



深度学习理论与实践

第8课 深度学习在NLP中的应用

主讲人 郑元春

中科院大数据挖掘与知识管理重点实验室
中科院虚拟经济与数据科学研究中心
从事机器学习与自然语言处理相关研究





NLP应用



文本表示学习



NNLM



文本分类应用



代码讲解



1. NLP应用

应用分类

研究类型

	词法与句法分析	语义分析	篇章分析
基础研究(6)	语言表示与深度学习	语言认知模型	知识图谱与计算
应用研(14)	文本分类与聚类	信息抽取	情感分析
自动文摘	信息检索	信息推荐与过滤	
自动问答	机器翻译	社会媒体处理	
语音技术	文字识别	多模态信息处理	
医疗健康信息处理	少数民族语言文字信息处理		

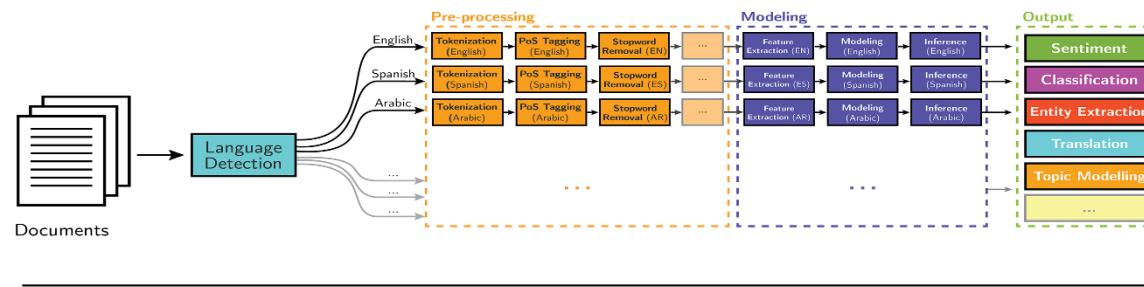


1. NLP应用

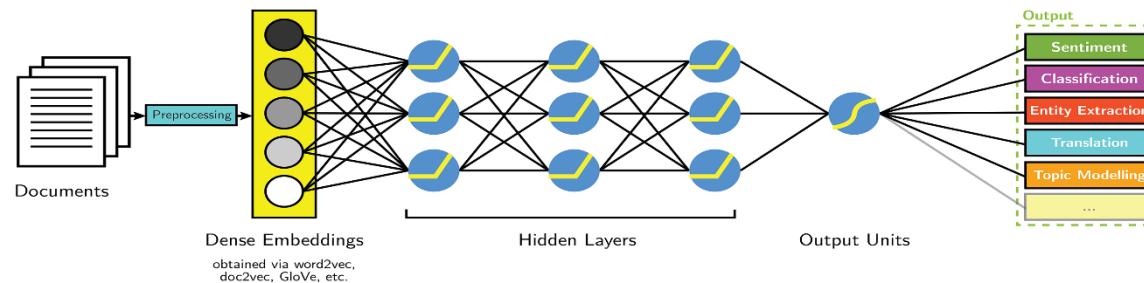
传统方法与基于神经网络的方法

- 传统的方法注重的是句法的表征和人工特征的构建，并不是语义构建。
- 深度学习拥有可表达性、可训练性、可泛化性特性。

Classical NLP



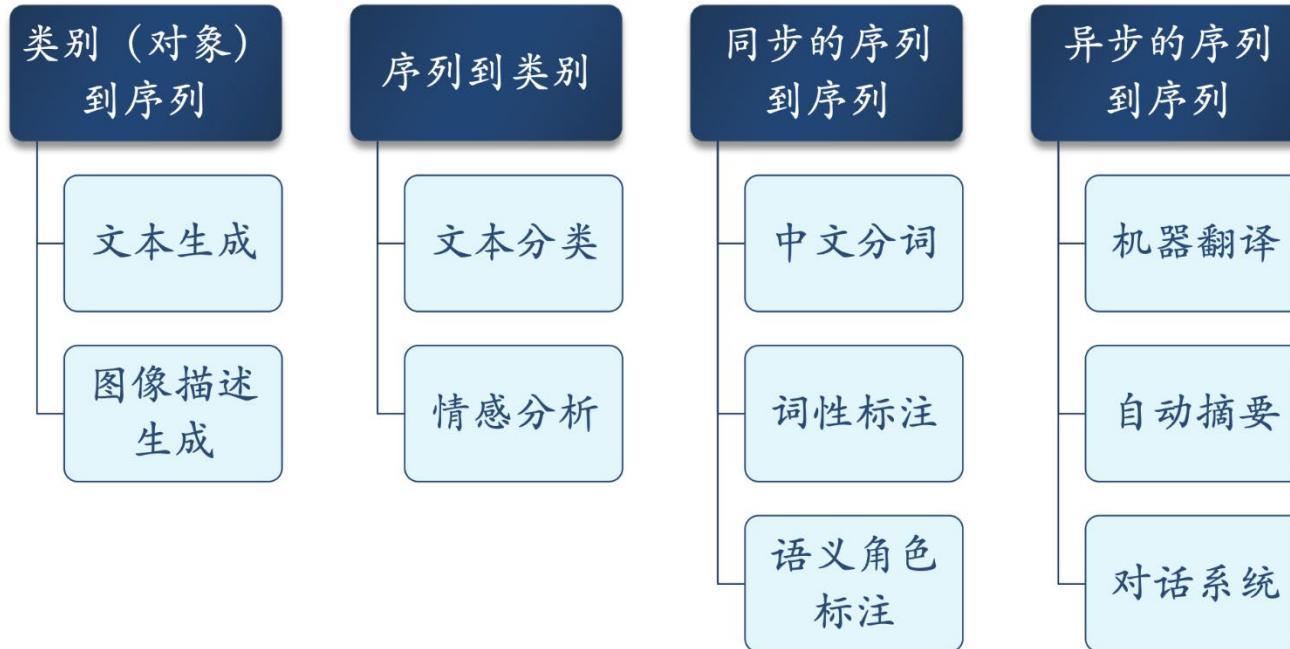
Deep Learning-based NLP





1. NLP应用

自然语言处理新范式





1. NLP应用

为什么文本处理更难

1. 配钥匙师傅：你配吗？
2. 食堂阿姨：你要饭吗？
3. 算命先生：你算什么东西？
4. 快递小哥：你是什么东西？
5. 垃圾分拣员：你是什么垃圾？
6. 滴滴司机：你搞清楚自己的定位了吗？
7. 理发师傅：你自己照照镜子看看自己。
8. 小区保安：你是谁？你从哪里来？到哪去？

生活中遇到的文本歧义性问题

不同的语言环境中的同形异构现象，按照具体语言环境才能有其语义

- 交叉歧义
 - 组合成
 - 的确实
- 组合歧义
 - 两 | 个 | 人 | 一起 过去、个人 | 问题
 - 从马 | 上 | 下来、马上 | 就来
- 句子级歧义
 - 白天鹅在水里游泳 白天？天鹅？
 - 该研究所获得的成果 研究？研究所？

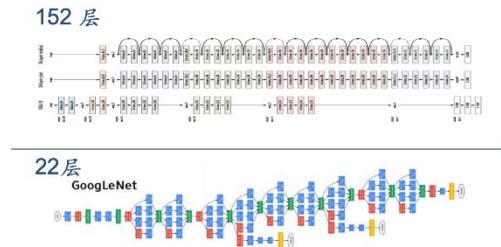
中文分词遇到的歧义性问题



1. NLP应用

为什么文本处理更难

	输入量	信息量	关系	底层特征
图像	二维像素集 200x200	黑白: 128-256 彩色: 3 (128-256)	欧氏空间	纹理, 形状, 色彩
文本	一维离散词符号序列 几千-几万个词	共250k (英文) 几千个词 (中文)	语法关系 句法关系 语义关系	句子长度, 句子在段落中的位置, 段落在文中中的位置



Results:

