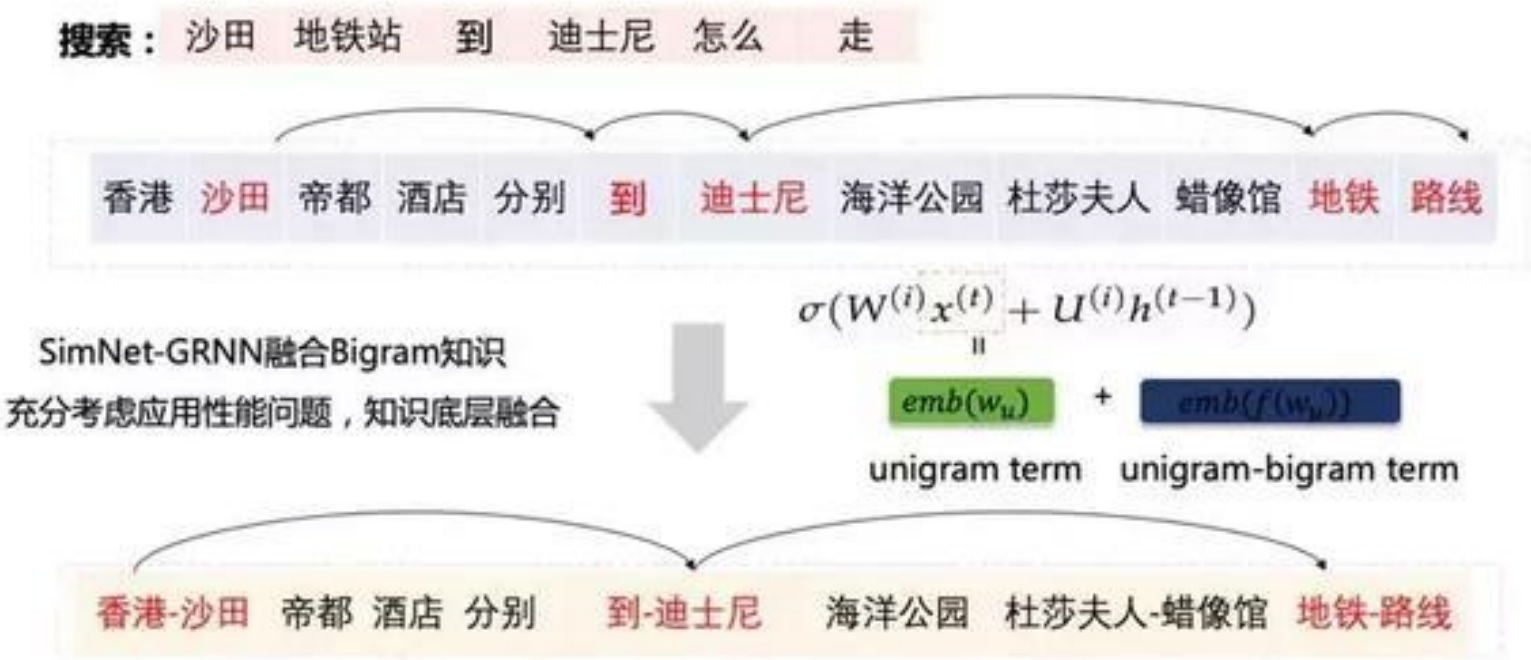
SimNet在百度搜索的发展

显著改善长冷query搜索效果，提升搜索智能化水平

SimNet系列特征在百度搜索系统中发挥重要作用

应用场景

知识融合：SimNet-RNN融合bigram粒度

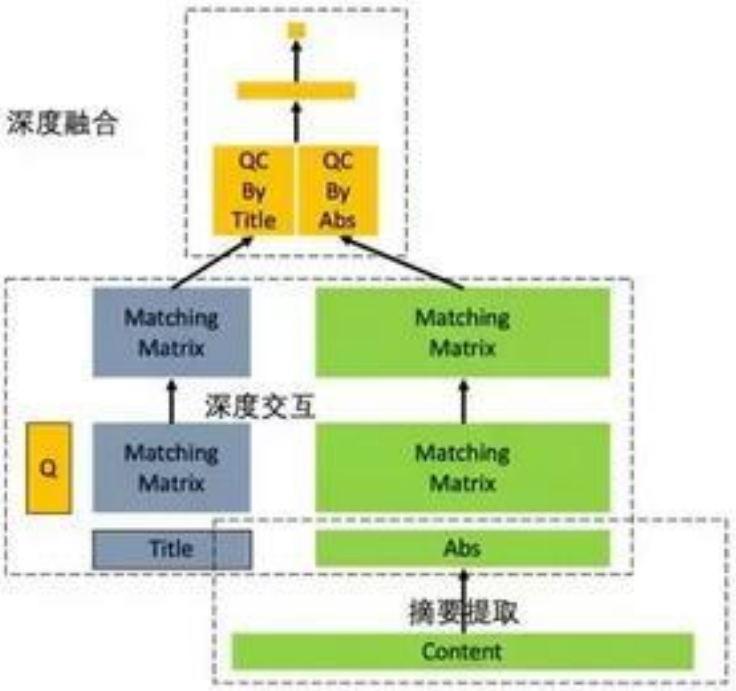


除了模型上的融合，百度把 Bigram 知识也融入了进去。尽管 RNN 已经很厉害了，但加入知识、模型还是会有很大的提升。

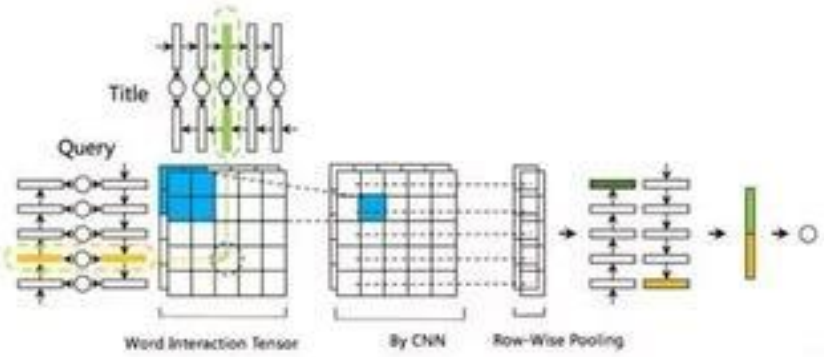
应用场景

4. 新模型：SimNet-QC-MM

新模型研发-基于SimNet-QC-MM的Query-Document匹配



- 算法：精确建模query中每个词被title和正文覆盖情况，并考虑文本匹配的不同层次的核心问题
- 架构：基于CPU和GPU异构计算架构
- 意义：业界前沿的Interaction-based范式模型工业级应用



应用场景

SimNet-QC-MM-Show Case

基线

