数字媒体技术基础

Meng Yang

www.smartllv.com

SUN YAT-SEN University

机器智能与先进计算教育部重点实验室

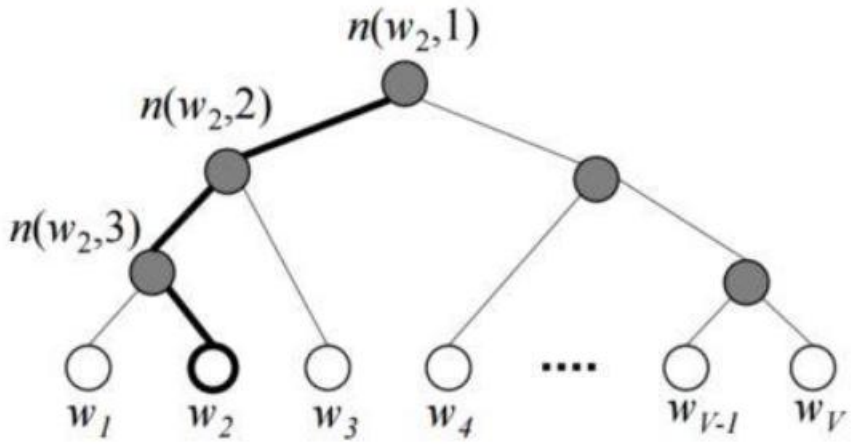
智能视觉语言学习研究组

词向量表示

然而，词库很大时，计算softmax十分困难。

问题：如何处理？

Word2vec提出了Hierarchical softmax把N分类问题拆解成log(N)次二分类。



$$O(N) \rightarrow O(\log N)$$

词向量表示

静态词向量和动态词向量：

- 静态词向量：每个单词的词向量不会变化。无法解决“一词多义”的问题。
- E.g., Word2vec、Glove、Fasttext

- 动态词向量：每个词的词向量会随着上下文而变化。因为单词含义会受到上下文影响，ELMO在一定程度上解决“一词多义”问题。
- E.g., ELMO、BERT

词向量表示

对于下面这两个句子：

- 句1：“**苹果**公司招聘研究员”
- 句2：“我喜欢吃**苹果**”