- golden retriever: 0.97293
- Tibetan mastiff: 0.01576
- Irish setter: 0.00364
- redbone: 0.00152
- standard poodle: 0.00127

计算机视觉中的深层网络模型

- 很多模型并不深
- LSTM+Attention对大多数NLP任务足够了
- 缺少大规模的标注数据
- 标注代价太高
- 无监督预训练与多任务学习



NLP应用



文本表示学习



NNLM



文本分类应用



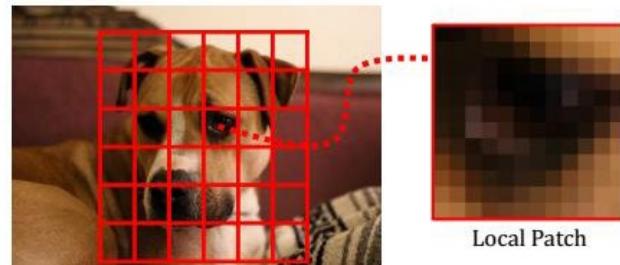
代码讲解



2. 文本表示学习

数据表示

数据表示是科学处理的第一步并且也是最重要的一步。高效简洁的数据表示不仅能够节省大量的计算资源，同时，还能直接蕴含数据的内部特性。不同的数据来源拥有不同的数据表示方法。自然界中存在的信号可以看成是连续的信号，比如说声和光等。



现在的计算机硬件系统是建立在数字电路基础之上的，只有0和1两种状态。不能存储模拟的信号，只能通过数字采样来尽可能的获取更多的信息，例如高清声源信号、高清图片、高清视频等。



2. 文本表示学习

概率建模

在自然语言中，语法与语义赋予了句子特别的含义。例如，“我爱吃苹果”和“苹果爱吃我”所包含的汉字都是一样的，但是其先后顺序导致了其表达的意思不一样。如果我们从整个句子的角度出来，我们一定希望是前者出现的概率要高于后者出现的概率，也就是赋予正常句子更大的概率。

给定一个含有 n 个单词的句子 $s = (w_1, w_2, \dots, w_n)$ ，将整个句子的构造过程看作是一个从左向右根据历史句子生成下一个单词的马尔可夫过程，那么 $p(w_1)$ 代表第一个单词出现的概率，用 $p(w_2|w_1)$ 代表在已经有 w_1 的条件下生成 w_2 的概率， $p(w_n|w_1^{n-1})$ 代表在前 $n - 1$ 个单词已经存在的情况下，生成第 n 个单词的概率。那么整句话的概率可以写成：

$$\begin{aligned} p(s) &= p(w_1^n) = p(w_1) \cdot p(w_2|w_1) \cdot p(w_3|w_1^2) \cdots p(w_n|w_1^{n-1}) \\ &= \prod_{i=1}^n p(w_i|w_{i-1}, \dots, w_1) \end{aligned}$$



2. 文本表示学习

概率建模

假设整个语料库有 V 个不重复的单词，那么对于当前句子中的单词 w_i 对应的上文词序列就有 $|V|^{i-1}$ 种不同组合情况，模型需要从中选出产生 w_i 概率最大的那种情况。随着 i 的增大，需要考虑的词序列组合情况呈指数增长，计算量也随之增加。为了使上下文的概率变得可计算，使用 N 元语法模型来对其进行简约映射。

将两个不同的历史组合情况 $h_1 = w_{i-1}, \dots, w_2, w_1$ 和 $h_2 = v_{k-1}, \dots, v_2, v_1$ 映射到同一个等价类中，当 h_1 和 h_2 最近的 $n-1$ 个词相同，也就是当 $(w_{i-n+2}, \dots, w_{i-1}) = (v_{k-n+2}, \dots, v_{k-1})$ 的时候，认为 $E(w_{i-1}, \dots, w_2, w_1) = E(v_{i-1}, \dots, v_2, v_1)$ ，满足上述条件的语言模型称为 n 元语法或者是 n 元文法 (n-gram)。

在实际的情况下，最常用的和能够承受的计算情况为 $n = 3$ ，此时的语言模型就是一个三元语法模型，句子中每个单词依赖于它前面相邻的两个单词。这可以看做是一个二阶的马尔可夫链。

$$p(s) = p(w_1^n) = p(w_1) \cdot p(w_2 | w_1) \cdot p(w_3 | w_1^2) \cdots p(w_n | w_{n-2}^{n-1}) = \prod_{i=3}^n p(w_1) \cdot p(w_2 | w_1) \cdot p(w_i | w_{i-1}, w_{i-2})$$



2. 文本表示学习

名词区别

- **分布假说**: 1954年Harris提出, 上下文相似的词, 其语义也相似。
- **分布表示(Distributional Representation)**: 基于分布假说的方法都可以称作是分布表示, 词的语义是用上下文来描述的。
- **局部表示(Local Representation)**: 向量中只有某一个维度存储了信息。
- **分布式表示(Distributed Representation)**: 把信息(可以指词义)分布存储在向量的各个维度中。通过矩阵降维或是神经网络的方式可以将语义分散存储到向量的各个维度中。

0	1	0	0	0
nlp	python	word	one-hot	ruby

one-hot representation

	50~300 dim			
python	0.52	0.21	0.37	...
ruby	0.48	0.21	0.33	...
word	0.05	0.23	0.06	...

distributed representation



2. 文本表示学习

表示学习方法

分布式表示方法主要有3类：基于矩阵的分布表示、基于聚类的分布表示、**基于神经网络的分布表示**。

尽管这些不同的分布表示方法使用了不同的手段来获取词的表示方法，但由于这些方法均是基于分布假说，他们的核心思想也是由两部分来组成的。

- 选择一种方式来描述上下文。
- 选择一种模型来刻画目标词与上下文的对应关系。

在一个句子中，我们统一的将需要预测的词称为**目标词**(target word)，将预测目标词用到的词汇称为**上下文**(context word)。根据目标词与上下文词汇的相对位置，又可以将语言模型分为AR(Autoregressive Language Modeling)和AE(Autoencoding Language Modeling)两种不同的方式。其中，AR模型只利用了历史词汇(上文词汇)来预测目标词，而AE则是同时利用了上下文词汇来预测目标词。

我的手机坏了，我打算____一部新手机。



NLP应用



文本表示学习



NNLM



文本分类应用



代码讲解



3. 神经网络语言模型

n-gram模型参数的复杂度

n-gram模型由于随着n的逐渐增大，会导致可能出现的组合数量成倍的增加。因此当n大于3的时候则无法很好的计算概率矩阵，而且这个概率矩阵也是很稀疏的形式。

n	n-gram 模型参数数量
n=1(unigram)	2×10^5
n=2(bigram)	4×10^{10}
n=3(trigram)	8×10^{15}
n=4	16×10^{20}

在模型效果方面，理论上是 n 越大越好。随着硬件的发展使得能够计算更加高级的语言模型(比如 $n > 10$)，但是随着 n 增大到一定的数值之后，模型所能带来的效果提升幅度将会逐渐变小。事实上，这里还涉及到一个**可靠性和可区别性**的问题。参数越多，可区别性越好，但同时参数的实例变少从而降低了可靠性，因此需要可靠性和可区别性之间进行折中。



3. 神经网络语言模型

计数模型与预测模型

N-gram模型的主要工作就是在语料中统计各种词串出现的次数以及平滑化处理，概率值计算完成之后就储存下来，下次在计算一个句子的概率时候只需要通过查找找到相关的概率参数，将他们连续相乘就好了。这种利用频率来模拟概率的方式为[Count-based](#)的模型。

在机器学习领域，则是采用一种逐渐学习的过程。构造一个目标函数，然后优化目标函数的参数，使得目标函数取得最小值，从而得到一组最优化的参数。最后，利用这组最优化的参数所对应的模型来进行预测。这种直接对概率进行建模的方式为[Predicted-based](#)的模型。

目标函数：

$$L = \prod_{w \in C} p(w | context(w))$$

最大似然函数：

$$L = \log \prod_{w \in C} p(w | context(w)) = \sum_{w \in C} \log p(w | context(w))$$



