第六讲：文本大数据分析

文本大数据

- 文本是人类知识的重要载体，随着互联网的普及，文本大数据成为亟待解决的问题
  - 谷歌、百度、必应等主流搜索引擎索引的网页数量均超过百亿
- 文本分析是将非结构化的原始文本转化为计算机可识别处理的结构化信息的过程
- 本讲主要介绍大数据驱动的文本表达、文本匹配和文本生成三大文本大数据分析的核心任务
- 这些核心任务广泛应用在机器翻译、智能问答、信息检索、情感分析等诸多领域，是文本大数据分析的核心模块

文档

网页

目录

- 5.1 文本表达
  - 单词表达方法、句子表达方法
- 5.2 文本匹配
  - 基于规则的文本匹配、基于学习的文本匹配
- 5.3 文本生成
  - 文本生成任务、方法与评价方式

5.1 文本表达

文本表达

- 单词的表示
  - 局部性表示
  - 分布式表示
  - 评价方法
- 句子的表示
  - 传统表示方式
  - 分布式表示方式

文本表达

- 单词的表示
  - 局部性表示
  - 分布式表示
  - 评价方法
- 句子的表示
  - 传统表示方式
  - 分布式表示方式

单词的表示方法

- 要将自然语言的问题转化为计算机可以处理的问题，首先要找到可以将文本符号进行数字化的方法。文本表达的结果直接影响整个机器学习系统的性能

- 单词作为语言的基本单元，其表示学习一直是文本处理领域的核心问题

- 常用的表示方法可以分为局部性表示和分布式表示两种

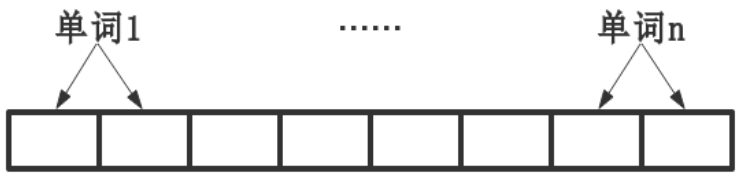
单词的表示方法

- 局部性表示
  - 独热表示
- 分布式表示
  - 横向组合关系
    - 隐性语义索引(Latent Semantic Indexing, LSI)
    - 概率隐性语义索引(Probabilistic Latent Semantic Indexing, PLSI)
    - 隐性狄利克雷分析(Latent Dirichlet Allocation, LDA)
  - 纵向聚合关系
    - 神经网络概率语言模型(Neural Prob. Language Model, NPLM)
    - 排序学习模型(C&W)
    - 上下文预测模型(Word2Vec)
    - 全局上下文模型(GloVe)

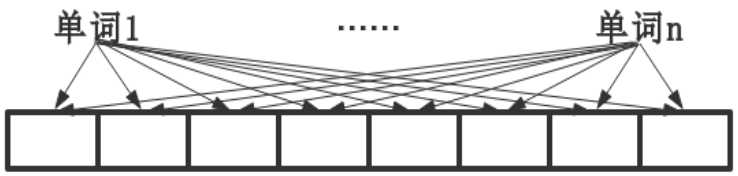
局部性表示 vs. 分布式表示

- 局部性表示(Local Representation)：在将单词表示为向量时，每个单词使用向量中独有且相邻的维度。在这种表示下，单词之间是相互独立的

- 分布式表示(Distributed Representation)：将单词映射到特征空间中，每个单词由刻画它的多个特征来高效表示；在形式上使用稠密实数向量（向量多于一个维度非0，通常为低维向量）来表示单词。分布式表示可以编码不同单词之间的语义关联



局部性表示



分布式表示

单词的局部性表示—独热表示

- 局部性表示中，如果仅仅使用一个维度，便称为独热表示 (One-hot Representation)。

- 独热表示将单词表示为一个长向量，向量的维度等于词表大小，仅有一维其值为1，其他维皆为0，如：

车票	[0,0,...,0,**1**,0,0,0,...,0,0]
硕士	[0,0,...,0,0,**1**,0,0,...,0,0]
本科生	[0,0,...,0,0,0,**1**,0,...,0,0]

- 如果使用紧凑方式存储独热表示，只需给每一个单词分配一个数字ID，非常简洁。比如，在上例中，“车票”ID记为10，“硕士”ID记为11，“本科生”ID记为12，在编程中可使用Hash表给每一个单词分配一个编号

独热表示的优势和不足

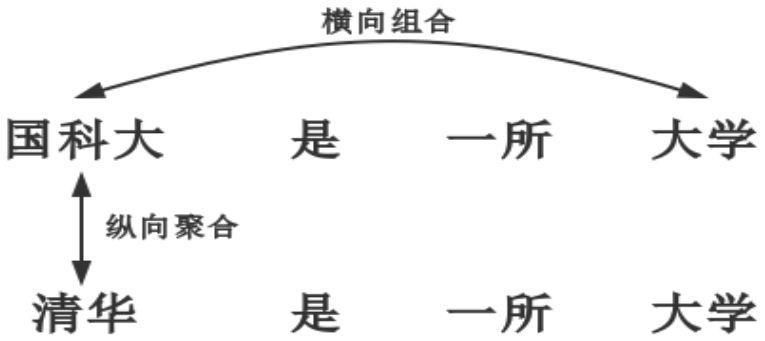
- 独热表示假设所有单词都是相互独立的，在其向量空间中所有的词向量都是正交的，因此其具有很强的判别能力。配合支持向量机SVM、条件随机场CRF等学习算法，在众多问题上均可取得良好的结果

- 同样，由于其词向量间的正交性，无论使用余弦相似度或是欧氏距离度量出的单词间语义相似度均相等，也就是说其丢失了单词之间的语义相关信息

- 此外，独热表示在实际应用时经常会面临着维度灾难问题

单词的分布式表示

- 分布式表示方法都基于分布语义假设(Distributional Hypothesis)，即单词的语义来自其上下文(context)。因此，
- 所有的分布式表示模型都利用某种上下文的统计信息来学习单词的分布式表示，使用不同的上下文使得模型建模了单词间的不同关系，可分为横向组合关系模型(Syntagmatic Models)和纵向聚合关系模型(Paradigmatic Models)



单词的分布式表示

- 横向组合关系指两个单词可以同时出现在一段文本区域中（如同一个句子），强调它们可以进行组合，在其中往往起到不同的语法作用。如下图中“国科大”和“大学”即存在横向组合关系。对横向组合关系建模的模型通常使用文档作为上下文

- 纵向聚合关系指的是纵向的可替换的关系，强调的是相似的词可以拥有相似的上下文（context）但通常不同时出现。如上图中的“国科大”和“清华”。纵向聚合关系通常使用当前单词周边的单词作为其上下文

单词的分布式表示

- 文档1: I love playing football.
- 文档2: I love playing tennis.
- 文档3: You love playing football.

	$doc1$	$doc2$	$doc3$
I	1	1	0
$love$	1	1	1
$playing$	1	1	1
$football$	1	0	1
$tennis$	0	1	0
you	0	0	1

- 可以观察到：
  - love 和 playing 这两个较强组合关系的词的词表示是相似的；
  - football 和 tennis这两个具有较强替换关系的词的表示是不相似的。

横向组合关系

- 常用的横向组合关系有:
  - 隐性语义索引(Latent Semantic Indexing, LSI)
  - 概率隐性语义索引(Probabilistic Latent Semantic Indexing, PLSI)
  - 隐性狄利克雷分析(Latent Dirichlet Allocation, LDA)

基础知识：低秩逼近矩阵

- 给定$m \times n$的矩阵C及正整数k，寻找一个秩不高于k的$m \times n$的矩阵C_k，使其与原矩阵C之间的差异最小，即$X = C - C_k$的F-范数最小。显然，当k等于C的秩r时，$C_k = C$，此时两个矩阵差值的F-范数为0；当$k \ll r$时，称C_k是C的低秩逼近矩阵(low-rank approximation matrix)

- F-范数：矩阵X的F-范数(Frobenius Norm，弗罗宾尼奇范数)为

$$\|X\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}^2}$$

隐性语义索引(LSI)

- LSI是指通过对词项-文档矩阵C (每行代表一个词项，每列代表一篇文档；元素c_{ij}表示第i个单词在第j篇文档中出现的次数)进行矩阵分解(具体采用SVD分解)来找到它的某个低秩逼近，进而利用得到的低秩逼近形成对词项和文档的新的表示
- 给定$m \times n$的词项-文档矩阵C和正整数k，对C进行LSI的过程如下：
  - 1. 将矩阵C分解为$C = U_{m\times m}\Sigma_{m\times n}V^T_{n\times n}$；
  - 2. 保持Σ对角线上前k大个奇异值不变，其余元素置为0，得到Σ_k；
  - 3. 计算$C_k = U_{m\times k}\Sigma_{k\times k}V^T_{n\times k}$作为$C$的低秩逼近。一般而言，不同于非负整数构成的较为稀疏的矩阵C，C_k是一个实数构成的稠密矩阵；
    - 3.1. 矩阵$U_{m\times k}$的每一行是相应词项的向量表示，每一维代表该词项在主题空间中的该主题上的映射；
    - 3.2. 矩阵$V_{n\times k}$的每一行是相应文档的向量表示，每一维代表该文档在主题空间中的该主题上的映射。对于给定的文档；

隐性语义索引(LSI)

- LSI仅保留了矩阵C中最大的k个奇异值，相当于将原有的词项-文档矩阵从r维降至k维，每一个奇异值可以理解为对应一个“主题”维度，其值的大小表示与这一“主题”的相关程度，因此LSI也是一种主题模型(Topic Model)

- 保持较大的奇异值而将较小的奇异值置为0可以保留文档集中较为重要的信息，并且忽视不重要的细节，从而解决多词一义(synonymy)和语义关联的问题

- LSI得到的不是一个概率模型，缺乏统计基础，结果难以直观解释。此外，很难选择合适的k值，而k的选取对结果的影响非常大

概率隐性语义索引(PLSI)

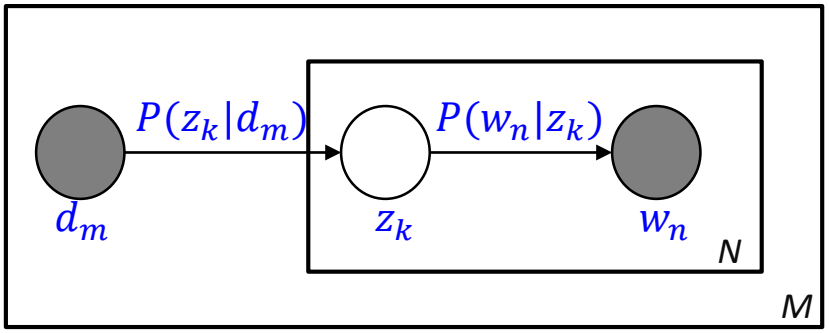
- 鉴于LSI的不可解释性，在其提出后许多学者尝试找到比较严谨的数学方法，一种途径是通过引入概率图模型得到概率化的解释

- Hofmann于1999年提出的PLSI(Probability Latent Semantic Indexing)文本模型便是在这一方向上取得的重要学术成果

- PLSI也是一种主题模型，不同于LSI中启发式的选择秩k进行低秩逼近的做法，PLSI定义了一个概率模型，并对模型中使用的变量及其对应的概率分布和条件概率分布给出了明确的解释

概率隐性语义索引(PLSI)

- PLSI假设在M篇文档和N个词项之间存在k个隐藏的主题，但我们无法对其进行观测。按概率$P(d_m)$选择一篇文档$d_m \in D$，$P(z_k|d_m)$表示在d_m中主题$z_k \in Z$出现的概率，$P(w_n|z_k)$表示在主题z_k中词项$w_n \in W$出现的概率。那么，我们可以将文档到词项的过程看做一个有向图，如下图所示：



- ✓ 方框表示集合，右下角的字母表示集合的元素数目，M表示文档数，N表示词项数；
- ✓ 灰色圆圈表示可观测变量，白色圆圈表示隐变量；

概率隐性语义索引(PLSI)

- 从概率图模型可知道，可观测变量是d_m和w_n，隐变量是z_k，可以写出三者的联合概率分布为：

$$P(d_m, z_k, w_n) = P(d_m)P(z_k|d_m)P(w_n|z_k)$$

- 在上式中对主题z_k进行概率边缘化，即可得到可观测数据文档-词项(d_m, w_n)的联合概率分布为：

$$P(d_m, w_n) = P(d_m)\sum_k P(z_k|d_m)P(w_n|z_k)$$

这里，$P(z_k|d_m)$与$P(w_n|z_k)$是要估计的参数，假设均服从多项式分布

概率隐性语义索引(PLSI)

- PLSI采用极大似然法对参数进行估计。对于一个包含M篇文档的集合来说，能观察到的数据就是(d_m, w_n)这样的共现对，其似然函数定义如下：

$$l(\theta) = \prod_{m=1}^{M} \prod_{n=1}^{N} P(d_m, w_n)^{n(d_m, w_n)}$$

其中$n(d_m, w_n)$表示d_m和w_n共现的次数，即文档d_m中w_n出现的次数

概率隐性语义索引(PLSI)