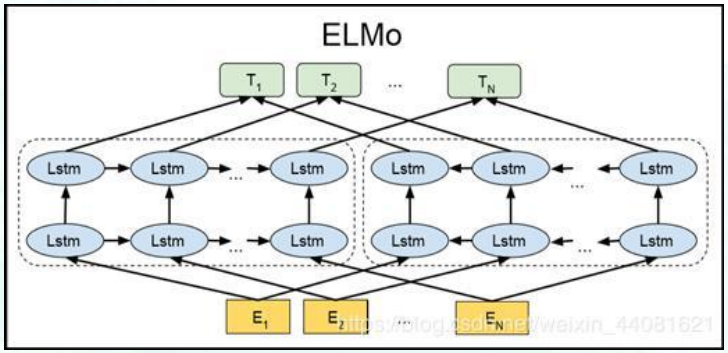
静态词向量：两句的“苹果”的词向量是一样的

动态词向量：两句的“苹果”的词向量是不同的

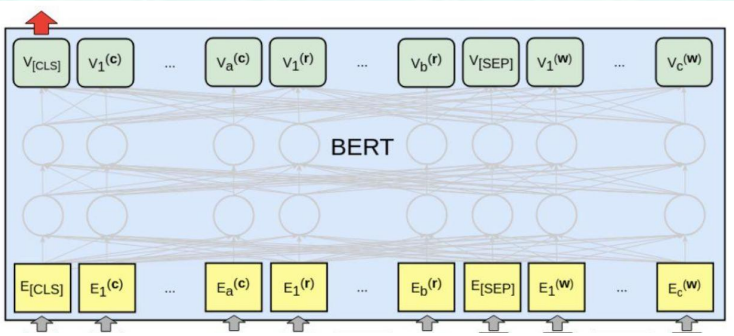
词向量表示

动态词向量：

- ELMO采用双向LSTM来训练模型，分别考虑“上文”和“下文”：



- BERT改用transformer来提取特征，其中由self-attention来完成词语之间的交互：



目 录

7.3 文本的表示方法

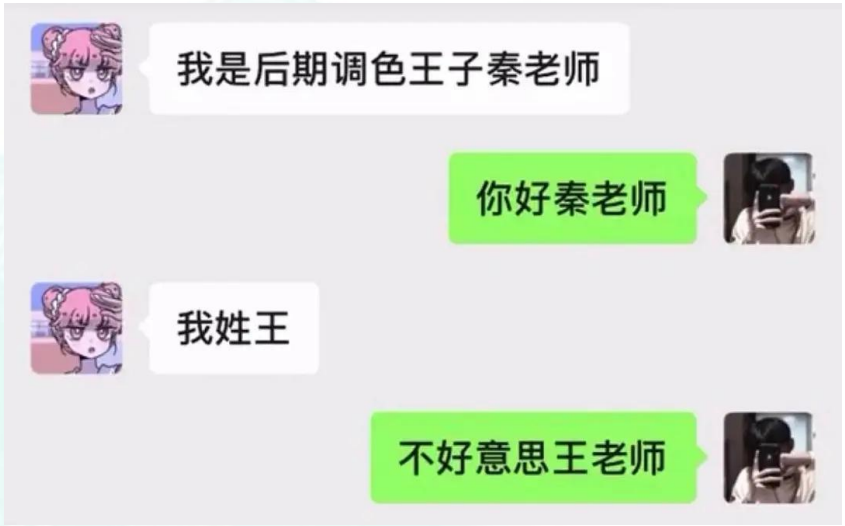
文本分词

如何理解一个中文句子？

“你/想过/去/旅游/吗”
“你想/过去/旅游/吗”

“南京市/长春大桥”
“南京/市长/春大桥”

“乒乓球拍/卖完/了”
“乒乓球/拍卖/完/了”



文本分词

文本分词指的是将一段文本拆分成维持原顺序的多个词语片段的过程。

- 倘若不对文本进行分词就输入到模型中，模型参数会变得过于庞大。
- 而如果对文本直接拆分成字，则容易丢失词的语义信息；并且，拆分后的序列过长，不便于模型处理。

总的来说，分词方法可分为基于词典的分词和基于机器学习模型的分词。

文本分词——基于词典

正向最长匹配算法：

- 如果文本中出现了“中山大学”，那么我们希望它不要被拆分成“中山”和“大学”两个词语。原因在于：越长的单词，其表达的含义就越丰富。
- 正向最长匹配算法：按照从左到右的顺序，优先切分为长度更长的词。

“我来自中山大学”

“我/来自中山大学”

“我/来自/中山大学”

文本分词——基于词典

然而，也会有出乎意料的情况：

“他是中山大学霸”

“他/是中山大学霸”

“他/是/中山大学霸”

“他/是/中山大学/霸”

“研究生命起源”

“研究生/命起源”

“研究生/命/起源”

文本分词——基于词典

逆向最长匹配算法：按照从右到左的顺序，优先切分为长度更长的词。

“他是中山大学霸”

↓

“他是中山大/学霸”

↓

“他是中山/大/学霸”

↓

“他是/中山/大/学霸”

↓

“他/是/中山/大/学霸” √

“研究生命起源”

↓

“研究生命/起源”

↓

“研究/生命/起源” √

文本分词——基于词典

然而，逆向最长匹配算法也会存在不符合预期的分词结果。

“项目的研究方法”

“项/目的/研究/方法”

在一些句子中，正向最长匹配效果好；

在一些句子中，逆向最长匹配果好……

文本分词——基于词典

Sun等人采样了3680个句子，对正向和逆向最长匹配的结果进行了统计：

二者的结果是否相同	结果是否正确	占比
相同	均正确	**90.30%**
相同	均错误	**0.41%**
不同	其中一个正确	**9.24%**
不同	均错误	**0.054%**

有9.24%的句子，如果一种方法分词错误，用另一种方法就能正确分词。

Sun M, Tsou B K. Ambiguity resolution in Chinese word segmentation[C]//Proceedings of the 10th Pacific Asia Conference on Language, Information and Computation. 1995: 121-126.