3. 神经网络语言模型

从有监督到无监督

利用神经网络进行训练的时候，最大的困难就是需要标注数据。对于语言模型来说，由于是对预测目标单词的概率进行最大化的建模，因此，天然地可以利用现有的文本数据将整个的训练过程转化为有监督学习的过程。

在用神经语言概率模型建模的时候，将概率转化为 $context(w)$ 对 w 的一个预测问题，目标是尽可能的通过上下文来预测准确目标词。这时目标词就可以转换成模型的标签，对 $F(w, Context(w), \theta)$ 建造的时候就可以通过 $softmax$ 的方式变成一个有监督学习。从有监督转华为无监督问题，其中的转变过程可以用下面的式子来表示出。

$$\frac{count(w_{k-n+1}^k)}{w_{k-n+1}^{k-1}} \Leftarrow p(w_k | w_{k-n+1}^{k-1}) \Rightarrow \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$



3. 神经网络语言模型

从有监督到无监督

利用神经网络进行训练的时候，最大的困难就是需要标注数据。对于语言模型来说，由于是对预测目标单词的概率进行最大化的建模，因此，天然地可以利用现有的文本数据将整个的训练过程转化为有监督学习的过程。

在用神经语言概率模型建模的时候，将概率转化为 $\text{context}(w)$ 对 w 的一个预测问题，目标是尽可能的通过上下文来预测准确目标词。这时目标词就可以转换成模型的标签，对 $F(w, \text{Context}(w), \theta)$ 建造的时候就可以通过 softmax 的方式变成一个有监督学习。从有监督转华为无监督问题，其中的转变过程可以用下面的式子来表示出。

$$\frac{\text{count}(w_{k-n+1}^k)}{w_{k-n+1}^{k-1}} \Leftarrow p(w_k | w_{k-n+1}^{k-1}) \Rightarrow \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$

这样就可以通过神经网络的方式来训练得到n-gram的表示方式，而不再是通过记录词汇或是短语的频率矩阵。同时由于采用了神经网络的方式，可以比count-based模型取更大的n值，使得模型更加逼近最终的句子概率表示。



3. 神经网络语言模型

神经语言模型的发展

基于神经网络语言模型建模的发展可以分为以下几个重要历史阶段：

1. Word Embedding

- NNLM(Neural Network Language Model)
 - HLBL
 - CW
 - Word2vec
 - Glove
-

3. Meta Embedding

- ConC
 - SVD
 - 1ToN
 - DME
 - CDME
-

2. Multiple Word Embedding

- Huang
 - Neelakantan
 - Liu
-

4. Pre-trained Language Model

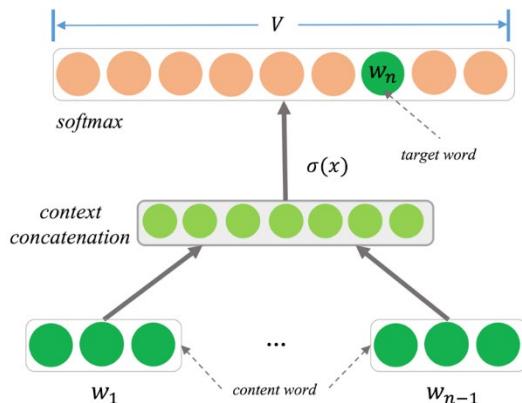
- Elmo
 - GPT
 - Bert
 - 其他
-



3. 神经网络语言模型

神经语言模型的发展-Word Embedding

- ① 百度的徐伟等人在2000年首次尝试使用神经网络来求解二元语言模型
- ② Bengio等人在2001年提出了神经语言模型 (**NNLM**)



目的: 使用前面 $n-1$ 个词预测第 n 个词

结构: 三层神经网络

概率归一化: 输出层使用的是 *softmax*.

$$\begin{cases} z_w = \tanh(WX_w + p) \\ y_w = Uz_w + q \end{cases}$$

$$p(w|context(w)) = p(y_w|X_w)$$

$$= \frac{\exp(y_w, X_w)}{\sum_{i=1}^N \exp(y_i, X_w)}$$



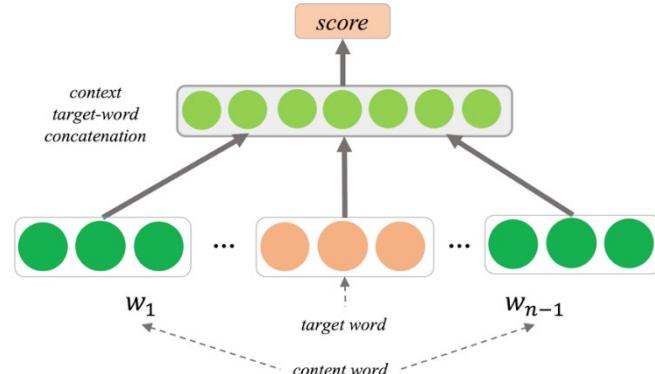
3. 神经网络语言模型

神经语言模型的发展-Word Embedding

- ① 百度的徐伟等人在2000年首次尝试使用神经网络来求解二元语言模型
- ② Bengio等人在2001年提出了神经语言模型 (**NNLM**)
- ③ 2007年, Mnih和Hinton提出了log双线性语言模型 (**LBLM**)。Minh等又提出**Hierarchical Softmax**方法, 将softmax复杂度降到了对数级别。
- ④ Collobert和Weston直接从训练词向量的角度出发, 提出了C&W模型。

$$\sum_{(w,c) \in \mathbb{D}} \sum_{w' \in V} \max(0, 1 - score(w, c) + score(w', c))$$

使用 w' 替换原始的句子中的单词 w 作为负样本





3. 神经网络语言模型

神经语言模型的发展-Word Embedding

- ① 百度的徐伟等人在2000年首次尝试使用神经网络来求解二元语言模型
- ② Bengio等人在2001年提出了神经语言模型 (**NNLM**)
- ③ 2007年，Mnih和Hinton提出了log双线性语言模型 (**LBLM**)。Minh等又提出**Hierarchical Softmax**方法，将softmax复杂度降到了对数级别。
- ④ Collobert和Weston直接从训练词向量的角度出发，提出了C&W模型。
- ⑤ Mikolov等人在2010年，提出了循环神经网络语言模型 (**RNNLM**)，使用循环神经网络对 $P(w_i|w_1^{i-1})$ 建模，能够使用更多的上文信息。
- ⑥ 2013年，Mikolov提出了**CBOW**和**Skip-gram**两个模型，使用**Hierarchical Softmax**和**Negative Sampling**两种算法加速求解模型。开源了词向量训练工具**word2vec**，词向量被广为人知。

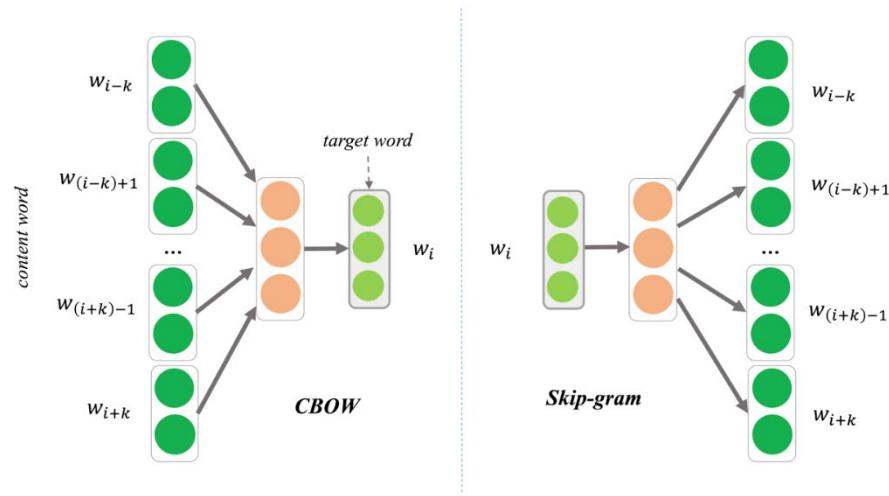


3. 神经网络语言模型

神经语言模型的发展-Word Embedding

CBOW: 通过上下文单词来预测目标词

$$\mathcal{L} = \sum_{w \in C} \log \prod_{j=2}^{l^w} \left\{ [\sigma(X_w^T \theta_{j-1}^w)]^{1-d_j^w} \cdot [1 - \sigma(X_w^T \theta_{j-1}^w)]^{d_j^w} \right\}$$