- 对似然函数取对数并展开得到

$$L(\theta) = ln\, l(\theta) = ln \prod_{m=1}^{M} \prod_{n=1}^{N} P(d_m, w_n)^{n(d_m, w_n)} = \sum_{m=1}^{M} \sum_{n=1}^{N} n(d_m, w_n)\, ln\, P(d_m, w_n)$$

$$= \sum_{m=1}^{M} \sum_{n=1}^{N} n(d_m, w_n)\, ln \left(P(d_m) \sum_{\boldsymbol{k=1}}^{\boldsymbol{K}} (P(z_k|d_m) P(w_n|z_k)) \right)$$

$$= \sum_{m=1}^{M} \sum_{n=1}^{N} n(d_m, w_n)\, ln(P(d_m)) + \sum_{m=1}^{M} \sum_{n=1}^{N} n(d_m, w_n)\, ln \left(\sum_{\boldsymbol{k=1}}^{\boldsymbol{K}} (P(z_k|d_m) P(w_n|z_k)) \right)$$

- 在给定文档集时， $P(d_m)$是已知，因此式中第一项为常数项。去掉之后，对数似然函数变为

$$L(\theta) = \sum_{m=1}^{M} \sum_{n=1}^{N} n(d_m, w_n)\, ln \left(\sum_{\boldsymbol{k=1}}^{\boldsymbol{K}} (P(z_k|d_m) P(w_n|z_k)) \right)$$

- 由于需要对数运算，无法通过导数对函数求极值，故使用EM算法求解

隐性狄利克雷分析(LDA)

- LDA是一种应用更为广泛的主题模型。LDA模拟的文档生成过程如下：
  - 1. 假设要生成的文档$d_m \in D$的长度为N_m，为其选定一个主题分布θ_m，并使其先验分布服从Dirichlet分布$\theta_m \sim Dirichlet(\alpha)$；
  - 2. 在生成文档d_m中第n个词项d_{mn}时，首先在分布θ_m下生成主题z_n，即$z_n \sim multinomial(\theta_m)$；

隐性狄利克雷分析(LDA)

- 3. 词项的概率分布服从Dirichlet分布，即对任一主题z_n，其中的词项分布服从$\beta_{z_n} \sim Dirichlet(\eta)$，此时依多项式分布$P(d_{mn}|\beta_{z_n}) = P(d_{mn}|z_n, \beta)$生成$d_{mn}$。因此，我们可以得到$\theta_m, z_n, d_m$的联合概率分布为

$$P(\theta_m, z_n, d_m|\alpha, \beta) = P(\theta_m|\alpha)\prod_{n=1}^{N_m} P(z_n|\theta_m)P(d_{mn}|z_n, \beta)$$

- 为求得生成文档d_m的概率，在上式中对θ_m (连续变量)求积分，对z_n(离散变量)求和，即

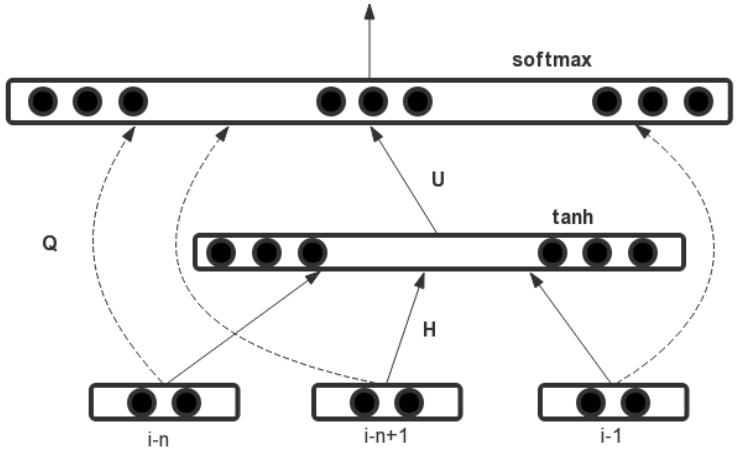
$$P(d_m|\alpha, \beta) = \int P(\theta|\alpha)\prod_{n=1}^{N_m}\sum_{z_n} P(z_n|\theta)P(d_{mn}|z_n, \beta)\, d\theta$$

- 生成整个语料库的概率等于生成语料库中各篇文档的乘积，即

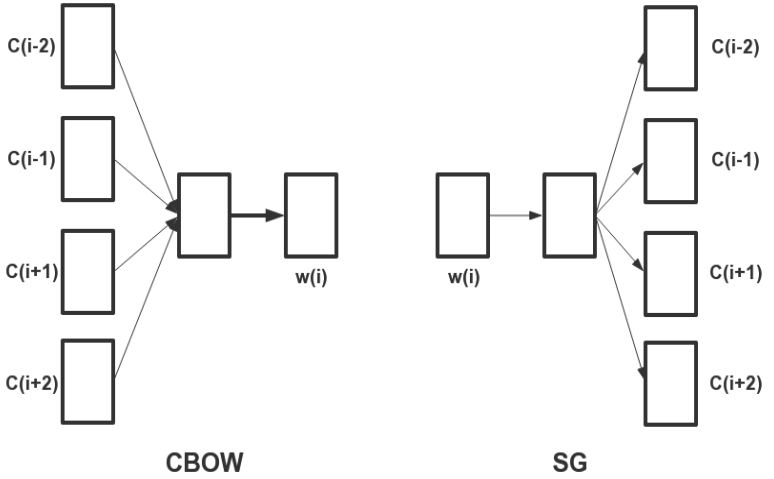
$$P(D|\alpha, \beta) = \prod_{m=1}^{M} P(d_m|\alpha, \beta) = \prod_{m=1}^{M}\int P(\theta|\alpha)\prod_{n=1}^{N_m}\sum_{z_n} P(z_n|\theta)P(d_{mn}|z_n, \beta)\, d\theta$$

纵向聚合关系

- 常用的纵向聚合算法有
  - 神经网络概率语言模型(Neural Probabilistic Language Model, NPLM)
  - 排序学习模型(C&W)
  - 上下文预测模型(Word2Vec)
  - 全局上下文模型(GloVe)等



NPLM模型



CBOW和SG模型框架

神经网络概率语言模型(NPLM)

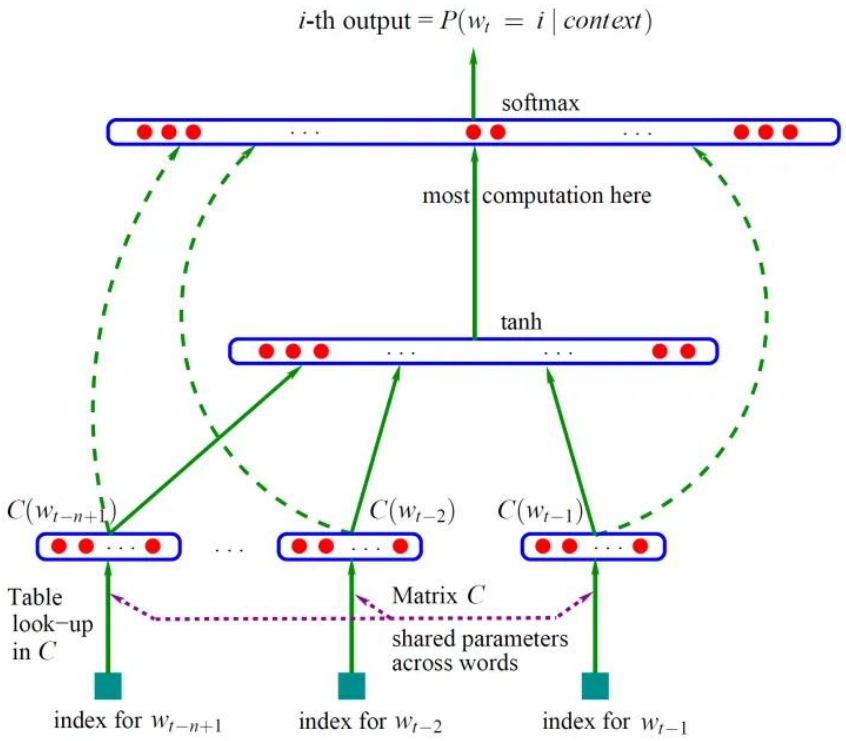
- 与矩阵分解模型不同，Bengio等人将神经网络应用于单词表示，提出了神经网络概率语言模型(Neural Probabilistic Language Model)

- NPLM通过训练一个语言模型得到单词表示，本质上也是一种n-gram模型

- NPLM使用一个三层神经网络，将当前单词w_t的前n个单词$[w_{t-n+1}, w_{t-n}, \cdots, w_{t-1}]$作为上文输入网络，经中间隐层转换后，利用softmax层作为输出层，给出在前述上文下任何一个单词出现的概率

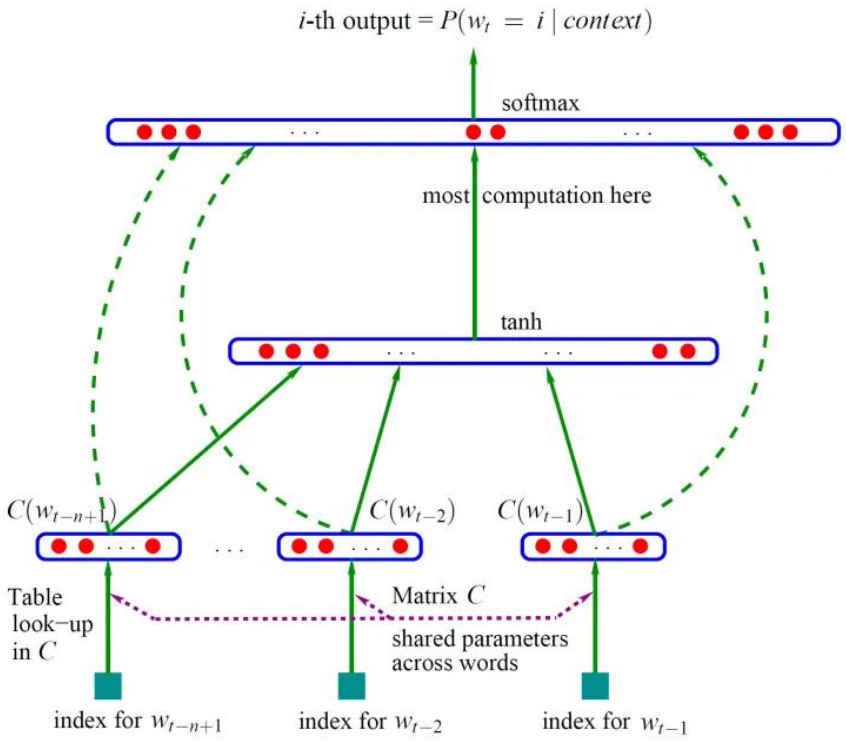
神经网络概率语言模型(NPLM)

- 网络第一层（输入层）的输入向量x由前n个单词的词向量$C(w_i)$首尾相连拼接得到；若每个单词向量长度为k，则向量x长度为kn；



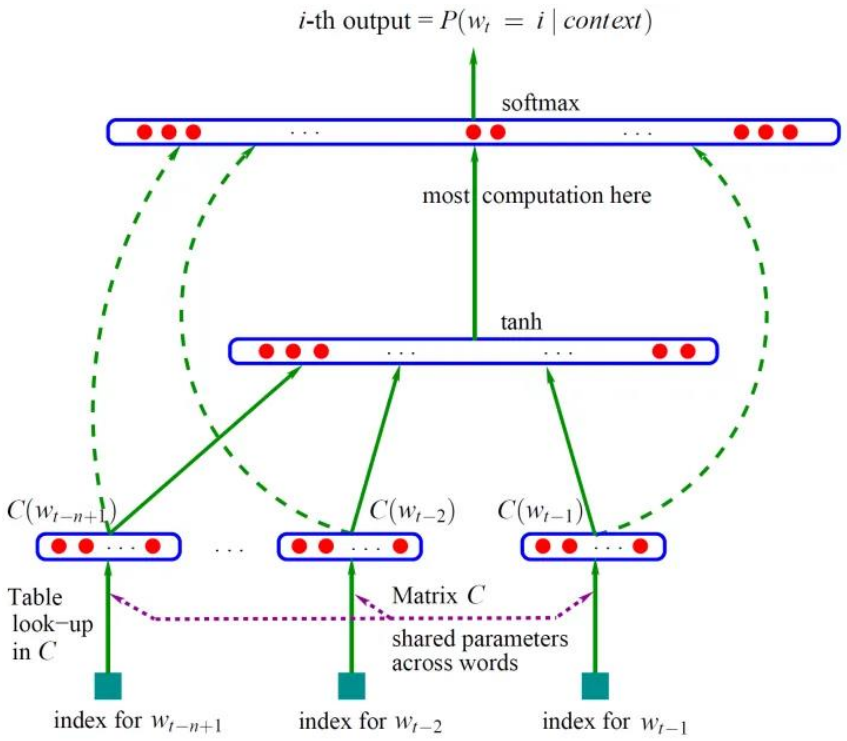
神经网络概率语言模型(NPLM)

- 网络第二层（隐藏层）是个全连接层，输入是x的线性变换$Hx + d$，其中H是权重矩阵，d为偏移量；经过激活函数$tanh$后输至下一层；



神经网络概率语言模型(NPLM)