文本分词——基于词典

有人提出，能否将正向和逆向匹配相结合，取长补短，以适应更多的情况。

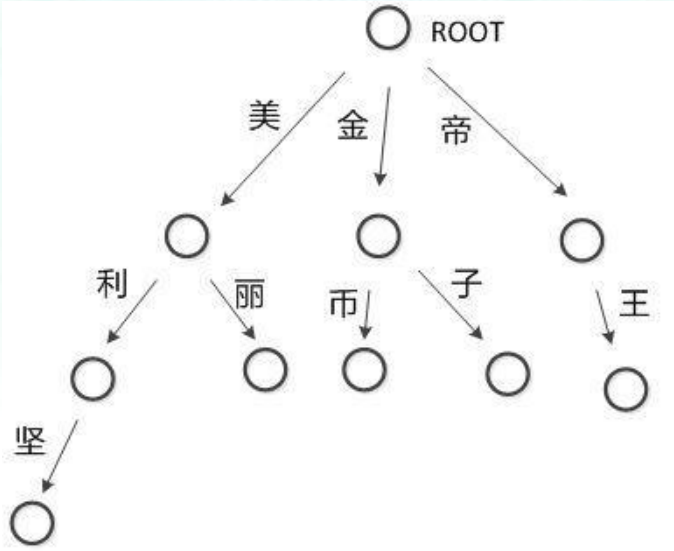
双向最长匹配算法：

- 同时执行正向和逆向最长匹配，若二者的词数不同，则返回词数少的那个；
- 否则，返回二者中单字少的那个；
- 当单字数量也相同时，返回逆向最长匹配的结果。

文本分词——基于词典

瓶颈：如何判断词典中是否含有字符串（即候选的分词片段），能否做到较低的时间复杂度和空间复杂度？

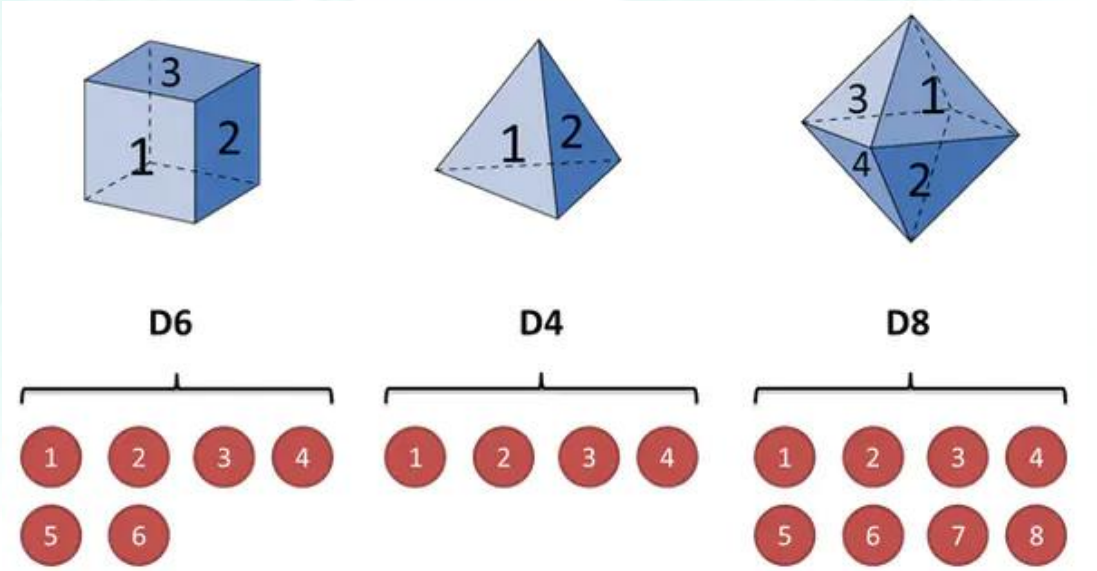
字典树（前缀树、Trie树）：顺着路径往下走，如果能找到候选词，则说明这个候选词其在集合中。