Skip-Gram: 使用目标词预测周围的上下文单词

$$\mathcal{L} = \sum_{w \in C} \log p(w | \text{context}(w))$$

$$\mathcal{L} = \sum_{w \in C} \log p(\text{context}(w) | w)$$

$$\mathcal{L} = \sum_{w \in C} \log \prod_{u \in c(w)} \prod_{j=2}^{l^u} \left\{ [\sigma(v(w)^T \theta_{j-1}^u)]^{1-d_j^u} \cdot [1 - \sigma(v(w)^T \theta_{j-1}^u)]^{d_j^u} \right\}$$



3. 神经网络语言模型

神经语言模型的发展-Word Embedding

- ① 百度的徐伟等人在2000年首次尝试使用神经网络来求解二元语言模型
- ② Bengio等人在2001年提出了神经语言模型 (**NNLM**)
- ③ 2007年，Mnih和Hinton提出了log双线性语言模型 (**LBLM**)。Minh等又提出**Hierarchical Softmax**方法，将softmax复杂度降到了对数级别。
- ④ Collobert和Weston直接从训练词向量的角度出发，提出了C&W模型。
- ⑤ Mikolov等人在2010年，提出了循环神经网络语言模型 (**RNNLM**)，使用循环神经网络对 $P(w_i|w_1^{i-1})$ 建模，能够使用更多的上文信息。
- ⑥ 2013年，Mikolov提出了**CBOW**和**Skip-gram**两个模型，使用**Hierarchical Softmax**和**Negative Sampling**两种算法加速求解模型。开源了词向量训练工具**word2vec**，词向量被广为人知。
- ⑦ 2014年，斯坦福语言小组提出**Glove** (Global Vectors for Word Representation) 模型，同时利用将矩阵分解(全局信息)和上下文窗口(局部信息的)的信息。



3. 神经网络语言模型

神经语言模型的发展-Multiple Word Embedding

问题：语言歧义、一词多义

思路：上下文能够决定当前目标词的具体涵义（建立在分布假说：**词的语义有其上下文决定**）

举例：

He is a movie star in Hollywood

At night, star shines in the sky

在左边的句子中，由于上下文中有movie和Hollywood等关键词，所以这里的star代表着电影明星的意思。

在右边的句子中，shines和sky则说明这里的star指代的天上的星星。



3. 神经网络语言模型

神经语言模型的发展-Multiple Word Embedding

作者	年份	题目	要点	优点	缺点
Eric Huang	2012	Improving word representations via global context and multiple word prototypes	词上下文聚类 聚类中心re-label	可以对词的不同原型进行分别训练	对每个词聚类太耗时
Tian Fei	2014	A Probabilistic Model for Learning Multi-Prototype Word Embeddings	隐变量 EM求解	使用隐变量求解	算法不稳定
Neelakantan	2014	Efficient Non-parametric Estimation of Multiple Embeddings per Word in Vector Space	保存不同语境词向量的上下文信息	逐渐学习 保存上下文信息	上下文训练没有使用优化方法
Antonio	2016	Word embeddings and recurrent neural networks based on Long-Short Term Memory nodes in supervised biomedical word sense disambiguation	LSTM 解决歧义问题	循环神经网络 上文时段更长	模型复杂

Li	2015	Do Multi-Sense Embeddings Improve Natural Language Understanding?	探究现存的多语境词向量学习的有效性
Tian Fei	2014	Real Multi-Sense or Pseudo Multi-Sense - An Approach to Improve Word Representation	歧义词和伪歧义词对词向量表示的作用

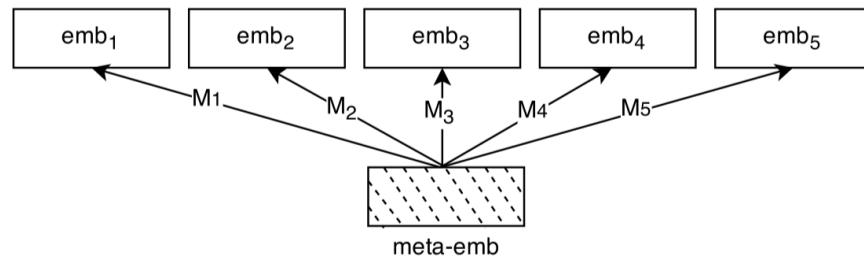


3. 神经网络语言模型

神经语言模型的发展-Meta Embedding

通过不同模型和不同语料训练出来的词向量的向量空间是不同的，而且包含的词汇量也可能不一致。有的模型利用的是AR的建模方式，有的模型利用的是AE的建模方式，因此训练出来的词向量其语义表示能力不同。利用Meta-embedding则是将已经存在的词向量进行元表示学习，映射到同一个向量空间，吸取众长。

- ① ConC
- ② SVD
- ③ 1ToN
- ④ DME
- ⑤ CDME





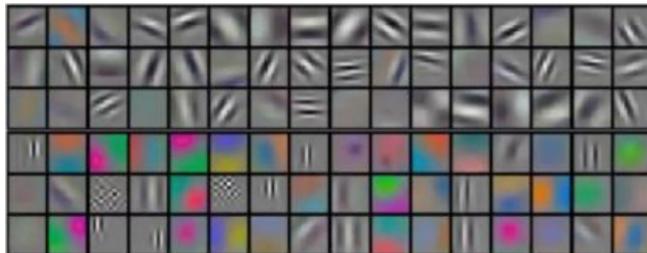
3. 神经网络语言模型

神经语言模型的发展-Pre-trained Language Model

神经语言模型的思路和模型构建陷入了一个固有模式：训练出词向量并将词向量存储下来，以后使用的时候在将这些词向量load进新的NLP任务当作初始化word embedding。（在这一过程中模型是丢弃的）

最大的问题是：词向量很难解决多义性问题

图像处理领域预训练的兴起，将一个模型进行充分的训练之后，模型的底层可以学习到图像的基本信息，比如纹理和轮廓等基本特征。将这部分模型移植到其他的图像领域作为模型的底层特征抽取，发现可以实现很好的性能。最大的好处就是不用在目标领域再重复进行训练，只需要简单的微调就可以使用了。



在ImageNet上进行图像分类训练之后，可以发现在底层的卷积层中，第一个卷积核对纹理信息敏感，而第二个卷积核对色彩信息敏感，这种能力对大部分图像任务是共通的。



3. 神经网络语言模型

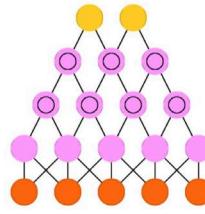
神经语言模型的发展-Pre-trained Language Model

神经语言模型的思路和模型构建陷入了一个固有模式：训练出词向量并将词向量存储下来，以后使用的时候在将这些词向量load进新的NLP任务当作初始化word embedding。（在这一过程中模型是丢弃的）

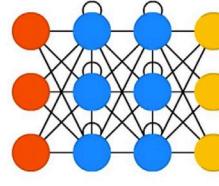
最大的问题是：词向量很难解决多义性问题

图像处理领域预训练的兴起，将一个模型进行充分的训练之后，模型的底层可以学习到图像的基本信息，比如纹理和轮廓等基本特征。将这部分模型移植到其他的图像领域作为模型的底层特征抽取，发现可以实现很好的性能。最大的好处就是不用在目标领域再重复进行训练，只需要简单的微调就可以使用了。