- 网络第三层（输出层）共包含字典大小|V|个节点，节点y_j表示在给定上文之下单词w_j出现的概率(未归一化)，并通过softmax函数将输出y归一化。具体表示为：$y = U\ tanh(Hx + d) + Wx + b$



$|V|$

U是个大小为$|V| \times h$的矩阵，是隐藏层到输出层的参数，整个模型的多数计算集中在U和隐藏层的矩阵乘法中

W是个$|V| \times n$的矩阵，是将输入层的数据结果也放到输出层进行计算的线性变换，称为直连边。如果不需要直连边的话，将W置为0即可

神经网络概率语言模型(NPLM)

- NPLM的核心思想在于，对于同一个单词，其上文中出现的单词总是相似的，即相似的输出需要相似的输入；

- 这一模型避免了传统n-gram模型中复杂的平滑算法。但其在输出层计算时受词表大小的影响，导致其学习和推断过程都异常耗时；

排序学习模型(C&W)

- 排序学习模型(C&W) 由Collobert & Weston于2008年提出
- 相比NPLM模型，主要有两点改进：
  - 1. C&W同时使用了单词的上下文，这成为其后学习单词表示的基本做法；
  - 2. C&W对单词序列打分使用了排序损失函数，而不是基于概率的极大似然估计，其损失函数定义为

$$\max[0,1 - s(w,c) + s(w',c)]$$

其中，c 表示单词w 的上下文(context)，w'表示将当前上下文中的单词w替换为一个随机采样出的无关单词w'（负样例）；s为打分函数，分数越高表明该段文本越合理

- 显然，在大多数情况下将普通短语中的特定单词随机替换为其他单词时，得到的都是不正确的短语。因此模型的目标是，尽量使正确短语的得分 比 随机替换后的短语的得分高1

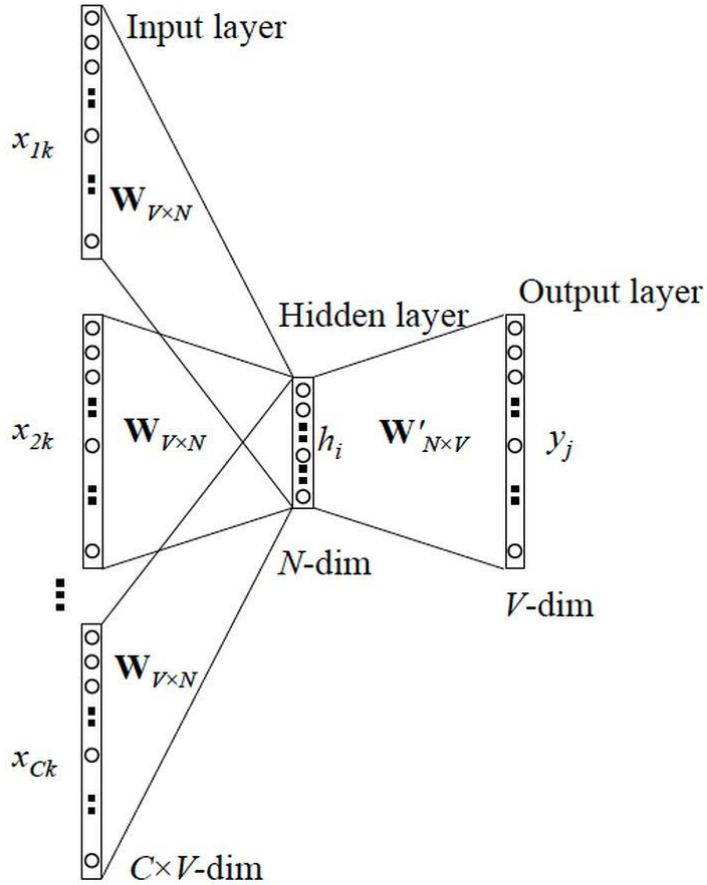
上下文预测模型(Word2Vec)

- 为了更好的利用单词的上下文，Mikolov等人提出了两个简单的神经网络模型CBOW(Continuous Bag of Word)和SG(Skip Gram)进行学习；

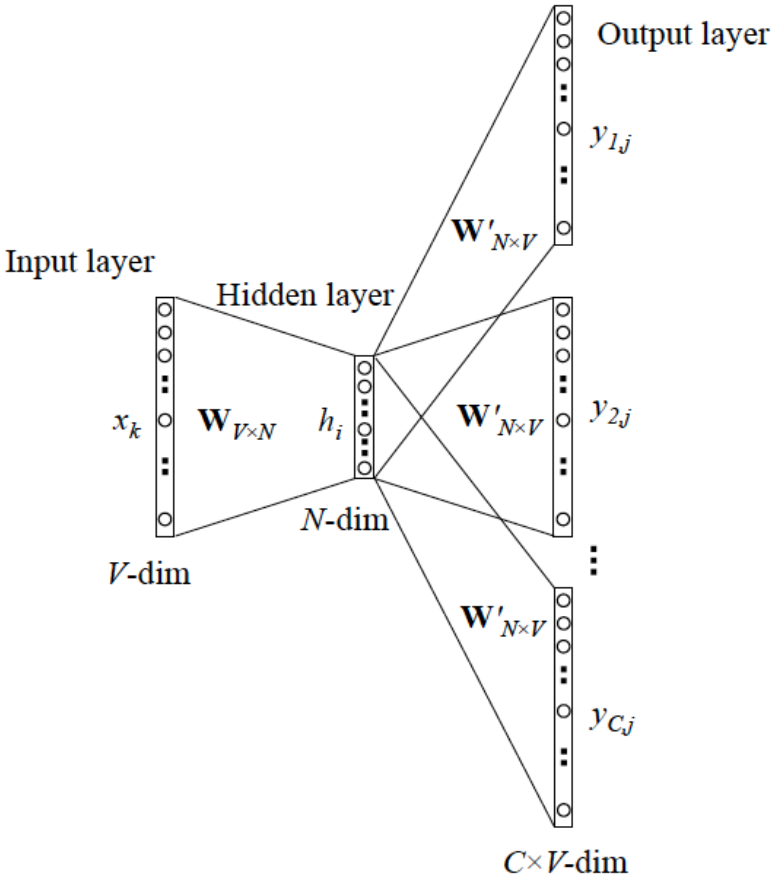
- 相比NPLM模型，CBOW模型去除了中间的非线性隐层，将单词w_i上下文的表示经过求和或平均等计算后，用得到的结果h_i直接预测单词w_i；而SG模型则使用单词w_i预测其上下文中的每一个单词。

- 以短语 “国科大 是 位于 北京 的 大学”为例：
  - CBOW基于上下文“国科大 是 位于 的 大学”预测中心词“北京”
  - SG基于中心词“北京”预测上下文“国科大 是 位于 的 大学”

上下文预测模型(Word2Vec)



CBOW模型



SG模型

全局上下文模型(GloVe)

- 在Word2Vec算法中，主要使用单词的上下文信息获得单词的表示。GloVe模型是一种对词-词共现矩阵进行分解而得到词表示的模型。

	I	$love$	$playing$	$football$	$tennis$	you
I	0	2	0	0	0	0
$love$	2	0	3	0	0	1
$playing$	0	3	0	2	1	0
$football$	0	0	2	0	0	0
$tennis$	0	0	1	0	0	0
you	0	1	0	0	0	0

football和tennis这两个较强替换关系的词的词表示是相似的，而love和playing这两个较强组合关系的词的词表示是不相似的

- 矩阵中的元素值表示的是，以行指标所代表的词作为中心词的窗口内，列指标所代表的词出现的次数，说的简洁一点就是两个词在窗口内的共现次数。
- 上面这个矩阵中，所取的窗口大小为1。比如，以love作为中心词、窗口大小为1的窗口就是“I, love, playing”、“I, love, playing”、“You, love, playing”，那么在窗口内love和playing共现了3次，所以该矩阵的第二行第三列就是3。

全局上下文模型(GloVe)

- 相比Word2Vec算法，GloVe算法利用单词的共现信息，将全文的统计信息与句子的信息相结合，以期得到单词在语义和语句上更好的表达

- 令单词w_i出现的次数为X_i，单词w_i与w_k同时出现的次数为X_{ik}，则在单词w_i出现的情况下单词w_k出现的条件概率为$P(w_k|w_i) = \frac{X_{ik}}{X_i}$

- 研究发现，条件概率的比值$ratio_{i,j,k} = \frac{P(w_k|w_i)}{P(w_k|w_j)}$存在如下规律：

$ratio_{i,j,k}$	单词 j, k 相关	单词 j, k 不相关
单词 i, k 相关	趋于1	很大
单词 i, k 不相关	很小	趋于1

全局上下文模型(GloVe)

- 基于这一观察，对每个单词对相应的向量定义如下的软约束

$$v_i^T v_j + b_i + b_j = log\, X_{ij}$$

其中，v_i和v_j是单词w_i和w_j的向量，b_i和b_j为对应w_i和w_j的偏差，X_{ij}为权重项，正比于单词w_i和w_j共现的次数

- 进一步，定义如下的目标函数

$$J = \sum_{i,j=1}^{N} f(X_{ij})(v_i^T v_j + b_i + b_j - log\, X_{ij})^2$$

其中，

$$f(X_{ij}) = \begin{cases} \left(\frac{X_{ij}}{XMAX}\right)^{\alpha} & if\, X_{ij} < XMAX \\ 1 & \text{Otherwise} \end{cases}$$

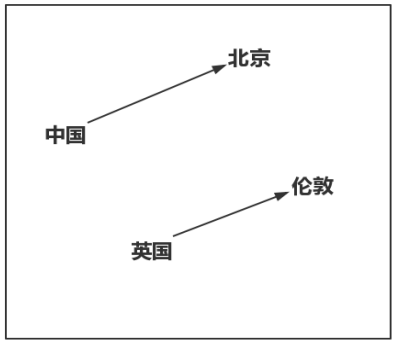
$f()$防止只从共现率很高的单词对中学习

单词表示的评价方式

- 相似度评价
  - 评价单词表示效果最基础的方法是衡量单词间的相似度。通过学习得到单词的向量表示后，利用余弦相似度或欧式距离等方式估计单词对的相似度
  - 这其中使用最广泛的数据集是WordSim353，这一数据集包含了353个单词对。除此之外，还有两个常用数据集RareWord(RW)和SimLex-999(SL-999)

- 单词类比
  - 单词表示的质量也可以通过单词类比任务来评价，其主要数据集来自于Mikolov等人的工作，这一数据集包含了大量类似于“a之于b正如c之于?”的问题，如“北京对于中国正如伦敦对于？”，其中缺失的单词需要寻找整个词汇表来回答

- 特征
  - 单词表示也可作为特征用于具体任务中进行评价。如将单词向量表示作为额外的特征加入现有机器学习系统中，或将其作为唯一的特征独立解决实际问题



文本表达

- 单词的表示
  - 局部性表示
  - 分布式表示
  - 评价方法
- 句子的表示
  - 传统表示方式
  - 分布式表示方式

句子的表示方法

- 传统方法
  - 词集模型
  - 词袋模型
  - TF-IDF表示
- 分布式表示方法
  - 主题模型
  - 基于单词分布式表示组合的表示方法
  - 由原始语料直接学习的表示方法

词集模型

- 词集模型(Set of Words)是一个由单词构成的集合，忽略文本中的词序与语法，只记录单词是否出现的情况。按照集合的定义，集合中的每个元素只有1个，因此词集中的每个单词都只有一个

- 词集模型是最简单的句子表示方法，因为其数值非0即1，可以很好地支持位运算，在检索应用场景中能够执行快速的查询处理

- 示例:
  - 句子1：“我 来自 中国 科学院 大学”
  - 句子2：“他 在 中国 科学院 计算所 学习”
  - 单词表vocab={我:0, 来自:1, 中国:2, 科学院:3, 大学:4, 他:5, 在:6, 计算所:7, 学习:8}
  - 句子1的词集模型向量表示：(1, 1, 1, 1, 1, 0, 0, 0, 0)
  - 句子2的词集模型向量表示：(0, 0, 1, 1, 0, 1, 1, 1, 1)

词袋模型

- 词袋模型(Bag of Words)是在词集模型的基础上，考虑了单词出现的次数，因此，在词袋模型中，句子向量中每个单词对应的位置上记录的是该单词出现的次数，这也体现了各个单词在该句子中的重要程度

- 示例：
  - 句子：“我 来自 中国 科学院 大学，他 在 中国 科学院 计算所 学习”
  - 单词表vocab={我:0， 来自:1， 中国:2， 科学院:3， 大学:4， 他:5， 在:6， 计算所:7， 学习:8}
  - 词袋模型向量表示：(1， 1， 2， 2， 1， 1， 1， 1， 1)

TF-IDF模型

- TF-IDF（Term Frequency－Inverse Document Frequency)是一种用于信息检索与数据挖掘的常用加权技术。TF是词频(Term Frequency)，IDF是逆向文档频率(Inverse Document Frequency)