文本分词——基于机器学习模型

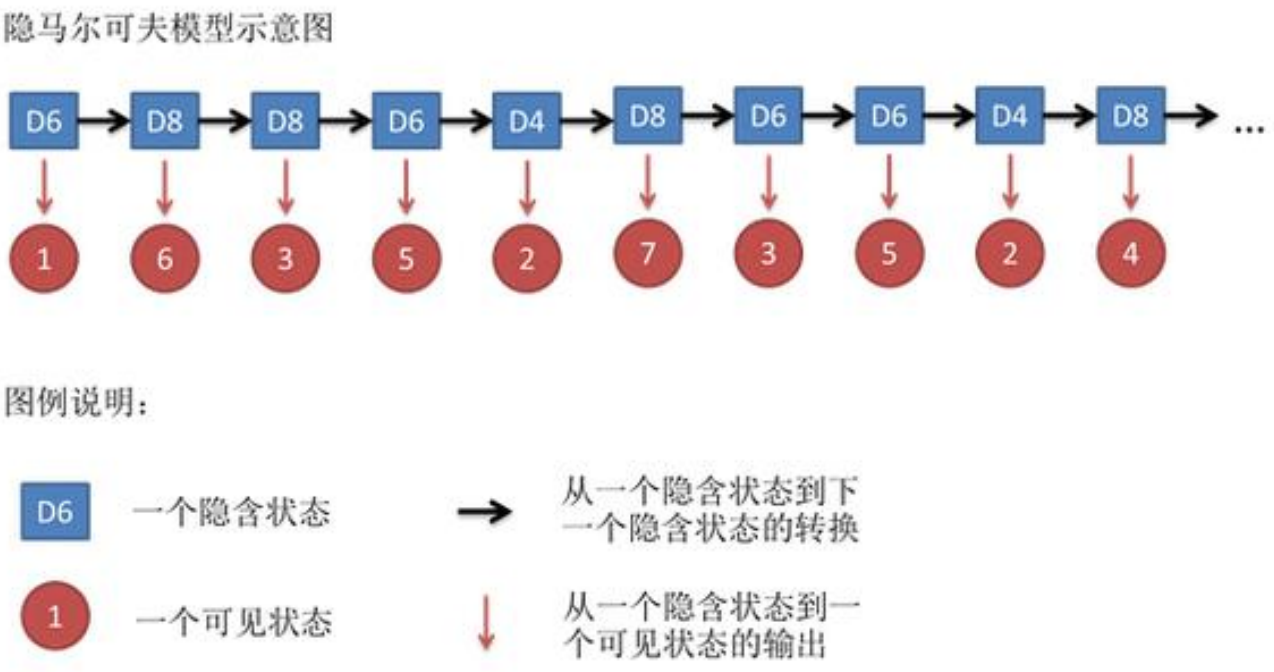
隐马尔可夫模型（HMM）：

假设有三个不同的骰子。第一个骰子有6个面，每个面的概率是1/6；第二个骰子有4个面，每个面的概率是1/4；第三个骰子有八个面，每个面的概率是1/8。



文本分词——基于机器学习模型

当我们看到 1 6 3 5 2 7 3 5 2 4 这个序列时，其实还包含了隐含状态：每个数字是由哪个骰子掷出。



HMM的预测过程（也叫解码过程），就是求隐含状态的过程。

文本分词——基于机器学习模型

对于分词，定义四个状态，对句子进行标注：

- B:begin，代表该字是词语中的起始字
- M:middle，代表是词语中的中间字
- E:end，代表是词语中的结束字
- S:single，代表是单字成词

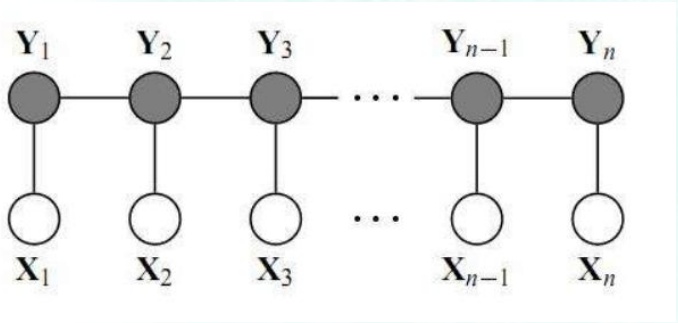
给你一个隐马尔科夫链的例子。

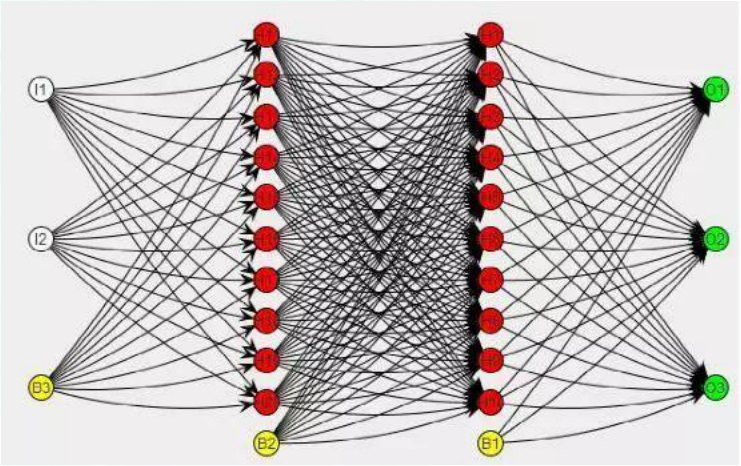
给/S 你/S 一个/BE 隐马尔科夫链/BMMMME 的/S 例子/BE 。/S