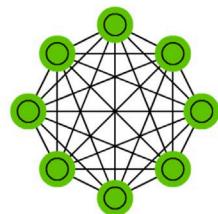
预训练语言模型和Transformer的兴起直接导致了Pre-trained Model的出现



CNN



RNN



Transformer

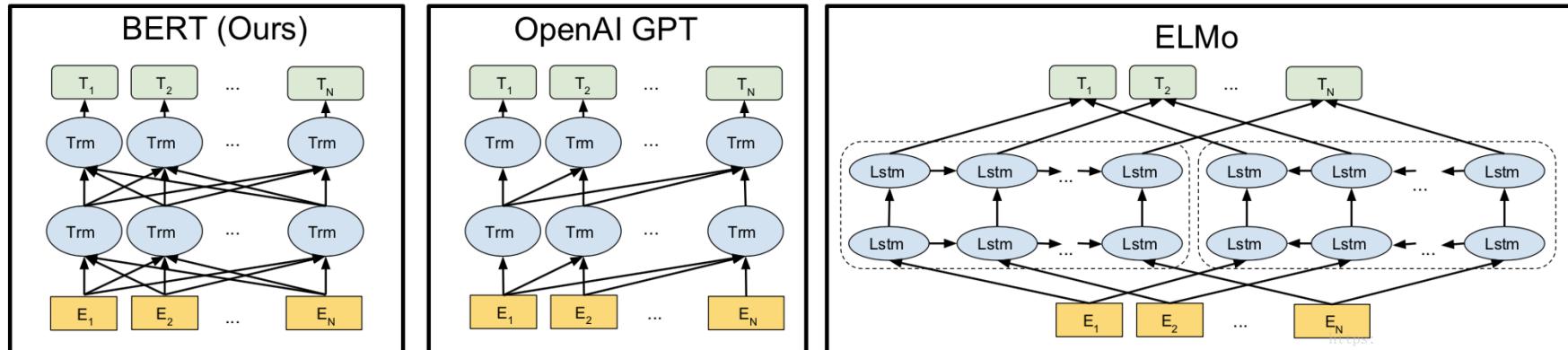


3. 神经网络语言模型

神经语言模型的发展-Pre-trained Language Model

Elmo是双向LSTM语言模型，但是实际上是利用了两个分开的前向和后向LSTM。GPT也是单向的，两者从本质上来说都是AR思想的语言建模。

Bert是在GPT的基础之上将Transformer搭建成了双向的语言模型，使用的是Transformer编码器。





NLP应用



文本表示学习



NNLM



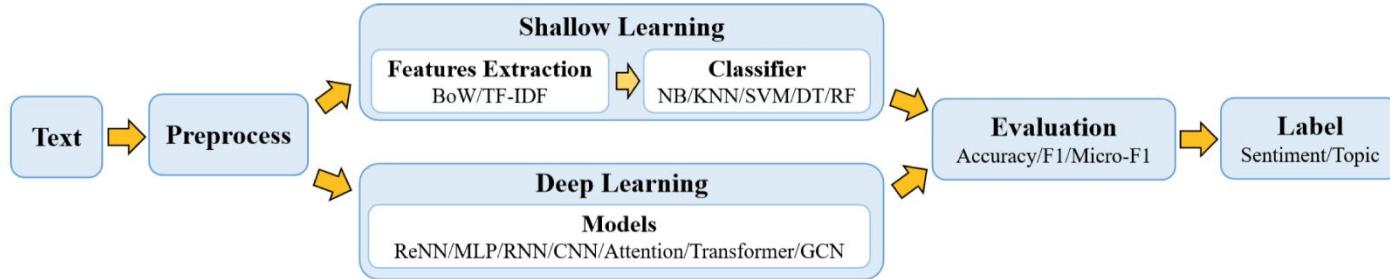
文本分类应用



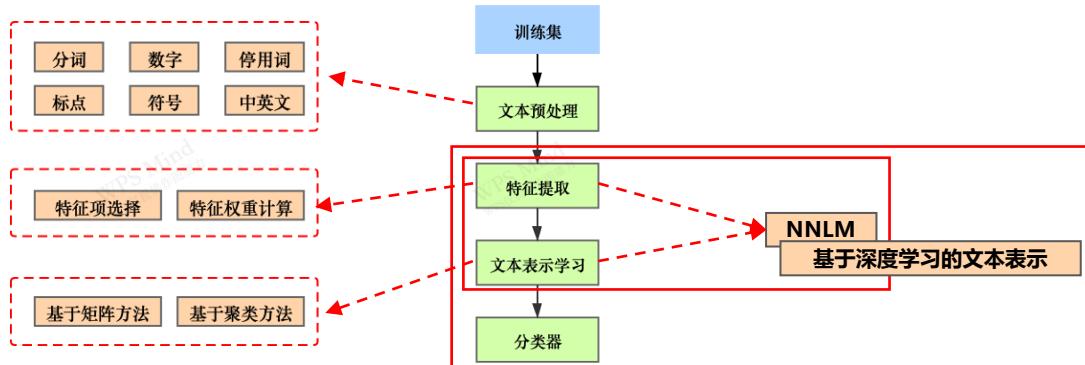
代码讲解



4. 文本分类应用

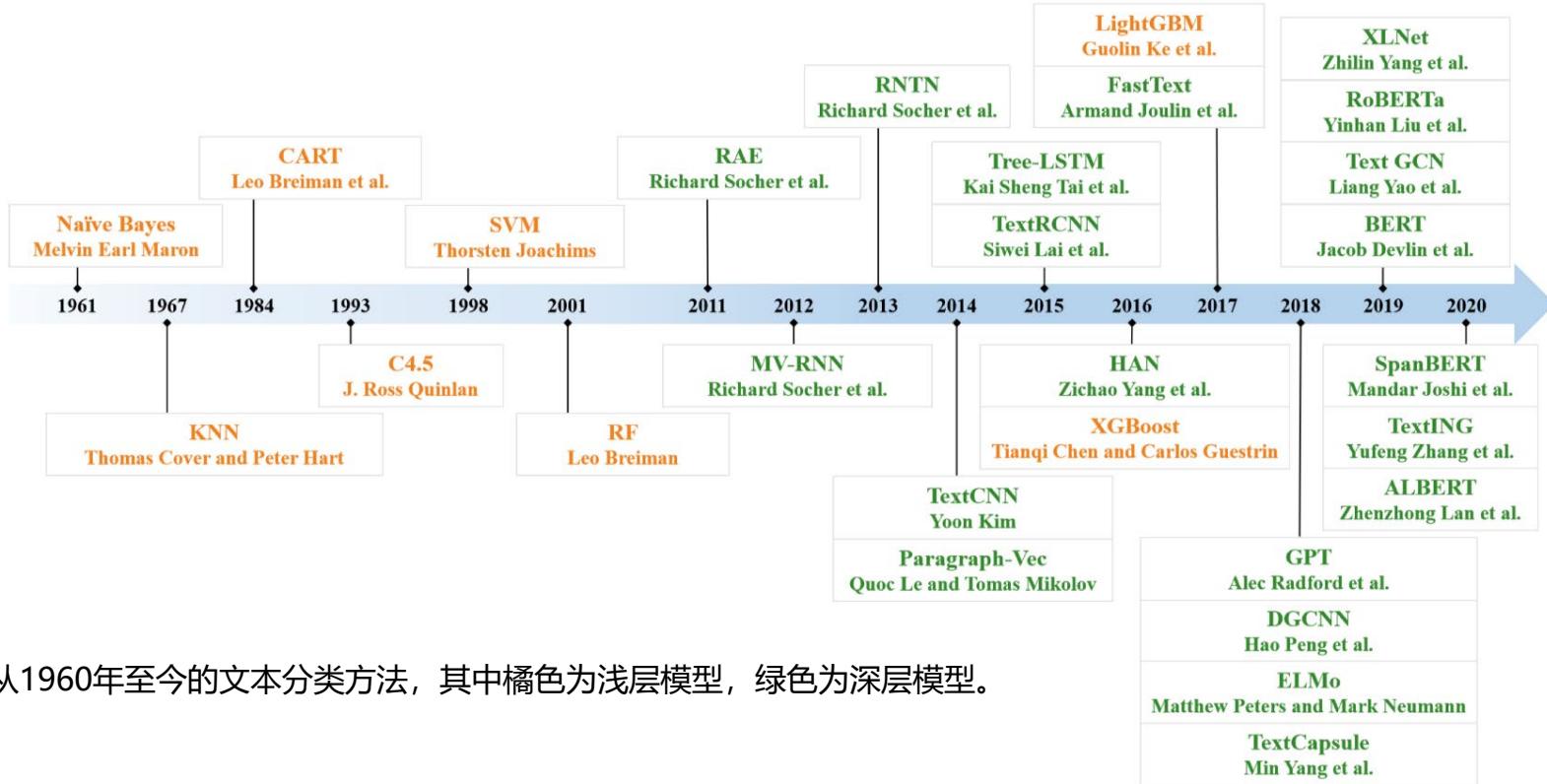


图：浅层模型中文本基本特征是非常重要的，模块之间分离并使用经典的机器学习算法。深度模型中采用端-端的方式来处理，可以自动提取特征。





4. 文本分类应用



从1960年至今的文本分类方法，其中橘色为浅层模型，绿色为深层模型。



4. 文本分类应用

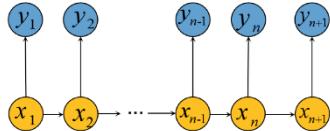
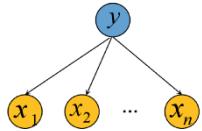
Year	Method	Venue	Applications	Citations
1961	NB [10]	JACM	TL	612
1967	KNN [11]	IEEE Trans.	-	12152
1984	CART [22]	Wadsworth	-	45967
1993	C4.5 [24]	Morgan Kaufmann	-	37847
1995	AdaBoost [26]	EuroCOLT	-	19372
1998	SVM [12]	ECML	-	10770
2001	RF [15]	Mach. Learn.	-	60249
2011	RAE [29]	EMNLP	SA, QA	1231
2012	MV-RNN [31]	EMNLP	SA	1141
2013	RNTN [33]	EMNLP	SA	3725
2014	Paragraph-Vec [35]	ICML	SA, QA	5679
2014	DCNN [7]	ACL	SA, QA	2433
2014	TextCNN [18]	EMNLP	SA, QA	7171
2015	TextRCNN [39]	AAAI	SA, TL	1141
2015	DAN [41]	ACL	SA, QA	467
2015	Tree-LSTM [2]	ACL	SA	1761