- TF-IDF的主要思想是：如果某个词或短语在一篇文章中出现的频率TF高，并且在其他文章中很少出现，则认为该词或者短语具有很好的类别区分能力，适合用来分类

- TF计算公式如下：

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

其中，$n_{i,j}$是单词t_i在文档d_j中的出现次数，分母是文档d_j中所有单词的出现次数之和。

TF-IDF模型

- IDF是对一个词语普遍重要性的度量。某一特定词语的IDF，可由总文档数目除以包含该词语之文档的数目，再将得到的商取对数得到。具体地，单词t_i的IDF计算如下：

$$idf_i = log \frac{|D|}{|\{j: t_i \in d_j\}| + 1}$$

其中|D|表示语料库中的文档总数

- TF-IDF：

$$\mathrm{tfidf}_{i,j} = \mathrm{tf}_{i,j} \times \mathrm{idf}_i$$

可以看出，某一特定文档内的高词语频率，以及该词语在整个文档集合中的低文档频率，可以产生出高权重的TF-IDF。因此，TF-IDF倾向于过滤掉常见的词语，保留重要的词语

- 对文档d_j的词集模型在对应每个单词t_i的维度赋予该单词的$\mathrm{tfidf}_{i,j}$值，就得到该文档的TF-IDF表示

主题模型

- 主题模型将句子或文档表示为主题分布
- 通常两篇文档是否相关往往不只取决于字面上的词语重复，还取决于文字背后的语义关联。对语义关联的挖掘，可以更精准的刻画文档
- 主题模型是对文字所隐含主题进行建模的方法。它克服了传统计算文档相似度方法的缺点，能够自动寻找出文字间的语义主题。主要的主题模型包括LSI、PLSI和LDA方法，这里不再重复介绍
- 通过主题模型对语料集的分析，能够同时得到单词和文档的分布式表示，其中把单个句子按文档来处理，即可得到句子的分布式表示

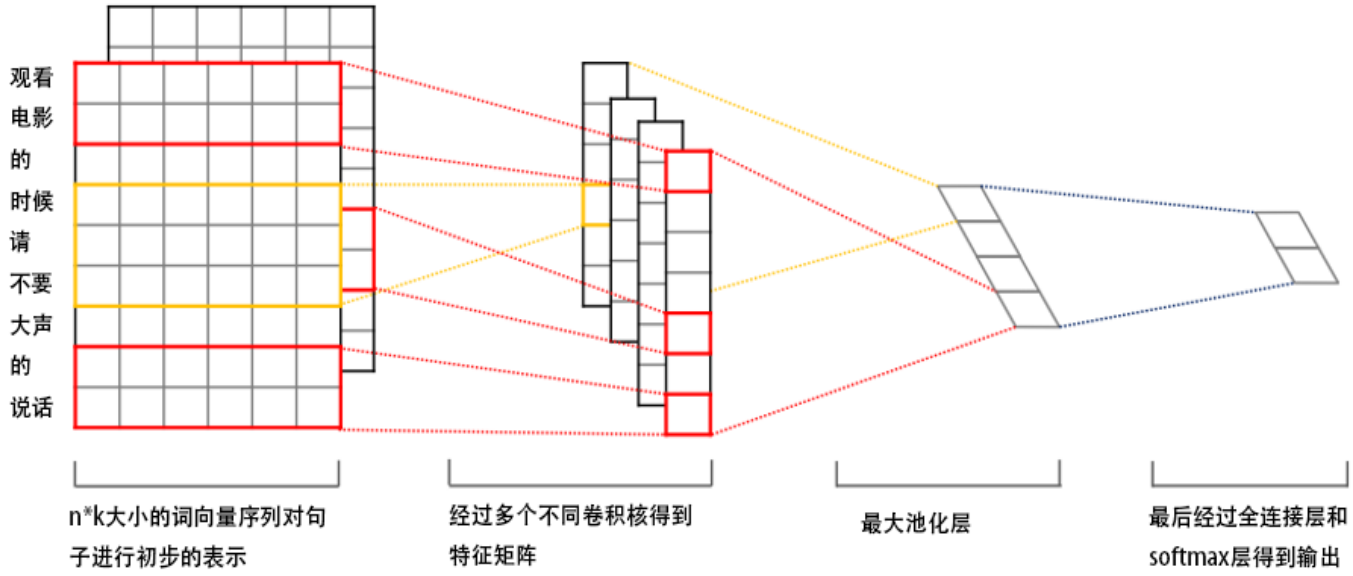
基于单词分布式表示组合的表示方法

- 句子的分布式表示建立在单词的分布式表示的基础之上。其主要思想是：针对具体任务，对单词的分布式表示进行组合或选择等，最终得到一个向量作为句子的分布式表示，这是一个特征组合、提取的过程。

- 常用的方法
  - 基于卷积神经网络(CNN)的分布式表示
  - 基于循环神经网络(RNN)的分布式表示
  - 基于递归神经网络(RecNN)的分布式表示
  - 基于DAN(Deep Averaging Networks)的分布式表示

基于单词分布式表示组合的表示方法

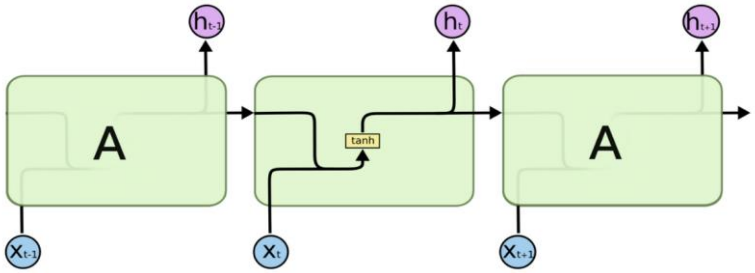
- 基于卷积神经网络(CNN)的分布式表示



特征向量 $c_i = f(w \cdot x_{i:i+h-1} + b)$

基于单词分布式表示组合的表示方法

- 基于循环神经网络(RNN)的分布式表示



$$\mathrm{h}^{(t)} = \sigma\left(z^{(t)}\right) = \sigma\left(Ux^{(t)} + Wh^{(t-1)} + b\right)$$

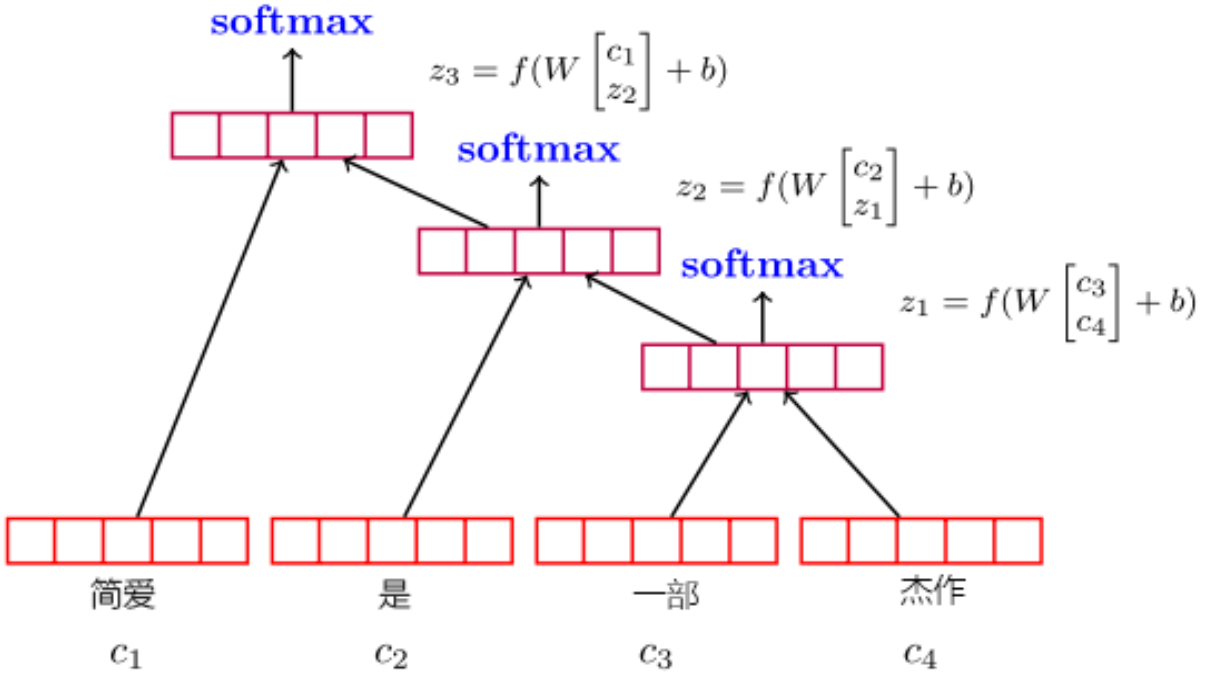
σ 为激活函数，一般为tanh；b为线性偏差

通常有以下5种用RNN表示句子特征的策略：

- 直接使用RNN的最后一个单元输出向量h(t)作为句子特征；
- 使用双向RNN的两个方向的输出向量的拼接或均值作为句子特征；
- 将所有RNN单元的输出向量的均值pooling或者max-pooling作为句子特征；
- RNN+Attention，即对每一个时间点的特征赋予权重，在针对后续具体任务的计算中，不同时间点的信息起到的作用程度不同；
- RCNN，将所有RNN单元的输出向量经过一层CNN和max-pooling得到句子表示。

基于单词分布式表示组合的表示方法

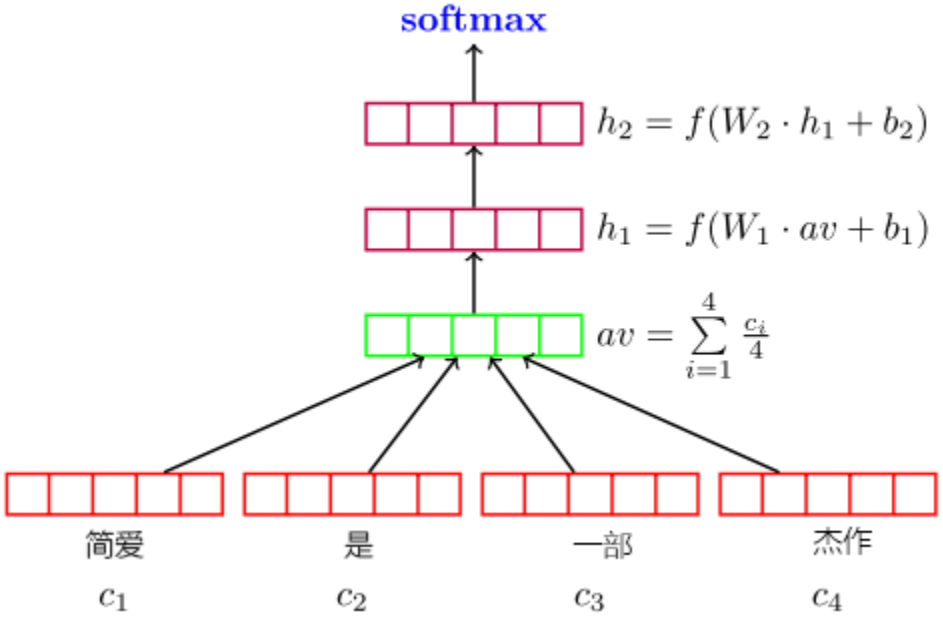
- 基于递归神经网络(RecNN)的分布式表示



- 分析句子的过程就同语法树一样，自底向上，遵循语法规则，最终根节点得到的向量z_n即为该句子的表示向量
- 实现效果依赖于输入文本的语法树，需要更多的训练时间

基于单词分布式表示组合的表示方法

- 基于DAN(Deep Averaging Networks)的分布式表示



这里的av是最简单的无序模型Neural Bag-of-Words Models (NBOW model)

- 该结构可以理解为在分类层前增加了隐藏层对句子进行特征提取，得到深层次的句子表示
- 也可以理解为仍然由词向量的平均来进行句子的表示，采用多层网络来进行具体的分类任务

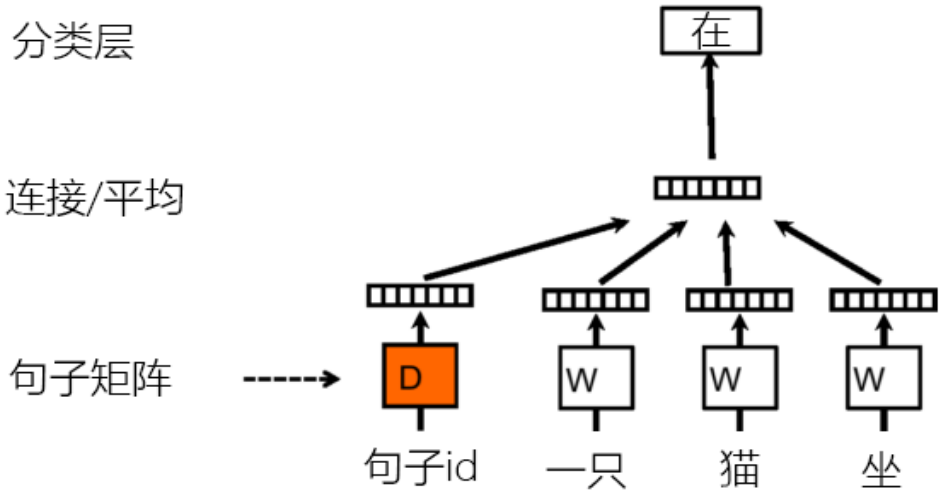
基于原始语料生成的表示方法

- 基于单词分布式表示组合的句子表示方法其效果依赖于单词分布式表示的准确程度，而且在大多数情况下，单词的分布式表示也是由word2vec或GloVe等方法生成，那么直接生成句子的分布式表示理论上更直接，效果上也不会受单词的分布式表示所影响