这四个状态可类比为上面提到的三个骰子，即可利用HMM求解。

文本分词——基于机器学习模型

同样的思想，在对句子进行标注后，也能利用其它机器学习模型，例如条件随机场、神经网络等。

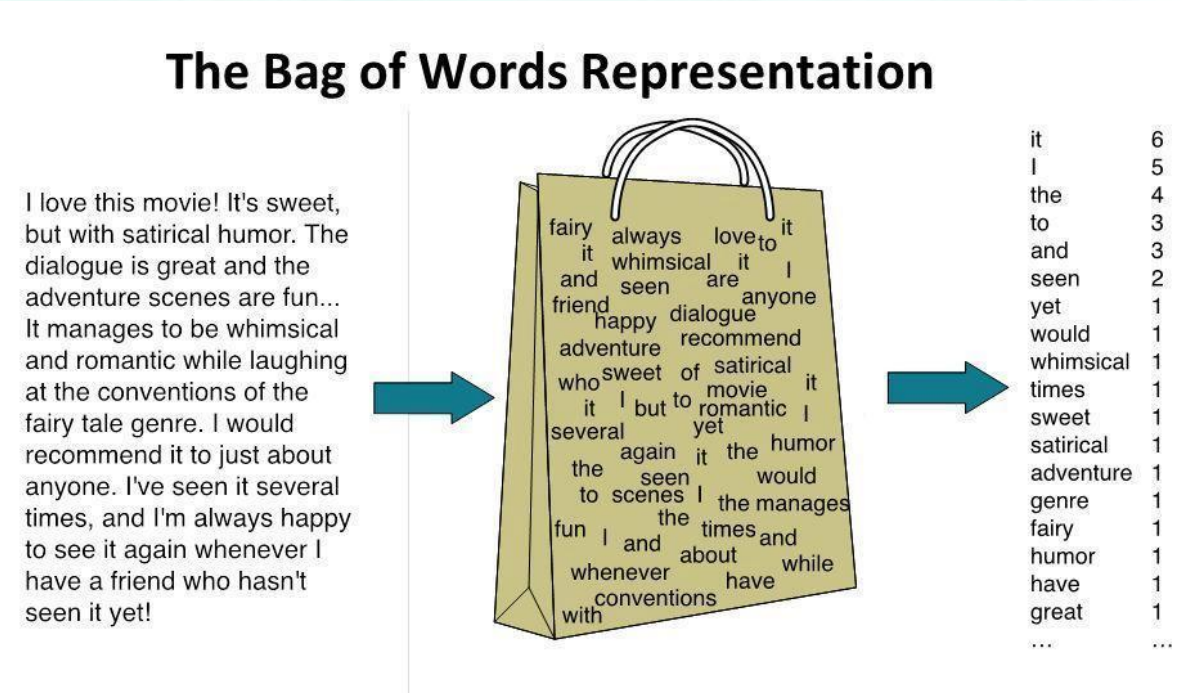




词袋模型

文本经过分词后，变成了便于处理和理解的序列。

为了将文本转化为向量表示，词袋模型将所有词语装进一个袋子里，不考虑其词法和语序的问题，即每个词语都是独立的。



词袋模型

E. G.

给定一个词典：

我	他	喜欢	学习	语文	数学

就能对下面两个句子编码：

我	喜欢	学习	语文

1	0	1	1	1	0

他	喜欢	学习	数学

0	1	1	1	0	1

词袋模型让语序关系完全丢失。

TF-IDF

阅读一篇期刊论文时，如何最快速（阅读最少的文字）知道这篇文章的主题？

记笔记

问答系统中问句分类方法研究综述

韩东方　吐尔地·托合提　艾斯卡尔·艾木都拉

新疆大学信息科学与工程学院

摘要： 问答系统作为信息检索的一种高阶形式,能够迅速、精准地为用户提供所需的信息服务,在给定一个问题后,会相应地给出准确的答案,这使得它在自然语言处理领域成为一个越发受人关注的研究方向。问句分类作为问答系统中的问题分析和处理的首要环节,是问答系统中尤为重要的一部分,其分类精度会直接影响到问答系统的性能。近些年来,机器学习和深度学习等技术的快速发展极大地促进了问句分类的研究和发展,其在问句分类上具有较强的可行性和优越性。为此就问句分类的国内外研究现状、问句分类标准体系、问句特征抽取、传统的机器学习分类方法和近来流行的深度学习分类方法进行总结和分析,阐述了问句分类当前所面临的一些研究难点,并对未来的研究和发展方向做了初步展望。

关键词： 问答系统；问句分类；分类体系；机器学习；深度学习；

基金资助： 国家自然科学基金（61562083,61262062）；国家重点研发计划（2017YFC0820603）；

专辑： 信息科技

专题： 计算机软件及计算机应用

分类号： TP391.1

TF-IDF

TF-IDF（term frequency－inverse document frequency，词频-逆向文件频率）是一种用于信息检索与文本挖掘的常用加权技术。

