文本分类

汇报人：主文浩、马翔雨、潘恋军、牟童瑶、周伊凡、李雪薇

指导老师：张华平、商建云

目录 | CONTENTS

1 文本分类介绍

- 文本分类简介
- 文本分类发展历史
- 文本分类流程介绍
- 文本分类主流算法
- 数据集介绍

汇报人：主文浩

文本分类介绍

文本分类概念

文本分类是指在给定分类体系下，根据文本内容自动确定文本类别的过程，是文本挖掘的一个重要内容。

20世纪90年代以前，占主导地位的文本分类方法一直是基于知识工程的分类方法，即由专业人员手工进行分类。近代，众多的统计方法和机器学习方法应用于自动文本分类。

文本分类介绍

文本分类类型及应用

- 文本分类根据文本语言不同：

  最常见的为中文文本分类以及英文文本分类及其他语种分类。

- 根据解决的问题不同：

  分为情感分析、新闻分类（体育、财经等）、垃圾邮件识别（二分类）、问答系统、自然语言推理。

- 根据分类模式不同：

  多分类、二分类、单文本多标签

文本分类发展历史

发展简史

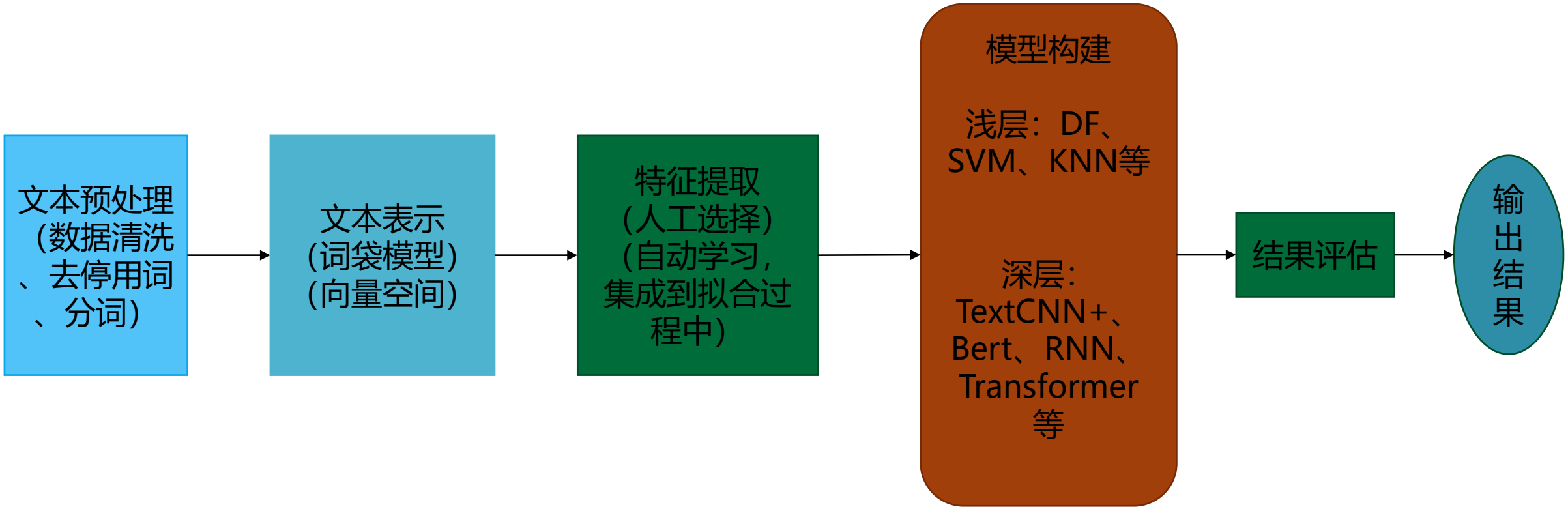
从1960年代到2010年代，基于浅层学习的文本分类模型占主导地位。

这里的浅层学习指的是基于统计的模型，例如朴素贝叶斯方法（NB），K近邻（KNN）和支持向量机（SVM）。与早期的基于规则的方法相比，该类方法的准确性和稳定性优势比较明显。