- 借鉴word2vec的思想，Quoc Le等人提出paragraph2vec的两种模型：PV-DM和PV-DBOW

基于原始语料生成的表示方法

- PV-DM(Distributed Memory Model of Paragraph Vectors)模型



- 模型依据前k个单词和当前句子id来预测下一个单词，相当于每次在预测单词的概率时，都利用了整个句子的语义。预测任务看做是一个分类任务，预测函数如下：

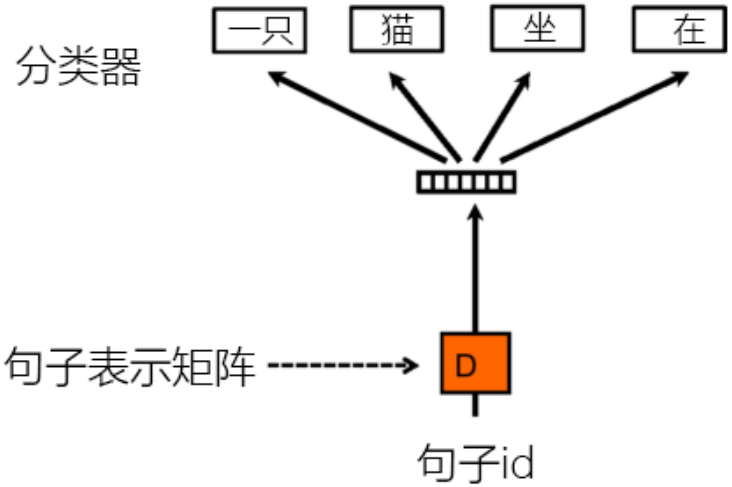
$$p(w_t|d_p, w_{t-k}, \ldots, w_{t-1}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_i}}$$

其中，w_t是第t个单词的词向量，d_p是当前句子p的句向量，y_{w_t}是每个单词w_t未标准化的输出分数

- 采用随机梯度下降法最大化文档集C的似然函数。当训练收敛后，输出矩阵W和D得到单词和句子的分布式表示

基于原始语料生成的表示方法

- PV-DBOW(Distributed Bag of Words)模型



  - PV-DBOW模型输入句子id对应的句向量paragraph vector，输出是该句子中随机采样的词。这种方法需要存储的数据更少，相比于PV-DM模型，只需存储输出层的参数，也不需要保存单词的向量

  - 两种Paragraph2Vec模型都以无监督学习方法来生成句子的分布式表示，以句子本身的语义来推断其上下文内容或以上下文内容推断句子的语义，同样的思想也可以运用在其他连续序列数据上的特征表示

5.2 文本匹配

匹配无处不在

- 生活中匹配的例子





文本中的匹配问题

文本匹配是自然语言理解的一个核心问题，许多文本处理的问题可以抽象成文本匹配的问题

信息检索	查询项 ⟷	文档
问答系统	问题 ⟷	答案
对话问题	前文 ⟷	回复
复述问题	原句 ⟷	改写
机器翻译	中文 ⟷	英文

搜索引擎

Google

bing

Bai du 百度

问答系统

YAHOO! ANSWERS

知乎

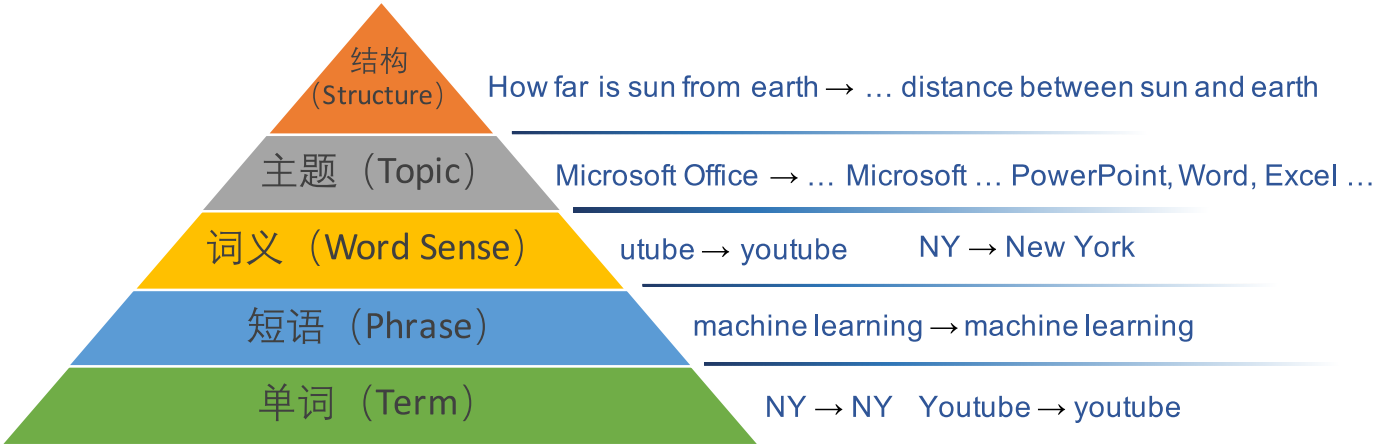
Bai du 知道

智能助手

Sir
i

微软小冰

文本匹配的挑战



- 挑战：
  - 词语匹配的多元性　荷花 = 芙蓉　　苹果 = 公司 or 水果
  - 短语匹配的结构性　机器学习 --- 学习机器
  - 文本匹配的层次性　词 - 短语 - 句子 - 段落 - 篇章

文本匹配方法与评价

- 基于规则的文本匹配
  - 启发式规则
  - 隐语义表达
- 基于学习的文本匹配
  - 人工特征融合
  - 表达学习
- 文本匹配的评价方法

文本匹配方法与评价

- 基于规则的文本匹配
  - 启发式规则
  - 隐语义表达
- 基于学习的文本匹配
  - 人工特征融合
  - 表达学习
- 文本匹配的评价方法

基于启发式规则的文本匹配

- 启发式规则的文本匹配模型直接建模了两段文本共同出现的词的分布。不难理解，在两段文本中同时出现的词，对于度量文本的匹配程度是至关重要的

- 两个经典模型
  - BM25
  - 查询似然模型（Query Likelihood Model）

基于启发式规则的文本匹配

- BM25
  - BM25是一个基于词袋的检索排序函数，用来评价查询项和文档之间相关性
  - 查询项被拆分成独立的查询词，BM25只考虑出现查询词的文档，并通过一个结合词频和逆文档频率构造的打分函数对文档进行排序

$$\text{Score}(\text{D},\text{Q}) = \sum_{\text{i}=1}^{\text{n}} IDF(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{avgdl}\right)}$$

$$IDF(q_i) = log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}$$

  - 其中$f(q_i, D)$是的词频函数，表示查询词q_i在文档D中出现的次数；$|D|$表示文档的长度，$avgdl$表示文档集内的平均文档长度，k_1和b是BM25的两个超参，可直接设定，$\text{k}_1 \in [1.2, 2.0], \text{b} = 0.75$；$N$表示文档集大小，$n(q_i)$表示文档集中包含$q_i$的文档的数量

基于启发式规则的文本匹配

- 查询似然模型
  - 查询似然模型是一种用于信息检索的语言模型，用来衡量查询项与文档的相关程度
  - 具体定义为，给定查询项Q的情况下产生文档D的似然概率P(D|Q)

$$P(D|Q) = \frac{P(Q|D) \cdot P(D)}{P(Q)}$$

  - 对于同样的查询项$P(Q)$是相等的；由于每一个文档视为是等概率产生的，所以$P(D)$也是个常数。所以，我们得到：

$$P(D|Q) \propto P(Q|D)$$

  - 这表明可以通过计算文档的参数查询项的概率来排序文档。通常将文档的语言模型定义为：

$$\mathrm{P}(Q|D) = K_Q \prod_{q \in Q} P(q|D)^{tf_{q,Q}}$$

基于启发式规则的文本匹配

- 查询似然模型
  - 通常将文档的语言模型定义为：

$$\mathrm{P}(Q|D) = K_Q \prod_{q \in Q} P(q|D)^{tf_{q,Q}}$$

  - 在实际计算中，K_Q对于给定的查询项是一个常数，所以只需要计算在给定文档D之后查询词q的产生概率$P(q|D)$
  - 由于并不知道文档的真实语言模型，因此通过查询词在文档中的出现频率作为估计量。最后将所有的查询词的概率连乘起来得到一个在区间[0,1]的值，就可以来排序相关文档

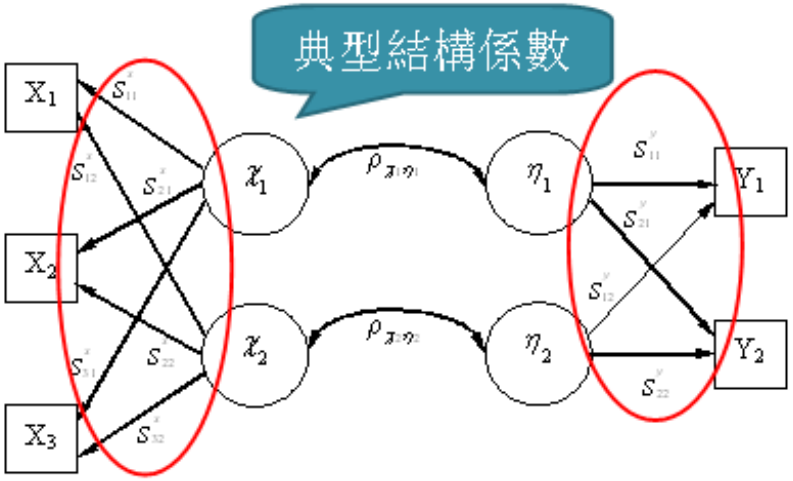
基于隐语义表达的文本匹配

- 除了两段文本中共同出现的词对计算匹配度有贡献，那些词与词之间的关系（如近义词、包含关系等）也应该考虑进匹配度的计算，因此提出了隐语义表达的文本匹配模型

- 这类方法将一段文本映射到一个向量（离散稀疏向量或者连续稠密向量），然后通过计算向量的相似度来表示文本的匹配度。文档表示的各类方法（如TF-IDF和BM25）可以参考上一节的相关内容

基于隐语义表达的文本匹配

- 典型相关分析(Canonical Correlation Analysis)
  - 基本原理：为了从总体上把握两组变量之间的相关关系，分别在两组变量中提取有代表性的两个综合变量u 和v（分别为两个变量组中各变量的线性组合），利用这两个综合变量之间的相关关系来反映两组变量之间的整体相关性



基于隐语义表达的文本匹配

- 给定两组向量x_1和x_2：
  - 令Σ_{11}是x_1的自协方差矩阵，Σ_{12}是$\mathrm{Cov}(x_1, x_2)$，Σ_{21}是$\mathrm{Cov}(x_2, x_1)$，也是Σ_{12}的转置，Σ_{22}是x_2的自协方差矩阵

- 定义$u = a^T x_1,\ v = b^T x_2$，可以计算$u$和$v$的方差和协方差如下：
  - $\mathrm{Var}(u) = \mathrm{Var}(a^T x_1) = \frac{1}{N}\sum_{i=1}^{N}(a^T x_{1i} - a^T \mu_1)^2 = a^T \frac{1}{N}\sum_{i=1}^{N}(x_{1i} - \mu_1)^2 a = a^T \Sigma_{11} a$
  - $\mathrm{Var}(v) = b^T \Sigma_{22} b$
  - $\mathrm{Cov}(u, v) = a^T \Sigma_{12} b$

- 最后，计算$\mathrm{Corr}(u, v) = \frac{a^T \Sigma_{12} b}{\sqrt{a^T \Sigma_{11} a} \cdot \sqrt{b^T \Sigma_{22} b}}$
  - $\mathrm{Corr}(u, v)$越大越好 ，说明u和v的相关性越强，文档匹配程度越高

文本匹配方法与评价

- 基于规则的文本匹配
  - 启发式规则
  - 隐语义表达
- 基于学习的文本匹配
  - 人工特征融合
  - 表达学习
- 文本匹配的评价方法

基于学习的文本匹配

- 在大数据的背景下，文本匹配除了考虑计算效率外，也要考虑结果的准确性。得益于大量有标注的文本匹配数据，基于学习的文本匹配模型可以通过有监督的方式，得到更为准确的结果