2015	CharCNN [5]	NeurIPS	SA, QA, TL	2114
2016	XGBoost [16]	KDD	QA	6187
2016	HAN [46]	NAACL	SA, TL	1889
2016	Multi-Task [48]	IJCAI	SA	410
2016	LSTMN [50]	EMNLP	SA	449
2017	LightGBM [17]	NeurIPS	QA	1065

2017	FastText [53]	EACL	SA, TL	1954
2017	Miyato et al. [55]	ICLR	SA	246
2017	TopicRNN [57]	ICML	SA	113
2017	DPCNN [59]	ACL	SA, TL	156
2017	IAN [61]	IJCAI	SA	222
2017	DeepMoji [63]	EMNLP	SA	260
2017	RAM [65]	EMNLP	SA	225
2018	ELMo [66]	NAACL	SA, QA, NLI	3722
2018	DGCNN [68]	TheWebConf	TL	81
2018	ULMFIt [70]	ACL	SA, TL, News	819
2018	LEAM [72]	ACL	TL, News	87
2018	SGM [74]	COLING	TL	42
2018	SGNN [76]	IJCAI	EP	26
2018	TextCapsule [78]	EMNLP	SA, QA, TL	118
2018	MGAN [80]	EMNLP	SA	46
2019	TextGCN [6]	AAAI	SA, TL	114
2019	BERT [19]	NAACL	SA, QA	5532
2019	MT-DNN [83]	ACL	SA, NLI	186
2019	XLNet [85]	NeurIPS	SA, QA, NC	652
2019	RoBERTa [87]	arXiv	SA, QA	203
2020	ALBERT [89]	ICLR	SA, QA	197
2020	SpanBERT [91]	TACL	QA	63

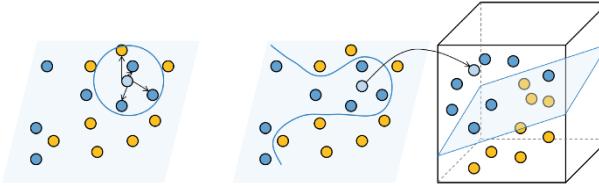
从1960年至今的文本分类方法及其引用情况



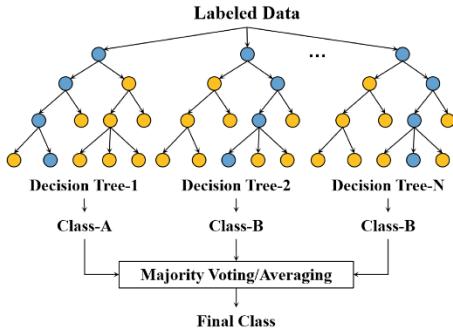
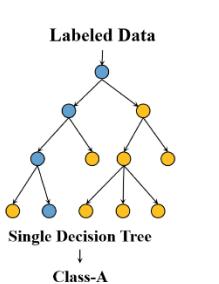
4. 文本分类应用



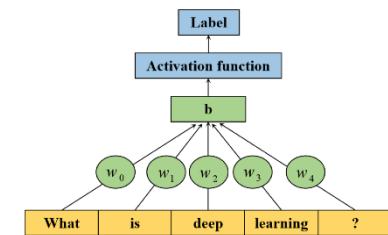
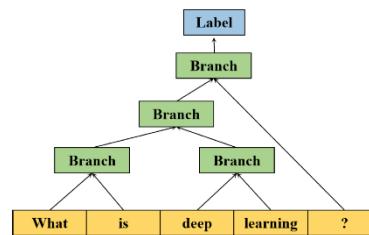
NB(朴素贝叶斯)与HMM(隐马尔可夫)方法



KNN(K最邻近)与SVM(支持向量机)方法



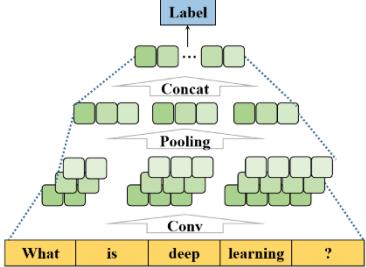
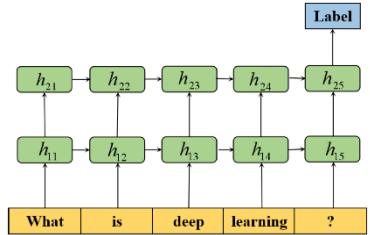
DT(决策树)与RF(随机森林)方法