2010年代之后，深度学习模型。深度学习方法避免了人工设计规则和特征，可以自动从文本中挖掘出大量且丰富的语义表示。

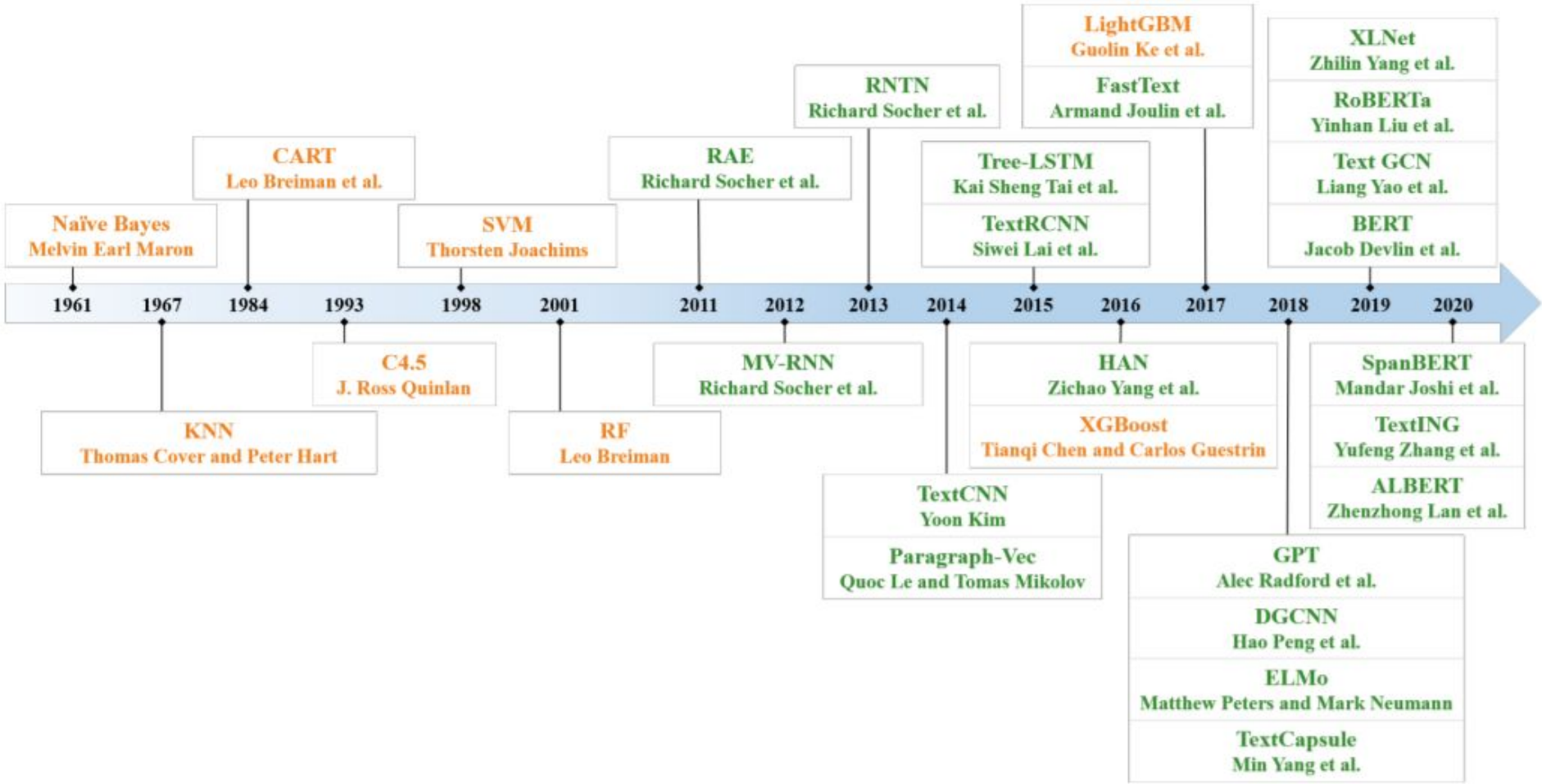
文本分类流程

文本分类处理流程



文本分类主流算法

主流算法介绍



数据集介绍

经典数据集介绍

- **20 Newsgroups (20NG)**
  20NG是新闻组文本数据集。它有20个类别，每个类别样本数目相同，一共包含18,846篇文本。

- **R8 and R52**
  R8和R52是路透社新闻的两个子集。
  R8有8个类别，分为2189个测试样本和5485个训练样本。
  R52有52个类别，分为6,532个训练样本和2,568个测试样本。