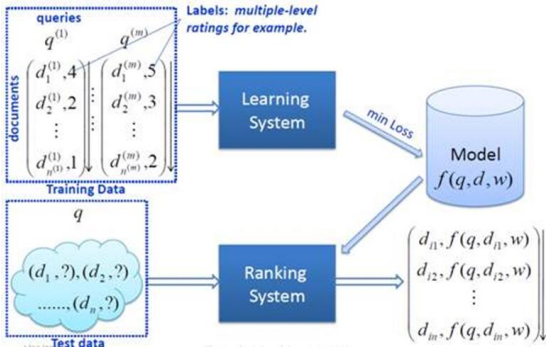
- 基于学习的文本匹配模型分为两类
  - 基于人工特征的排序学习模型
  - 基于表达学习的排序学习模型

基于人工特征的排序学习模型

- 人工特征
  - 人工特征是根据实际任务中对数据的理解，设计出的用以抽象数据的特征表示。BM25与TF-IDF都可以看作是一种人工设计的特征
  - 在文本匹配任务中，人工提取的特征可以分为两大类：基于文本内容的特征和基于文本交互的特征。通常情况下，人工提取的特征会拼接成一个特征向量，用来表示相应的文本
  - 基于文本内容的特征
    - 主要包括关键词、文本类型、文本长度等等
    - 还包括当前文本在与其他文本构造的关系图上的PageRank重要度特征，例如通过域名、链接等构造的文档关系图
  - 基于文本间交互的特征
    - 包括关键词匹配的数量、BM25、查询似然模型得到的匹配度得分等
    - 也包括一些精心设计的关于邻近度的特征

基于人工特征的排序学习模型

- 排序学习(Learning to Rank)
  - 排序学习算法以特征向量作为输入，以匹配度作为输出
  - 根据排序学习算法的输入数据的不同，可以将排序学习算法分为
    - 基于单样本的(Pointwise)排序算法
    - 基于样本对的(Pairwise)排序算法
    - 基于样本列表的(Listwise)排序算法



  - 在训练数据上，排序学习算法通过标注结果来训练模型的参数，学到从特征向量到匹配度的函数映射。而在使用的时候，为训练好的排序学习模型准备好输入数据，便能得到相应的匹配度

基于人工特征的排序学习模型

- 排序学习(Learning to Rank)
  - 基于单样本的(Pointwise)排序算法
    - 输入：以单个<查询项-文档>样本的特征向量作为输入
    - 输出：输出一个实数值，表示单个样本的匹配度
    - 假设：学习一个从特征向量到实数匹配度的映射函数，称为打分函数（scoring function）。得到这个匹配度之后，就能对文档列表进行排序得到最后的排序结果
    - 损失函数：单样本的排序算法可以建模成回归问题或分类问题，所以回归问题的损失函数和分类问题的损失函数都可以拿来使用
    - 预测的时候，如果得分大于设定阈值，则认为是相关的；如果小于设定阈值则可以认为不相关

基于人工特征的排序学习模型

- 排序学习(Learning to Rank)
  - 基于样本对的(Pairwise)排序算法
    - 输入：以一对<查询项-文档>样本的特征向量作为输入，这对样本有相同的查询项，但是文档不同，而且标注的匹配度也是需要有差异的
    - 输出：算法输出一个整数值-1或者+1，表示这对样本的一个偏序关系。-1表示第一个文档匹配度低于第二个文档，+1表示第一个文档匹配度高于第二个文档
    - 假设：样本对排序算法学习一个从一对样本的特征向量到该对样本的偏序关系的映射函数
    - 损失函数：样本对排序算法可以建模成分类问题来处理偏序关系，也可以使用类似于铰链损失(Hinge Loss)的方式定义损失函数

基于人工特征的排序学习模型

- 排序学习(Learning to Rank)
  - 基于样本列表的(Listwise)排序算法
    - 输入：以一个<查询项-文档>集合为输入，该集合内的所有查询项是相同的
    - 输出：输出一个集合的排列，得到这个排列之后就能得到最终的文档排序结果
    - 假设：学习一个从样本集合到样本排列方式的映射函数
    - 损失函数：主要有两种常用的损失函数：第一种是直接构造与评价指标相关的损失函数，而第二种则是与评价指标无关的损失函数

基于表达学习的排序学习模型

- 人工特征的设计是耗时耗力的，而且针对某一任务设定的特征在其他任务上往往会失效。如何从原始输入自动获取特征表达，是表达学习需要研究的任务

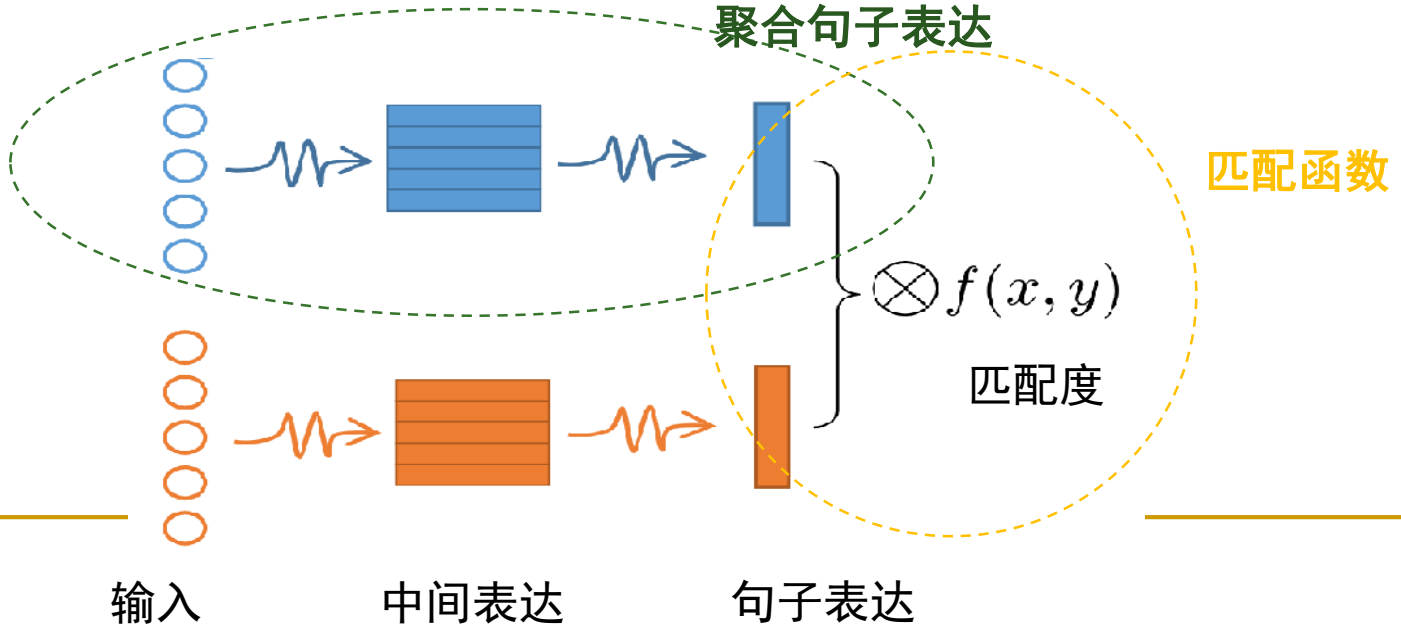
- 在文本匹配领域，随着有标注数据的积累，利用表达学习从原始输入中学到合适的特征表达的方式已经成为可能

基于表达学习的排序学习模型

- 目前已有大量基于表达学习的排序学习模型，该类模型可以直接以文本的内容作为输入，以匹配度作为输出，端到端的学习模型。文本匹配任务可以抽象成如下的形式：

$$匹配度 = \mathcal{F}\big(\Phi(S_1), \Phi(S_2)\big).$$

- 其中Φ用来将文本映射到对应的表达向量，$\mathcal{F}$用来定义两段文本表达的交互，并最终聚合成匹配度

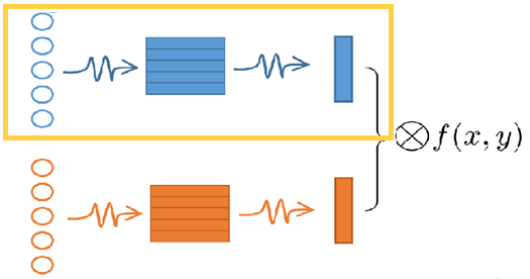


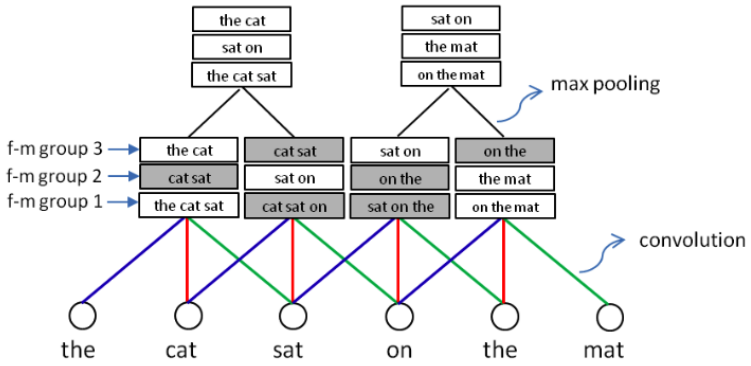
基于表达学习的排序学习模型

- 基于表达学习的文本匹配模型（Siamese 框架）
  - 第一步：计算文档的表达向量
    - 例如：全连接神经网络、卷积神经网络、循环神经网络等
  - 第二步：利用向量相似度函数度量文档表达之间的距离
    - 例如：余弦相似度、全连接神经网络等

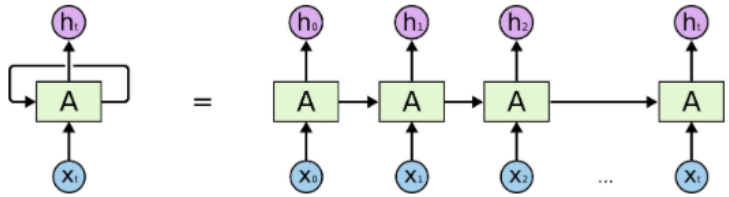
基于表达学习的排序学习模型

- 基于表达学习的文本匹配模型（Siamese 框架）
  - 第一步：计算文档的表达向量
    - 全连接神经网络
    - 卷积神经网络
    - 循环神经网络





卷积和池化
保持局部词序信息



循环结构
建模长距离依赖

基于表达学习的排序学习模型

- 基于表达学习的文本匹配模型（Siamese 框架）
  - 第二步：利用向量相似度函数度量文档表达之间的距离
    - 向量点积：

$$匹配度 = \Phi(S_1)^T \Phi(S_2)$$

    - 余弦相似度：

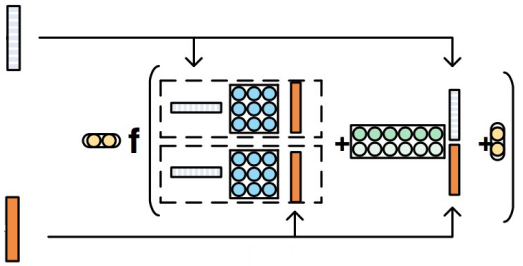
$$匹配度 = \frac{\Phi(S_1)^T \Phi(S_2)}{|\Phi(S_1)| \cdot |\Phi(S_2)|}$$

    - 全连接网络：

$$匹配度 = W_2(W_1[\Phi(S_1), \Phi(S_2)] + b_1) + b_2$$

    - 神经张量网络：

$$匹配度 = \Phi(S_1)^T M \Phi(S_2)$$



文本匹配方法与评价

- 基于规则的文本匹配
  - 启发式规则
  - 隐语义表达
- 基于学习的文本匹配
  - 人工特征融合
  - 表达学习
- 文本匹配的评价方法

文本匹配的评价方法

- 分类准确率（Accuracy）：用于评价分类任务的指标，对于文本匹配任务，只有两类标签，匹配为1，不匹配为0。因此可以把文本匹配看作是一个二分类问题。使用分类准确率可以方便的评价模型对每一对文本的分类是否正确。分类正确的数量占总测试样本数量的比例就是分类准确率

- P@k（Precision at k）：表示前k个文档的排序准确率。假定预测结果排序后，前k个文档中相关文档的数量为Y_k，那么$P@k$可定义为：

$$P@k = \frac{Y_k}{k}$$

- R@k（Recall at k）：表示前k个文档的排序召回率。假设所有相关文档的总数为N。按照预测结果排序后，前k个文档中相关文档的数量为G_k，那么$R@k$可定义为：

$$R@k = \frac{G_k}{N}$$

文本匹配的评价方法

- MAP（Mean Average Precision）：该指标综合考虑了所有相关文档的排序状况。将所有相关文档在预测结果排序中的位置定义为$r_1, r_2, \cdots, r_G$，则平均精度均值指标可定义为：

$$\mathrm{MAP} = \frac{\sum_{i=1}^{G} P@r_i}{G}$$

- MRR（Mean Reciprocal Rank）：如果只考虑预测结果排序中第一个出现的相关文档的位置r_1，可以定义MRR指标为：

$$\mathrm{MRR} = \mathrm{P}@r_1 = \frac{1}{r_1}$$

文本匹配的评价方法

- nDCG（normalized Discounted Cumulative Gain）归一化折扣累计收益
  - 有些任务当中标注的相关度本身就有大小之分而不是单纯的匹配和不匹配两个级别，这个时候nDCG这个指标就会更加有效。nDCG让相关度越高的排在越前面
  - 给定按照标注的文档相关度排序后的文档相关度值分别为 $\widehat{rel}_1, \widehat{rel}_2, \cdots, \widehat{rel}_N$，若按照预测结果排序后的文档相关度的值分别为 $rel_1, rel_2, \cdots, rel_N$。那么，nDCG指标的定义如下：

$$IDCG = \widehat{rel}_1 + \sum_{i=2}^{N} \frac{\widehat{rel}_i}{\log_2(i+1)}$$

$$DCG = rel_1 + \sum_{i=2}^{N} \frac{rel_i}{\log_2(i+1)}$$

$$nDCG = \frac{DCG}{IDCG}$$

小结

- 基于规则的文本匹配算法常用于大规模的信息初筛任务
- 基于学习的文本匹配算法常用于信息精细排序任务
- 进一步阅读：
  - 基于细粒度匹配信号的表达学习方法
  - DeepMatch模型[Lu & Li, 2013]
  - MatchPyramid模型[Pang et al., 2016]
  - DRMM模型[Guo et al., 2016]
  - DeepRank模型[Pang et al., 2017]