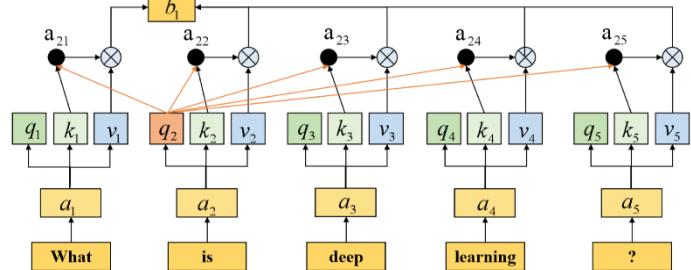
ReNN(递归神经网络)与MLP(全连接神经网络)方法



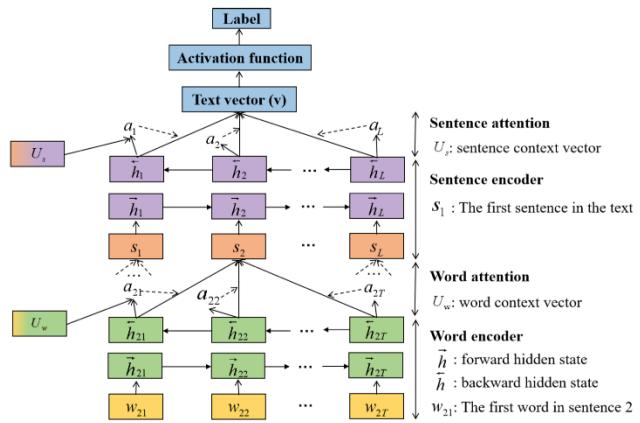
4. 文本分类应用



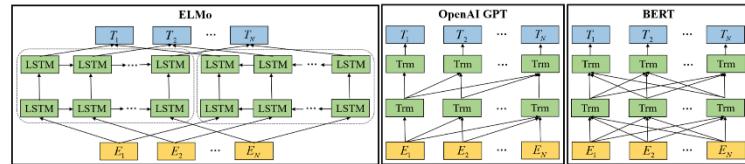
RNN(循环神经网络)与CNN(卷积神经网络)方法



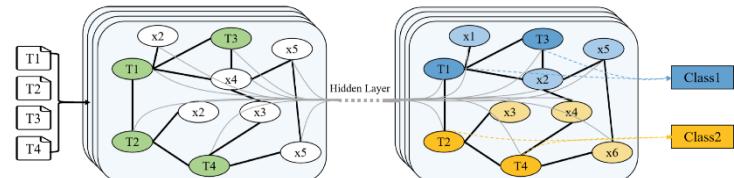
Attention(注意力网络)方法



HAN(分层注意力网络)方法



预训练语言模型方法



GCN(图神经网络)方法



4. 文本分类应用

TextCNN结构图

词向量	卷积操作	池化操作	拼接	全连接
(7×5)	$3 \times (2 \times 5)$	3×1	9×1	
	$3 \times (3 \times 5)$	3×1		
	$3 \times (4 \times 5)$	3×1		

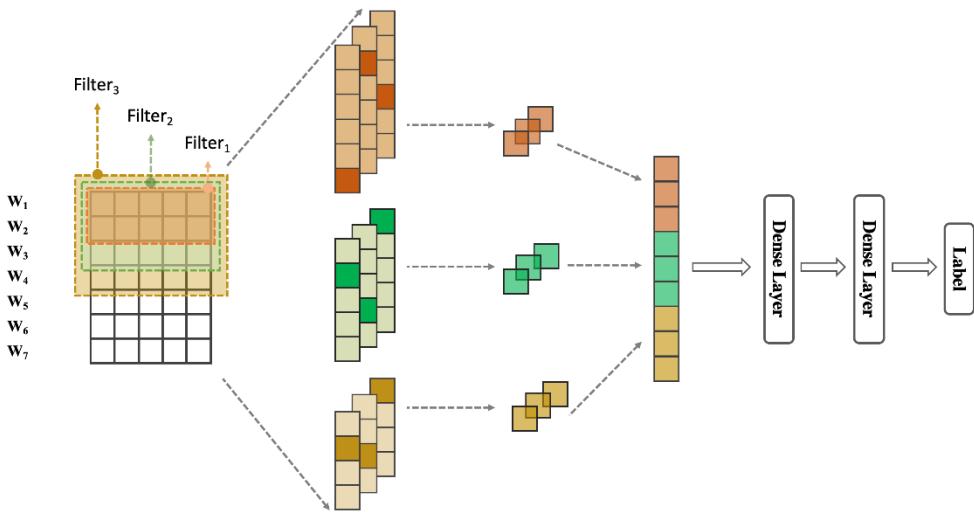
➤ 一维卷积(只抽语义关键信息)

➤ 预训练模型

- 随机词向量
- 固定预训练词向量
- 微调预训练词向量
- 多通道预训练词向量

➤ 微调方案

多通道和微调效果最好，但是使用预训练模型，然后进行微调是性价比最高的。



Model	MR	SST-1	SST-2	Subj	TREC	CR	MPQA
CNN-rand	76.1	45.0	82.7	89.6	91.2	79.8	83.4
CNN-static	81.0	45.5	86.8	93.0	92.8	84.7	89.6
CNN-non-static	81.5	48.0	87.2	93.4	93.6	84.3	89.5
CNN-multichannel	81.1	47.4	88.1	93.2	92.2	85.0	89.4



4. 文本分类应用

Model	Design	Metrics	Datasets
ReNN	recursive autoencoders [29]	Accuracy	MPQA, MR, EP
	recursive neural network [31]	Accuracy, F1	MR
	richer supervised training [33]	Accuracy	Sentiment Treebank
MLP	multiple recursive layers [116]	Accuracy	SST-1, SST-2
	a deep unordered model [41]	Accuracy, Time	RT, SST, IMDB
RNN	paragraph vector [35]	Error Rate	SST, IMDB
	tree-structured topologies [2]	Accuracy	SST-1, SST-2
	a memory cell [3]	Accuracy	SST
	RCNN and a max-pooling layer [39]	Accuracy, Macro - F1	20NG, Fudan, ACL, SST-2
	multi-timescale [8]	Accuracy	SST-1, SST-2, QC, IMDB
	embeddings of text regions [117]	Error Rate	IMDB, Elec, RCV1, 20NG
	2DCNN [118]	Accuracy	SST-1, SST-2, Subj, TREC, etc.
	multi-task [48]	Accuracy	SST-1, SST-2, Subj, IMDB
	distant supervision [63]	Accuracy	SS-Twitter, SE1604, etc.
	global dependencies [57]	Error Rate	IMDB
CNN	virtual adversarial training [55]	Error Rate	IMDB, DBpedia, RCV1, etc.
	capsule [119]	Accuracy	MR, SST-1, Hospital Feedback
	basic CNN [18]	Accuracy	MR, SST-1, SST-2, Subj, etc.
	dynamic k-Max pooling [7]	Accuracy	MR, TREC, Twitter
	character-level [5]	Error Rate	AG, Yelp P, DBpedia, etc.
	preceding short texts [9]	Accuracy	DSTC 4, MRDA, SwDA
	extreme multi-label [120]	P@k, DCG@k, etc.	EUR-Lex, Wiki-30K, etc.
	deep pyramid CNN [59]	Error Rate	AG, DBpedia, YelpP, etc.
	knowledge base [121]	Accuracy	TREC, Twitter, AG, Bing, MR
	8aARbit character encoding [122]	Accuracy	Geonames toponyms, etc.
GNN	dynamic routing [78]	Accuracy	Subj, TREC, Reuters, etc.
	hierarchical relations [123]	Micro - F1, Macro - F1, etc.	RCV1, Amazon670K
	meta-learning [124]	Accuracy	20NG, RCV, Reuters-2157, etc.