TF-IDF是一种统计方法，用以评估一个字词对于一个语料库中的一篇文章的重要程度，因此也能用于文章的关键词提取。

字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。

TF-IDF

- 词频（TF）的计算方法：

$$TF = \frac{\text{该词在该文章的出现次数}}{\text{该文章的总词数}}$$

文章中的高频词通常能代表文章的主题或是中心内容，因此TF高的词也能在一定程度上代表这篇文章。

但是，如果仅靠词频TF，会发现大量的文章的高频词都是“的”、“了”、“你”等词，没有太大意义。

TF-IDF

- 逆文档频率（IDF）的计算方法：

$$IDF = log(\frac{\text{语料库的文章总数}}{\text{包含该词的文章数}+1})$$

如果一个词在整个语料库中越常见，分母就越大，逆文档频率就越小。

TF-IDF

- TF-IDF同时考虑了词频和逆文档频率，计算方法：

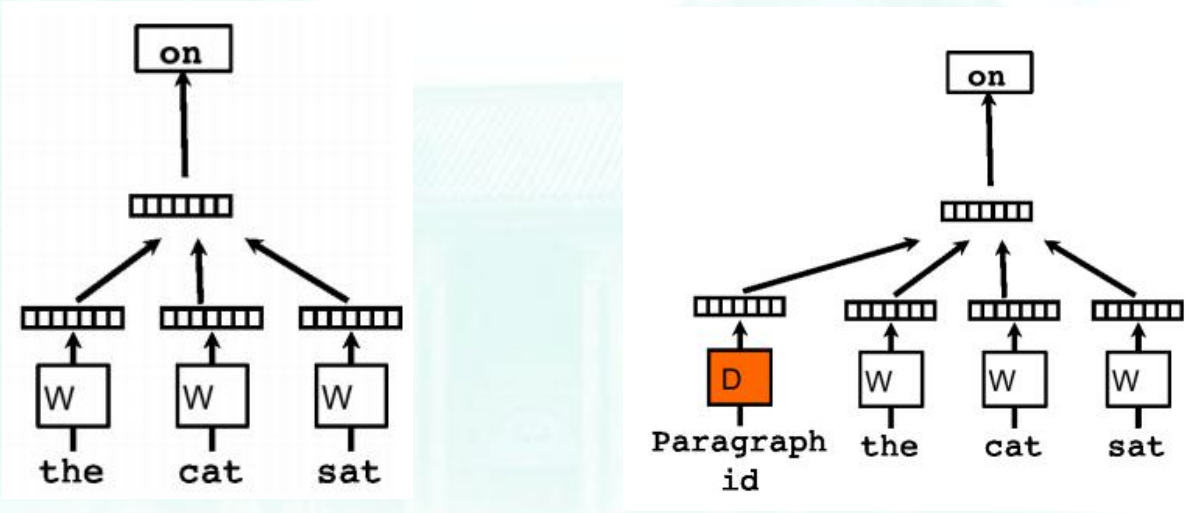
$$TF - IDF = TF * IDF$$

如果某个单词在一篇文章中出现的频率高，并且在其他文章中很少出现，则认为此词或者短语具有很好的类别区分能力，适合用来分类。

TD-IDF值可作为文章里各词的权重，然后便能对各词做加权求和，得到文章的表示。

基于词的聚合表示

- Doc2vec：Word2vec的拓展，增加了对文章的表示。

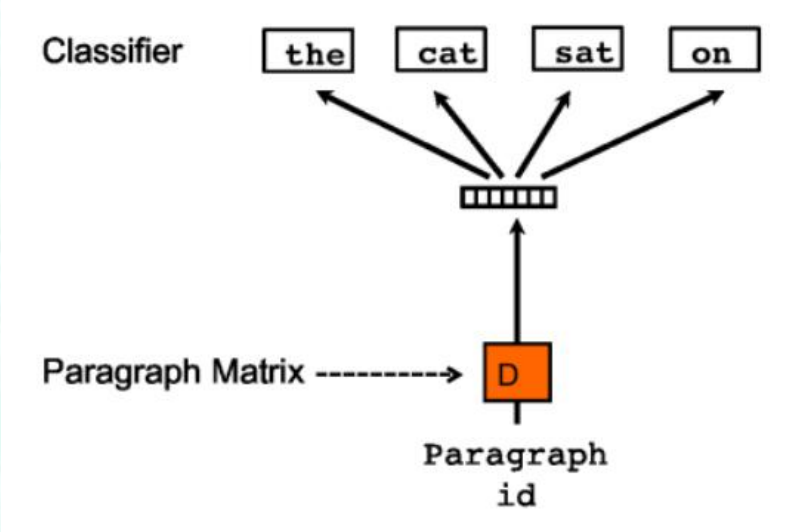


word2vec　　doc2vec

Le Q, Mikolov T. Distributed representations of sentences and documents[C]// International conference on machine learning. PMLR, 2014: 1188-1196.