5.3 文本生成

文本生成

- 文本生成简介
- 文本生成方法
  - 基于马尔科夫语言模型的生成方法
  - 基于深度学习的Seq2Seq模型
- 文本生成任务
  - 人机对话生成
  - 图片标题生成
- 生成文本的评价方法
  - 内在评价方法
  - 外在评价方法

文本生成

- 文本生成简介
- 文本生成方法
  - 基于马尔科夫语言模型的生成方法
  - 基于深度学习的Seq2Seq模型
- 文本生成任务
  - 人机对话生成
  - 图片标题生成
- 生成文本的评价方法
  - 内在评价方法
  - 外在评价方法

文本生成

- 文本生成，亦称为自然语言生成(Natural Language Generation, NLG)，包含两大类：
  - 文本到文本的生成：从文本生成语言句子，典型的应用就是自动摘要、机器翻译、人机对话等
  - 数据到文本的生成：通过图片或者视频等数据，生成图片的摘要或者视频的描述等，典型的应用就是视频评论生成、图片标题生成等
- 文本生成的输入可以多种多样，不受局限
- 本节介绍这两种任务的典型代表
  - 人机对话生成与图片标题生成

文本生成任务的组成

- 1. 内容确定：决定在文本中包含哪些信息
- 2. 文本顺序：确定呈现句子的文本顺序
- 3. 文本归并：决定每个句子中呈现哪些信息，进行句子合并
- 4. 词汇化：找到正确的词汇和短语来表达信息
- 5. 引用表达式生成：选择领域对象需要识别的单词和短语
- 6. 语言实现：将所有的单词和短语组合成句子

文本生成任务的组成

- 内容确定
  - 作为生成过程的第一步，NLG系统需要决定哪些信息应该包含在生成的文本中，哪些不应该
  - 通常情况下，数据中包含的信息比我们想表达的文本信息更丰富、更详细，因此需要进行内容的筛选
  - 内容筛选一般取决于目标受众或者系统意图
    - 例如，在根据体育新闻数据生成以篮球为主题的摘要时，我们只需要筛选出体育新闻数据中关于篮球的句子，忽略掉其他的内容

文本生成任务的组成

- 文本顺序
  - 在确定要传达什么信息之后，NLG系统需要决定向受众展示信息的顺序
  - 早期的方法是使用人工的方法构建依赖于领域的结构化规则，来约定消息之间的话语关系
    - 例如，首先选择篮球比赛的赛前介绍，然后是队员介绍，最后是比赛实况
  - 但是人工构建规则需要大量的人力操作，对数据和任务的敏感性很高。后来，研究人员开始探索使用机器学习方法进行文本顺序的决策

文本生成任务的组成

- 文本归并
  - 因为需要表达的信息中会出现重复或冗余，所以不是每条信息都需要用单独的句子进行表达。为此，往往需要将多个消息合并成一个独立的句子，使生成的文本可读性更强
  - 一般来说，聚合归并、消除冗余和语言结构化都是很困难的。早期的归并方法是强烈依赖于应用领域的。它们通常是人工定义，依赖于领域和具体的应用
  - 近期的工作使用数据驱动的方法，从语料库数据中获取归并的规则。通过计算相似性，在一个平行语料库上对需要聚合的句子和相应的数据库条目构建归并系统
    - 例如，针对篮球比赛生成抽取式摘要时，我们需要计算句子之间的相似度，相似度大于阈值的句子认为是相似的，通过这种简单的方式进行句子归并

文本生成任务的组成

- 词汇化、引用表达式生成和语言实现
  - 词汇化的难点在于词语的选择是多种多样的，因此词汇化需要依赖于上下文进行选择。除此之外，词汇化还会依赖于其他信息，如要表达的态度、回答的有效性等
  - 最简单的方式就是直接使用原文信息，但是这种方法缺乏灵活性，只在特定的应用领域才能取得较好的效果
  - 有时针对一些需要特定内容的实体进行引用表达式生成，例如篮球比赛的得分等具体准确的实体
  - 语言的实现主要包含人工定义模板、基于语法的系统、统计学方法等

文本生成

- 文本生成简介
- 文本生成方法
  - 基于马尔科夫语言模型的生成方法
  - 基于深度学习的Seq2Seq模型
- 文本生成任务
  - 人机对话生成
  - 图片标题生成
- 生成文本的评价方法
  - 内在评价方法
  - 外在评价方法

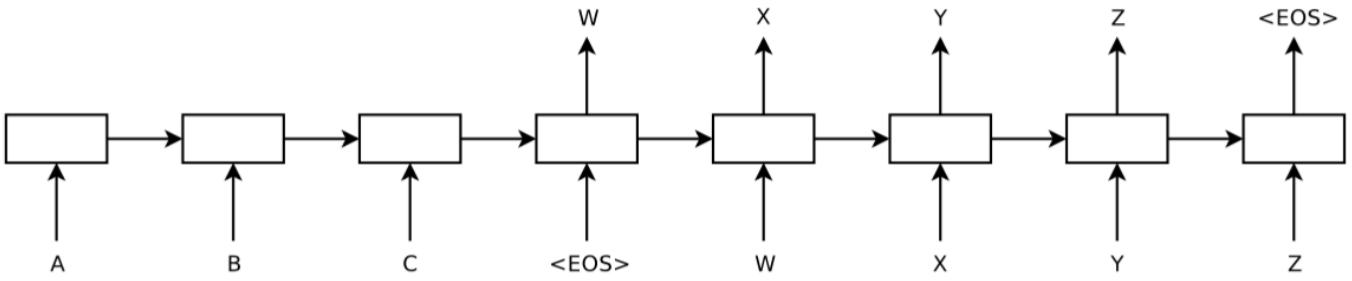
数据导向的文本生成方法

- 数据导向的文本生成是当前主要的流行方法，它依赖于统计学习而非语言学，通过学习输入和输出之间的对应关系，实现文本生成（包括句子的规划和生成）

- 基于马尔科夫语言模型的生成方法［Liang等人，2009］
  - 利用马尔科夫过程构建概率模型$p(w/s)$，即在输入状态s 下生成词语w 的条件概率，具体用一组隐变量描述w和s之间的关系
  - 缺点：建立在局部历史数据假设上，无法解决多轮对话等长距离依赖问题

数据导向的文本生成方法

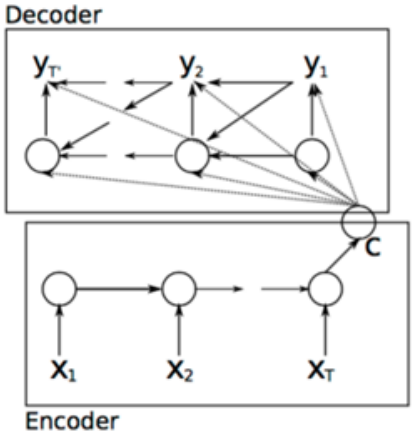
- 基于深度学习的Seq2Seq模型 [Sutskever等人, 2014]
  - 模型通过大量<X-Y>对来学习通过X生成Y的过程
    - 机器翻译：X是源语言句子，Y是目标语言句；对话：X是上一句话，Y是回复
  - 模型将源语言句子通过LSTM编码成一个固定维度的向量，然后用另一个LSTM进行解码，解码的目标是最大化目标语言句子的生成概率
    - 举例：源语言句子是ABC，编码过程一步一步将A、B、C的词向量表达输入到编码器LSTM中，LSTM编码器会得到一个固定维度的向量表示h_t，然后另一个LSTM模型作为解码器，根据h_t进行状态初始化，给定起始符<EOS>开始生成第一个词语W，然后将W的词向量输入到解码器中陆续得到X、Y、Z、<EOS>



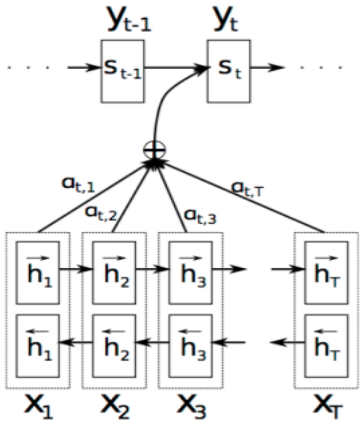
输入：ABC → 输出：WXYZ

数据导向的文本生成方法

- 基于深度学习的Seq2Seq模型
  - [Cho等人，2014] 为了使上下文信息更好地用于解码过程，将LSTM解码器的初始化状态设置为零向量，然后固定维度向量h_t作为LSTM解码器的输入，并与目标语言句子的每个词向量进行拼接



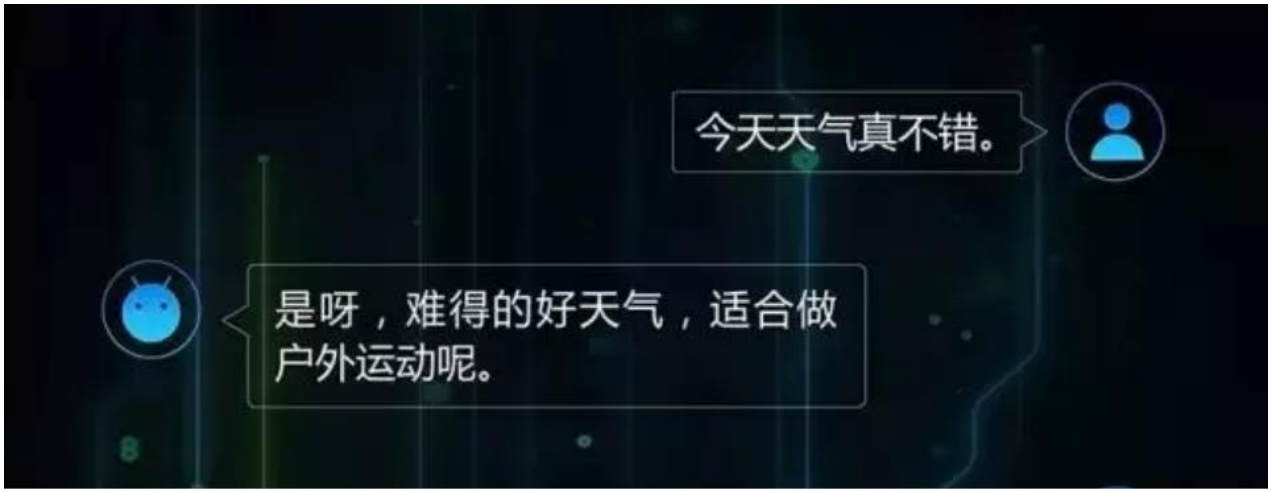
  - [Bahdanau等人，2014] 由于编码器把源语言句子进行向量映射会导致长句依赖问题，提出了使用注意力机制去加强对齐的信息，针对目标句子的每一个生成词语，计算它对应的源语言句子的注意力应该在哪些词语上



文本生成

- 文本生成简介
- 文本生成方法
  - 基于马尔科夫语言模型的生成方法
  - 基于深度学习的Seq2Seq模型
- 文本生成任务
  - 人机对话生成
  - 图片标题生成
- 生成文本的评价方法
  - 内在评价方法
  - 外在评价方法