- **Movie Review (MR)**：电影评论数据集

- **THUCNews**：是根据新浪新闻RSS订阅频道2005~2011年间的历史数据筛选过滤生成，包含**74万篇新闻文档**（2.19 GB）。

2 基于传统机器学习的文本分类方法

汇报人：牟童瑶

传统机器学习方法概述

- 从20世纪60年代到21世纪10年代，基于传统机器学习的文本分类模型占据了主导地位。
- 与早期基于规则的方法相比，这类方法在准确性和稳定性方面具有明显的优势。

基于PGM的方法

- 概率图形模型(PGM)表示图中特征之间的条件依赖关系，如贝叶斯网络、隐马尔可夫网络。这种模型是概率论和图论的结合。
- 朴素贝叶斯(NB)，主要使用先验概率来计算后验概率。

$$P(c|x) = \frac{P(c)P(x|c)}{P(x)} = \frac{P(c)}{P(x)}\Pi_{i=1}^{d}P(x_i|c)$$

- 隐马尔可夫网络(HMM)，适用于序列文本数据。

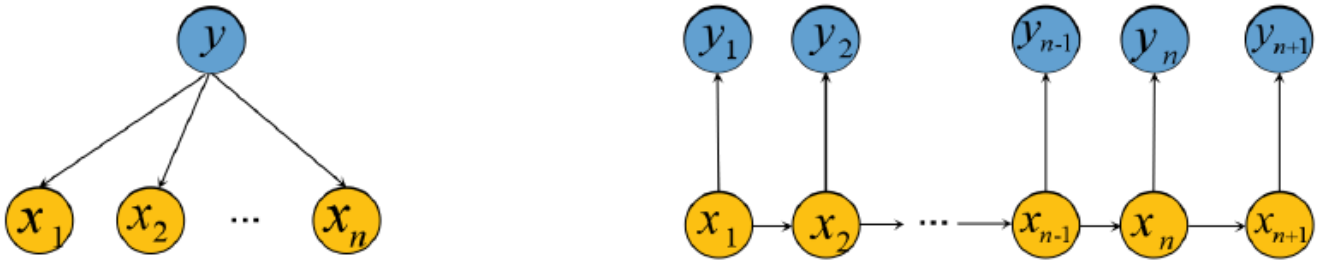
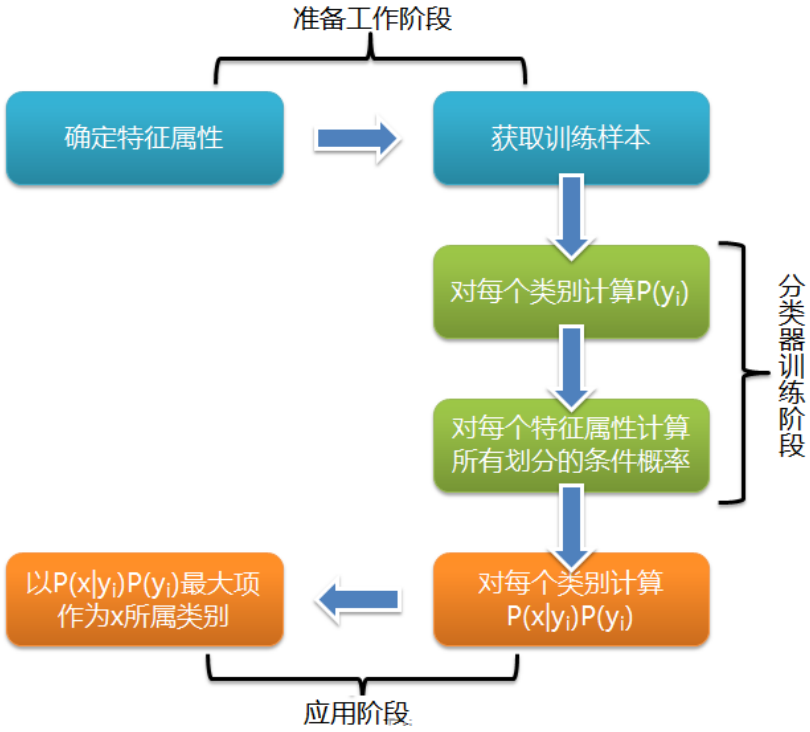
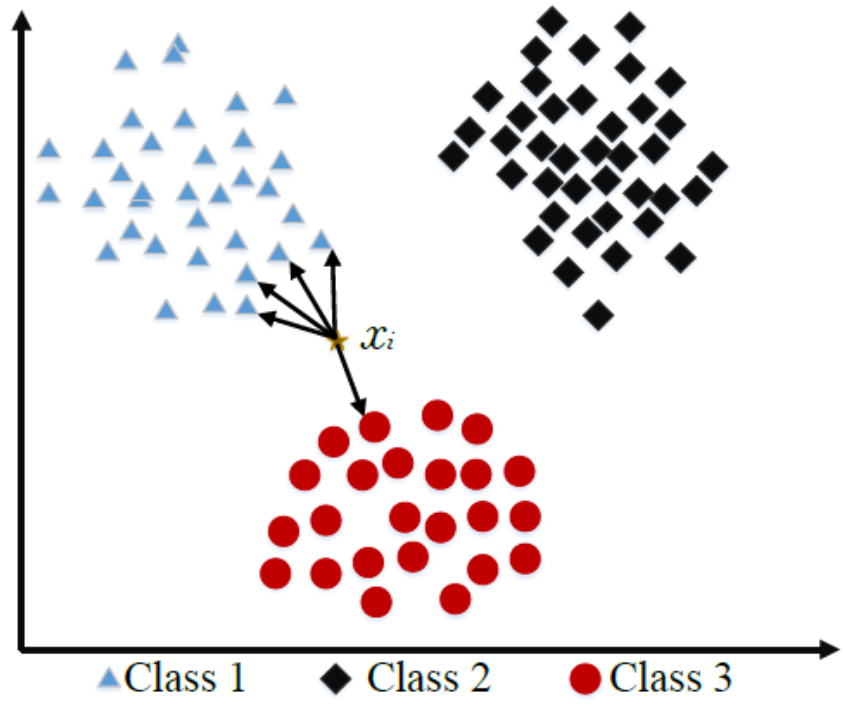