基于词的聚合表示

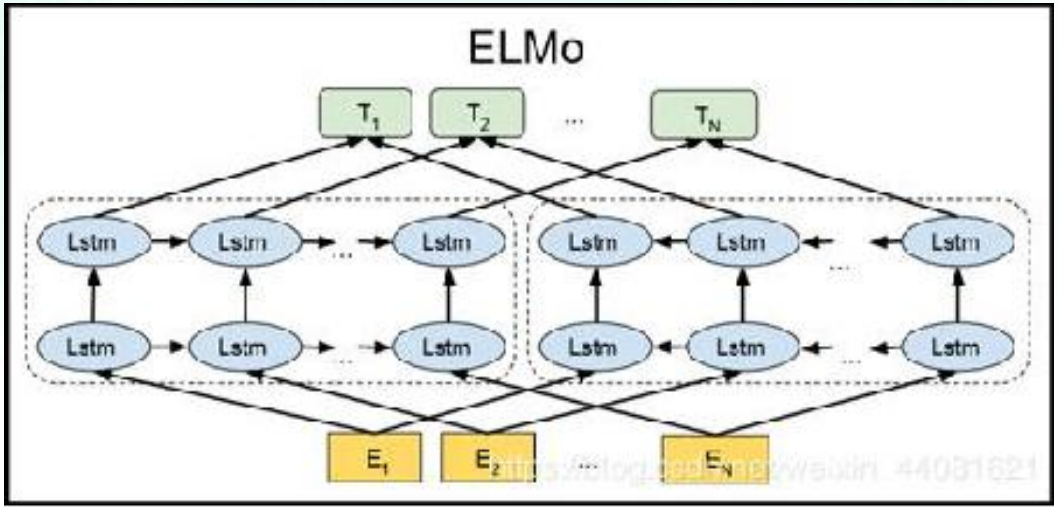
- Doc2vec还有另一种形式：忽略输入的上下文词，但强迫模型去预测从段落上采样得到的词。



- 训练完之后，段落向量可用于表示段落的特征，我们可以将这些特征直接用在传统的机器学习方法里，例如逻辑回归，支持向量机或K-means。

基于词的聚合表示

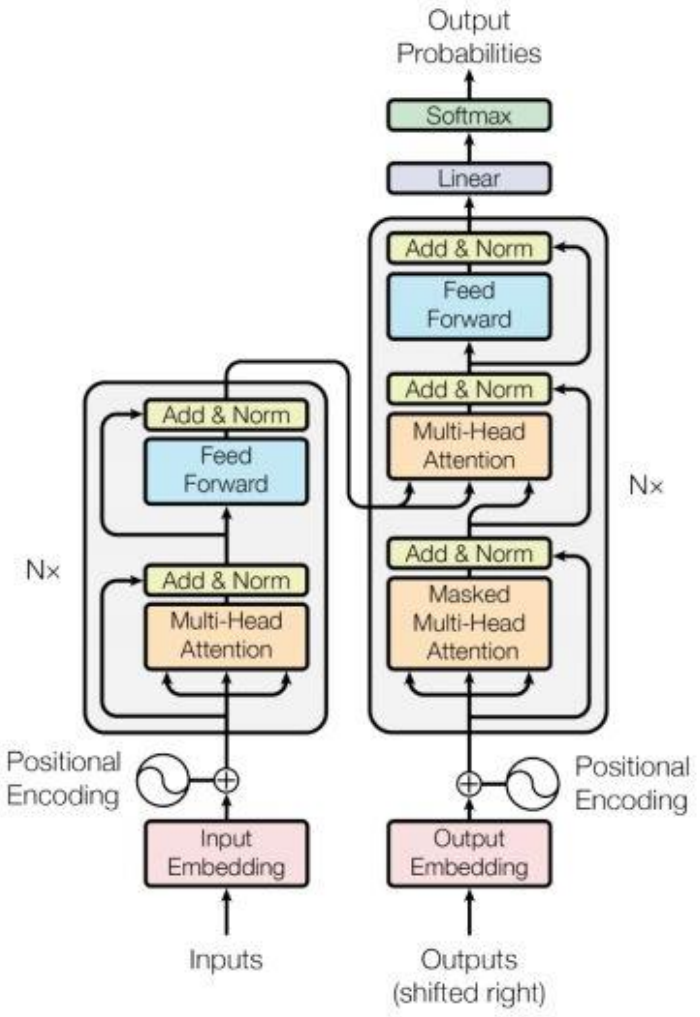
- 动态词向量模型，例如ELMO模型，采用双向LSTM对文本编码，因此得到的向量表示（句首位置+句尾位置）也可以当做是文本的表示。