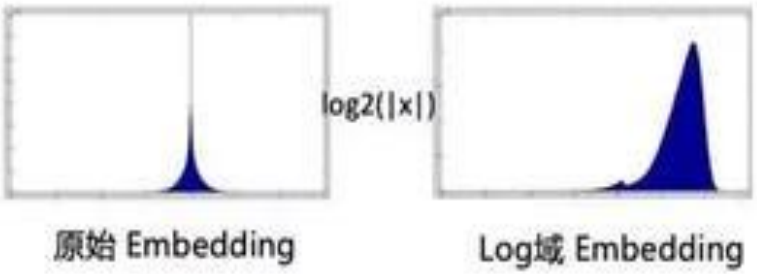
策略

应用场景

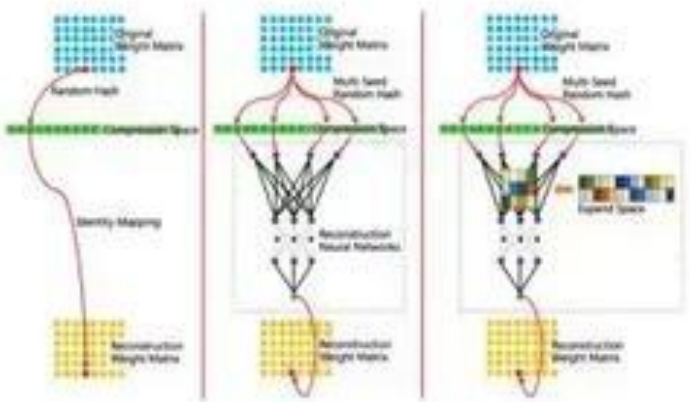
5. 语义模型压缩技术

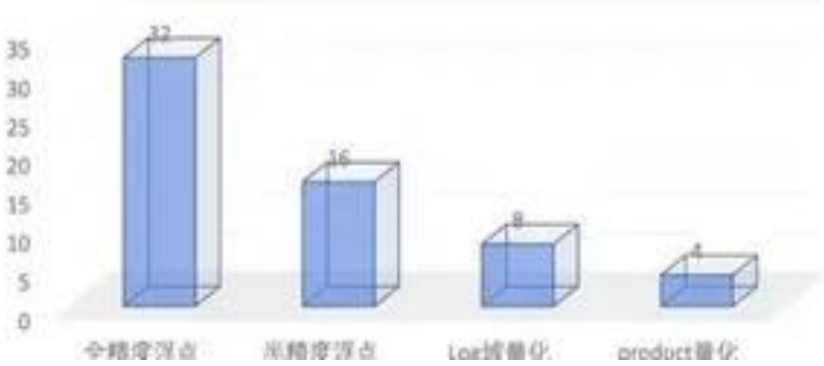
效果之外的优化：语义模型压缩技术

量化压缩技术



同源多种子随机哈希技术





Embedding一维仅需4bits

节省线上DNN匹配模型87.5%的内存消耗

应用场景

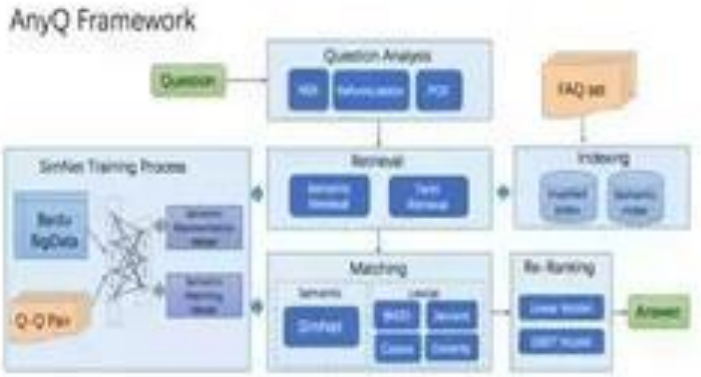
SimNet其他应用实例

网页搜索精准满足

问题-答案语义匹配

百度理财智能客服

问题-问题语义匹配



面向FAQ的问答系统 AnyQ
开源地址：https://github.com/baidu/anyQ

http://ai.baidu.com/

未来重点工作

接下来百度会在通用语义表示方面进一步研究与突破，除了如何充分的利用先验知识、多语言表示，面向生成、匹配等任务的表示，面向医疗、法律等领域的表示，多模态表示等都是百度的一些重点方向。

谢谢聆听