人机对话生成



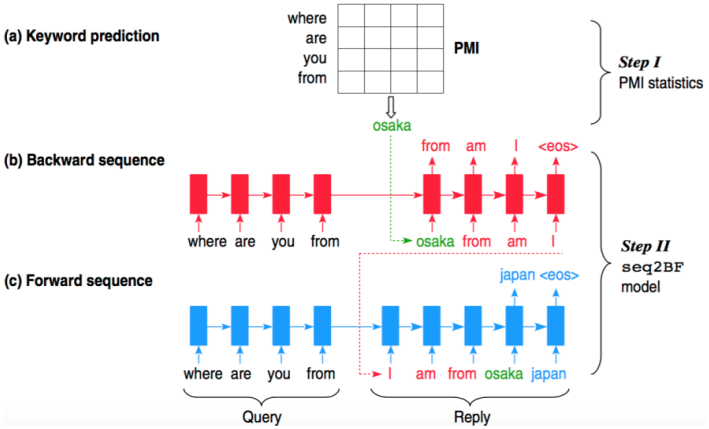
- 形式化描述
  - 给定一段由t-1个句子构成的对话历史信息$D = \{d_1, d_2, \dots, d_{t-1}\}$，预测下一句的回复$d_t = \{w_1, w_2, \dots, w_n\}$，建模条件概率$P(d_t|D)$

$$P(d_t|D) = \prod_{i=0}^{n} P(w_i|D, w_0, \dots, w_{i-1})$$

这里，w_0 一般为生成模型的起始字符<s>

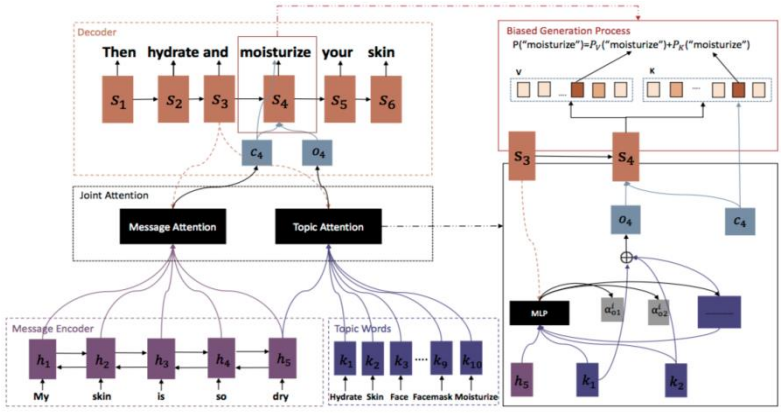
人机对话生成的核心问题

- 对话一致性问题
  - 对话的一致性是目前对话领域急需解决的问题。当前的Seq2Seq模型可以生成流畅的语句，但是很多回复都是通用回复，如“我不知道”“那是什么？”“有趣”
  - [Li等人，2016] 用Seq2Seq模型结合簇搜索生成多个候选回复，然后计算其与原句的互信息，选出互信息最大的句子作为最终回复
  - [Li等人，2016] 利用强化学习进行训练，将互信息、无聊回复的程度、话题一致性等信息作为奖赏



根据关键词向前向后生成回复

Lili等人[2016]

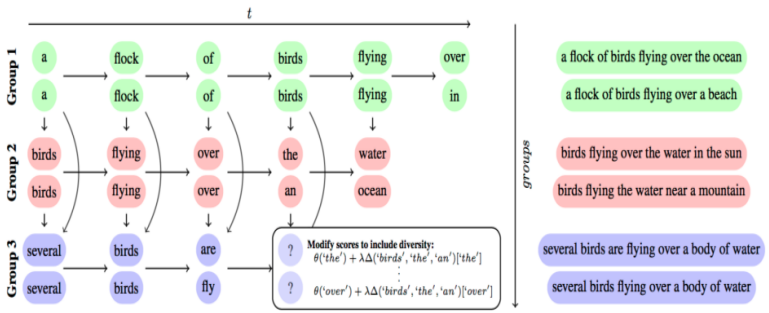


话题敏感的对话生成模型

Xing等人[2017]

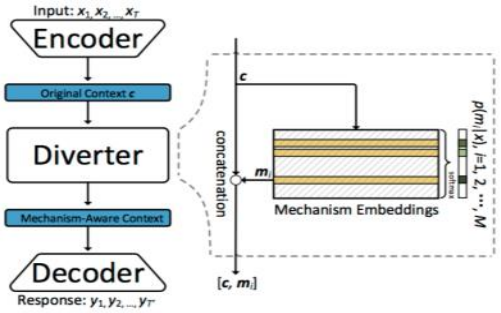
人机对话生成的核心问题

- 对话多样性问题
  - 回复多样化，对同一原句产生不同但都合理的回复
    - 例如，对于同一问题“今天天气如何”，可以回复“今天天气晴朗，适合出游”或“今日阳光明媚，气温舒适”
  - [Ashwin等人，2016] 将经典算法中的簇分组，引入一个惩罚机制使组间相似度尽量低，保证了候选回复相互之间有较大差异，满足多样性需求
  - [Zhou等人，2016] 为了使回复更加多样且有不同的风格，将原句映射成向量后与风格向量进行线性变换获得该原句在不同风格下的固定维度向量，再将这些向量输入到解码器中，生成不同风格的句子



面向多样性的簇搜索

Ashwin等人[2016]

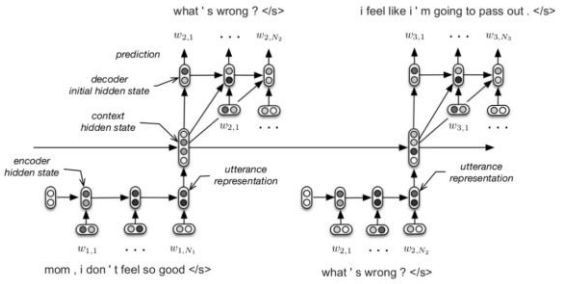


生成风格不同的回复

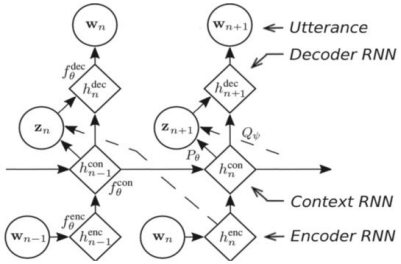
Zhou等人[2016]

人机对话生成的核心问题

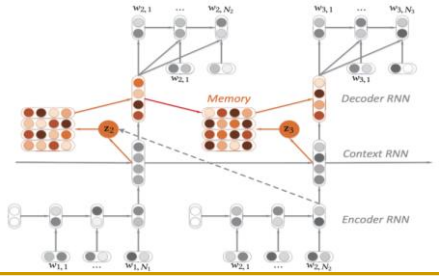
- 多轮对话长距离依赖问题：多轮对话拥有更多的上下文信息，如何利用好这些上下文信息，对于多轮对话生成极为关键
  - 例如：
    - Q1：明天天气状况如何？
    - R1：明天阴天有小雨，气温较低。
    - Q2：有哪些适合游玩的地方？
    - R2：建议去古北水镇或颐和园，雨天景色独特。（考虑雨天适合的景点）
    - Q3：去古北水镇的出行路线？
    - R3：坐980路转38路至司马台村，然后步行达到。(雨天避免骑行路线)
  - [Serban等人，2016] 使用层次化RNN编码解码器解决多轮对话的长依赖问题。首先，在词语级别使用RNN进行句子的隐层表达；然后，在高层使用一个上下文RNN对句子的隐层表达进行再编码，得到句子级别的向量表示，通过该向量进行回复的生成和查询的扩展。该模型可以获得整个上下文的语义表示



层次化编码解码器模型
Serban等人[2016]



层次化隐变量编码解码器模型
Serban等人[2017]



层次化隐变量记忆网络模型
Chen等人[2018]

图片标题生成

(a)一辆在路边的骑车和蔚蓝的天空

(b)一辆在路上行驶、在一个建筑物前面的

(c)壁虎站在树枝上[Hendricks et

- 图像标题生成：从静态、无序的图像输出连续的文字
- 形式化描述
  - 给定一张图片M，生成该图片的标题$C=\{w_1,w_2,\ldots,w_n\}$，建模条件概率$P(C|M)$

$$P(C|M)=\prod_{i=0}^{n}P(w_i|M,w_0,\ldots,w_{i-1})$$

其中，w_0 一般为生成模型的起始字符<s>

图片标题生成

- 图像分析
  - 检测
    - 利用计算机视觉方法检测图像中包含的人物与物体，然后将这些输出与语言结构进行映射，如树结构或模板
  - 整体场景分析
    - 分析图像所呈现的整体场景，以及物体之间的空间关系等
    - 通常使用场景更全面的表征，包括RGB直方图、尺度不变的特征变换，或低维的空间表示等
  - 特征提取
    - 鉴于卷积神经网络CNN在计算机视觉任务的成功应用，许多深度学习方法都使用它来做特征提取，例如AlexNet, VGG, Caffe

图片标题生成

- 标题生成或检索
  - 基于模板或树结构
    - 系统做完检测以后，可以将检测的特征映射到输出的语言结构。例如，对象可以被映射到名词，空间关系映射到介词等。模板可以生成更通顺的句子，但是缺乏可变化性
  - 基于语言模型
    - 使用语言模型可以促进<图像-语言>对的联合训练。相较于语法和模板的方法，它可以产生更有创造性和表达力的标题
    - 大多工作使用RNN或LSTM模型将标题生成转化为预测标题的下个单词的过程，基于已产生的不完整的标题和分析所得到的图像特征预测标题的下一个单词
  - 基于检索和重组
    - 除了生成的方式，还可以根据训练数据进行检索得到标题，好处是可以保证语言的流畅性和完整性
      - Hodosh等人[2013] 将检索看作是识别最近的标题和图像特征的过程。通过比较查询图像的标题和解析的图像，基于WordNet找到最相似的标题

文本生成

- 文本生成简介
- 文本生成方法
  - 基于马尔科夫语言模型的生成方法
  - 基于深度学习的Seq2Seq模型
- 文本生成任务
  - 人机对话生成
  - 图片标题生成
- 生成文本的评价方法
  - 内在评价方法
  - 外在评价方法

生成文本的评价方法

- 内在评价方法
  - 内在评价衡量系统的性能，它通常与文本质量有关。如，所生成文本的正确性和可读性都是内在的评价
  - 包括人的主观评价、基于语料库的仿人工评价
- 外在评价方法
  - 外在评价是评估系统是否真正实现了任务的目标

文本生成的评价

- 内在评价方法—人工评价
  - 人工评价通常是由专家按照一定的标准来评价所生成的文本
  - 标准主要包括：
    - 流畅性和可读性，即文本的语言质量
    - 准确性，充分性，相关性或正确性
  - 虽然这些评价标准很常见，但是并不很全面
    - 如，在图像标题生成中，系统至少要评价所生成标题的创新性
  - 主观评估的另一个问题就是评估者之间的差异性和可靠性
    - 不同专家的多重判断可能表现出很大的差异。这种差异可以通过反复迭代的方法减少，专家在通过一段时间的训练之后，重新更新评估准则。不过，这会消耗更多的时间和资源成本

文本生成的评价

- 内在评价方法--基于语料库的仿人工评价

	指标	描述	应用领域
n-gram重合度	BLEU	在可变长度的n-grams计算准确率，还有句长惩罚(Papineni et al., 2002)和平滑(Lin&Och, 2004)	机器翻译
	NIST	BLEU的另一版本，提高了低频n-grams和不同长度惩罚的权重(Doddington, 2002)	机器翻译
	ROUGE	比较不连续的n-grams和最长的公共子序列，计算召回率得分(Lin &Hovy, 2003)	自动摘要
	METEOR	Unigram 准确率和召回率的调和平均值(Lavie & Agarwal, 2007)	机器翻译
	GTM	连续匹配跨度在准确率和召回率之间的F1值(Turian et al., 2003)	机器翻译
	CIDEr	使用tf-idf计算n-gram的权重，然后计算n-gram的cosine相似度(Vedantam et al., 2015)	标题生成
句子距离	编辑距离	将候选字符串转化为目标字符串所需要的插入、删除、替换的数量(Levenshtein, 1966)	
	TER	翻译编辑率，编辑距离的一种(Snover et al., 2006)	机器翻译
	TERP	TER处理短语替换、词干还原、同义词的版本(Snover et al., 2006)	机器翻译
	TERPA	为充分判断相关性提出的优化TER(Snover et al.,2006)	机器翻译
内容重合度	Dice/Jaccard	两个无序集合之间重叠的集合论测度	
	MASI	集合项目之间的一致性度量，Jaccard的加权版本(Passonneau, 2006)	自动摘要
	SPICE	将标题解析为图，表示对象和关系，同事将文本解析为对象和关系，来测量候选和参考文本之间的重叠度(Anderson et al., 2016)	标题生成

文本生成的评价

- 外在评价方法
  - 外在的评价方法是评估是否达到预期的目标，其有效性取决于应用领域和系统的具体目的
    - 例如，在戒烟信件系统中是否说服和改变行为；通过用户模型的住房市场系统提出赞成和反对以后，是否购买住房；在个性化系统中，是否增强了复杂交流用户之间的语言互动等等
  - 外在评价的潜在缺点除了时间和费用外，还需要依靠一个足够的用户群

小结

- 文本表达 将人类的语言符号表示成计算机能够处理的向量，是理解文本大数据的基础

- 文本匹配 将人们的信息需求与文本内容相关联，是运用文本大数据的基础

- 文本生成 旨在模拟人类语言交流，是人工智能的高级目标，是产生文本大数据的基础

思考

- 语言是高度抽象和复杂的，传统基于规则的建模过于简单，而利用稠密向量表示又不可解释，如何结合语言的语法规律和统计规律得到更为高效、精准的文本表达？

- 由一个向量的距离判断文本匹配过于抽象，而且匹配可能发生在文本更细粒度的尺度之上，如何建模细粒度的匹配信号？

- 现阶段的文本生成还局限于较短的文本，在更大规模的数据之上，是否能层次化的生成更长更复杂的文本？

谢谢！