Fig. 3. The structure of NB (left) and the structure of HMM (right).



基于邻近性的方法

- 基于邻近性的分类器本质上使用基于距离的度量来进行分类。
- 最典型的是K近邻分类（KNN）,核心思想是在目标样本的K个最近邻样本上寻找样本数量最多的类别，对未标记样本进行分类。



A architecture of k-nearest Neighbor (KNN) model for the 2D data set and three classes.

基于SVM的方法

- 支持向量机（SVM, Support Vector Machine），用来解决二分类问题，主要原理是在搜索空间中确定能最有效地分离不同类别的超平面，其目标是要找到使超平面与两类训练集的距离最大的最大间隔超平面。

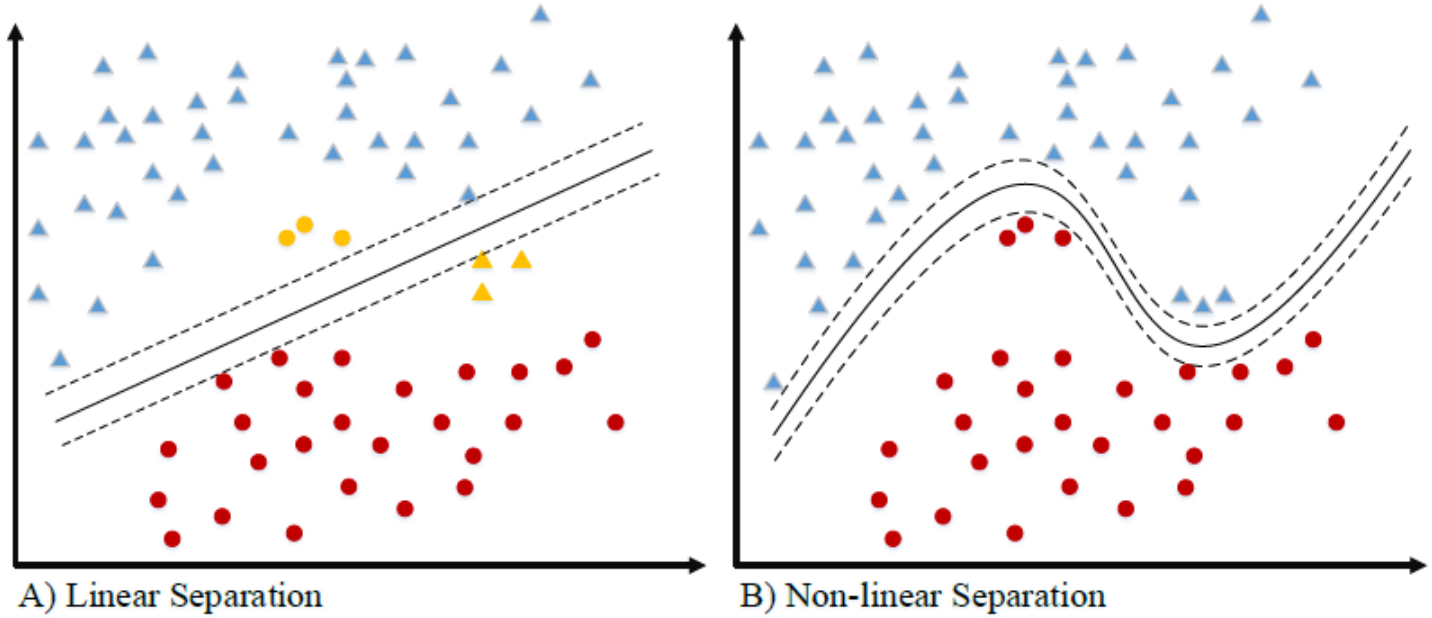
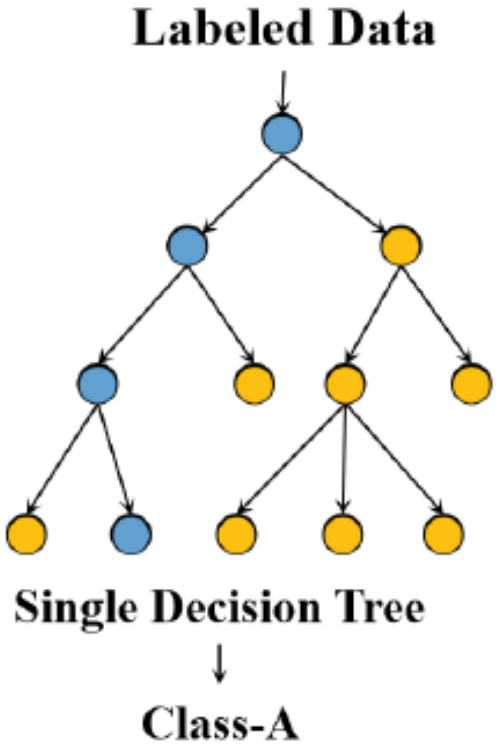


Figure 12. This figure shows the linear and non-linear Support Vector Machine (SVM) for a 2D