Attention	hierarchical attention [46]	Accuracy	Yelp,F, IMDB, YahooA, Amz.F
	add bilingual BiLSTM [125]	Accuracy	NLP&CC 2013 [126]
	intra-attention mechanism [50]	Accuracy	SST-1
	two-way attention mechanism [127]	P, MAP, MRR	TREC-QA, WikiQA, etc.
	Inner-Attention mechanism [128]	Accuracy	SNLI
	cross-attention mechanism [129]	F1	WebQuestion
	self-attention sentence embedding [130]	Accuracy	Yelp, Age
	sequence generation model [74]	HL, Micro - F1	RCV1-V2, AAPD
	deep contextualized representation [66]	Accuracy, F1	SQuAD, SNLI, SRL, SST-5, etc.
	a label tree-based model [131]	P@k, N@k, PSP@k	EUR-Lex, Amazon-670K, etc.
Trans	knowledge powered attention [132]	Accuracy	Weibo, Product Review, etc.
	bi-directional block self-attention [133]	Accuracy, Time	CR, MPQA, SST-1, SUBJ, etc.
	deep contextualized representation [66]	Accuracy	SQuAD, SNLI, SST-5
	bidirectional encoder [19]	Accuracy	SST-2, QQP, QNLI, CoLA
	multi-label legal text [134]	P@K, RP@K, R@K, etc.	EUR-LEX
	fine-tune BERT [135]	Error Rate	IMDB, TREC, DBpedia, etc.
	autoregressive pretraining [85]	DNCG@K, EM, F1, etc.	IMDB, Yelp-2, AG, MNLI, etc.
	modifications on BERT [87]	SQuAD, MNLI-m, SST-2	F1, Accuracy
	improvement of BERT [89]	F1, Accuracy	SST, MNLI, SQuAD
	graph-CNN for multi-label text [68]	Micro - F1, Macro - F1, etc.	RCV1, NYTimes
GNN	build a heterogeneous graph [6]	Accuracy	20NG, Ohsuned, R52, R8, MR
	removing the nonlinearities [136]	Accuracy, Time	20NG, R8, R52, Ohsuned, MR
	a text level graph [137]	Accuracy	R8, R52, Ohsuned
	hierarchical taxonomy-aware [138]	Micro - F1, Macro - F1	RCV1, EUR-Lex, etc.
	graph attention network-based [139]	Micro - F1, HL	Reuters-21578, RCV1-V2, etc.

方法分类及对应的评测指标与数据集



4. 文本分类应用

C: 类别数目

L: 句子平均长度

N: 数据集大小

SA: 情感分析

QA: 问题回答

NC: 新闻分类

TL: 话题标注

DAC: 对话行为分类

Model	Sentiment					News		Topic		NLI
	MR	SST-2	IMDB	Yelp.P	Yelp.F	Amz.F	20NG	AG	DBpedia	SNLI
RAE [29]	77.7	82.4	-	-	-	-	-	-	-	-
MV-RNN [31]	79	82.9	-	-	-	-	-	-	-	-
RNTN [33]	75.9	85.4	-	-	-	-	-	-	-	-
DCNN [7]	-	86.8	89.4	-	-	-	-	-	-	-
Paragraph-Vec [35]	-	87.8	92.58	-	-	-	-	-	-	-
TextCNN [18]	81.5	88.1	-	-	-	-	96.49	-	-	-
TextRCNN [39]	-	-	-	-	-	-	-	-	-	-
DAN [41]	-	86.3	89.4	-	-	-	-	-	-	-
Tree-LSTM [2]	-	88	-	-	-	-	-	-	-	-
CharCNN [5]	-	-	95.12	62.05	-	-	90.49	98.45	-	-
HAN [46]	-	-	49.4	-	-	63.6	-	-	-	-
SeqTextRCNN [9]	-	-	-	-	-	-	-	-	-	-
oh-2LSTMp [117]	-	-	94.1	97.1	67.61	-	86.68	93.43	99.16	-
LSTMNN [50]	-	87.3	-	-	-	-	-	-	-	-
Multi-Task [48]	-	87.9	91.3	-	-	-	-	-	-	-
BLSTM-2DCNN [118]	82.3	89.5	-	-	-	-	96.5	-	-	-
TopicRNN [57]	-	-	93.72	-	-	-	-	-	-	-
DPCNN [59]	-	-	-	97.36	69.42	65.19	-	93.13	99.12	-
KPCNN [121]	83.25	-	-	-	-	-	-	88.36	-	-
RAM [65]	-	-	-	-	-	-	-	-	-	-
RNN-Capsule [119]	83.8	-	-	-	-	-	-	-	-	-
ULMFIT [70]	-	-	95.4	97.84	71.02	-	-	94.99	99.2	-
LEAM [72]	76.95	-	-	95.31	64.09	-	81.91	92.45	99.02	-
TextCapsule [78]	82.3	86.8	-	-	-	-	-	92.6	-	-
TextGCN [6]	76.74	-	-	-	-	-	86.34	67.61	-	-
BERT-base [19]	-	93.5	95.63	98.08	70.58	61.6	-	-	-	91.0
BERT-large [19]	-	94.9	95.79	98.19	71.38	62.2	-	-	-	91.7
MT-DNN [83]	-	95.6	83.2	-	-	-	-	-	-	91.5
XLNet-Large [85]	-	96.8	96.21	98.45	72.2	67.74	-	-	-	-
XLNet [85]	-	97	-	-	-	-	-	95.51	99.38	-
RoBERTa [87]	-	96.4	-	-	-	-	-	-	-	92.6

Datasets	C	L	N	Related Papers	Sources	Applications
MR	2	20	10,662	[18] [7] [78] [6]	[174]	SA
SST-1	5	18	11,855	[33] [18] [2] [3] [50]	[175]	SA
SST-2	2	19	9,613	[33] [18] [8] [48] [19]	[176]	SA
Subj	2	23	10,000	[18] [48] [78]	[177]	QA
TREC	6	10	5,952	[18] [7] [8] [121]	[178]	QA
CR	2	19	3,775	[18] [78]	[179]	QA
MPQA	2	3	10,606	[29] [18] [133]	[180]	SA
Twitter	3	19	11,209	[7] [121]	[181]	SA
EP	5	129	31,675	[29]	[182]	SA
IMDB	2	294	50,000	[35] [41] [8] [48] [55] [85]	[183]	SA
20NG	20	221	18,846	[39] [117] [124] [6] [136]	[184]	NC
Fudan	20	2981	18,655	[39]	[185]	TL
AG News	4	45/7	127,600	[5] [59] [121] [78] [85]	[186]	NC
Sogou	6	578	510,000	[5]	[187]	NC
DBPedia	14	55	630,000	[5] [59] [55] [135]	[188]	TL
Yelp.P	2	153	598,000	[5] [59]	[189]	SA
Yelp.F	5	155	700,000	[5] [46] [59]	[189]	SA
YahooA	10	112	1,460,000	[5] [46]	[5]	TL
Amz.P	2	91	4,000,000	[131] [5]	[190]	SA
Amz.F	5	93	3,650,000	[5] [46] [131]	[190]	SA
DSTC 4	89	-	30,000	[9]	[191]	DAC
MRDA	5	-	62,000	[9]	[192]	DAC
SwDA	43	-	1,022,000	[9]	[193]	DAC
RCV1	103	240	807,595	[117] [123] [68] [134] [139]	[194]	NC
RCV1-V2	103	124	804,414	[74] [139]	[195]	NC
NLP&CC 2013	2	-	115,606	[125]	[126]	SA
SS-Twitter	2	-	2,113	[63]	[196]	SA
SS-Youtube	2	-	2,142	[63]	[196]	SA
SE1604	3	-	39,141	[63]	[197]	SA
Bing	4	20	34,871	[121]	[198]	TL
AAPD	54	163	55,840	[74] [139]	[75]	TL
Reuters	90	1	10,788	[78] [139]	[199]	NC
R8	8	66	7,674	[6] [136] [137]	[200]	NC
R52	52	70	9,100	[6] [136] [137]	[200]	NC
NYTimes	2,318	629	1,855,659	[68]	[201]	NC
SQuAD	-	5,000	5,570	[66] [66] [87] [89]	[202]	QA
WikiQA	-	873	243	[127]	[203]	QA
Ohsumed	23	136	7,400	[6] [136] [137]	[204]	TL
Amazon670K	670	244	643,474	[123] [131]	[205]	TL
EUR-Lex	3,956	1,239	19,314	[120] [131] [134] [138] [134]	[206]	TL



4. 文本分类应用

面临挑战

- 性能在不断的刷新，但是离真正的理解文本还差很远。
- 噪声样本导致决策置信度发生实质性改变，甚至导致逆转。
- 模型表示能力与鲁棒性。
- 垂直领域效果差强人意。

数据层面

- Zero shot/Few-shot Learning。
- 外部知识，如词典或知识图谱。
- 多标签文本分类。
- 属于众多的特殊领域，如法律文书，设备手册等。



语言模型



文本表示学习



NNLM



文本分类应用



代码讲解