基于词的聚合表示

Transformer：由Encoder和Decoder组成。

- Encoder：对输入的文本编码成向量表示。
- Decoder：对Encoder得到的向量表示进行解码，输出预测结果（例如预测单词）。

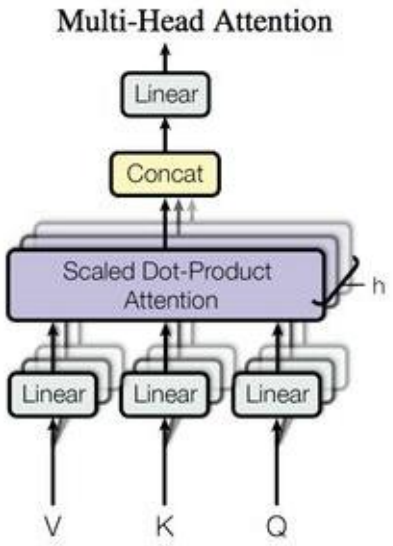


Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. arXiv preprint arXiv:1706.03762, 2017.

基于词的聚合表示

Multi-head self-attention:

在多个子空间中计算attention，
能考虑更多维度的语义信息。



$$MultiHead(Q,K,V) = Concat(head_1,\ldots,head_h)W^O$$

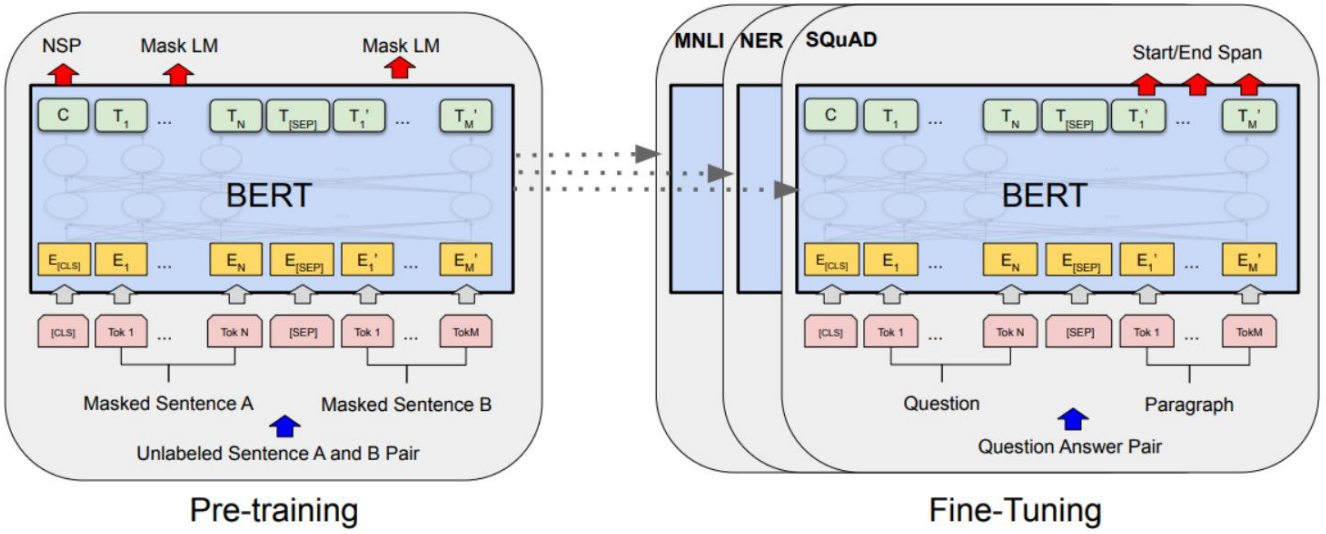
$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. arXiv preprint arXiv:1706.03762, 2017.

基于词的聚合表示

Bert:将Transformer（Encoder）在海量文本上预训练，然后：

- 可直接用来对新段落编码，得到其表示；
- 也可在其他具体任务上微调（finetune），以适应不同的任务：



Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.