data set (for text data we have thousands of dimensions). The red is class 1, the blue color is class 2 and yellow color is miss-classified data points.

基于DT的方法

- 决策树（Decision Trees, DT）本质上是数据空间的层次分解，一般可以分为树的构建和树的修剪两个阶段，构建决策树是为了确定类和属性之间的相关性，进一步用于预测未知类型的样本类别；剪枝策略有助于降低噪声的影响。
- 决策树的生成算法有ID3, C4.5和C5.0等。
- 局限性：无法应对爆炸式增长的数据量。



集成的方法

- 集成算法旨在聚合多种算法的结果，以获得更好的性能和解释。
  - 传统集成算法——Random forest (RF)
  - 基于增强的集成算法——AdaBoost、XGBoost
  - 基于叠加的集成算法——Stacking

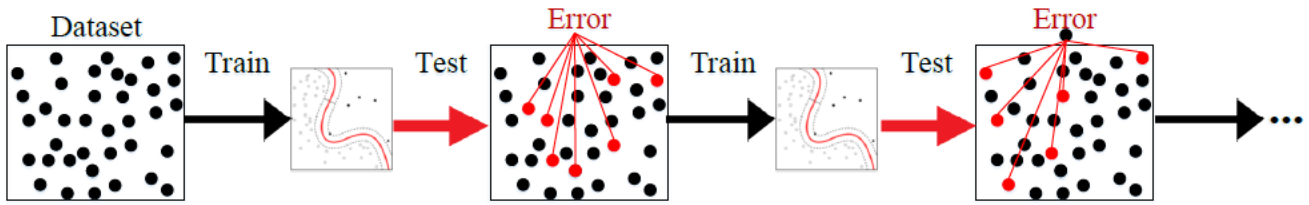


Figure 9. This figure is the boosting technique architecture.

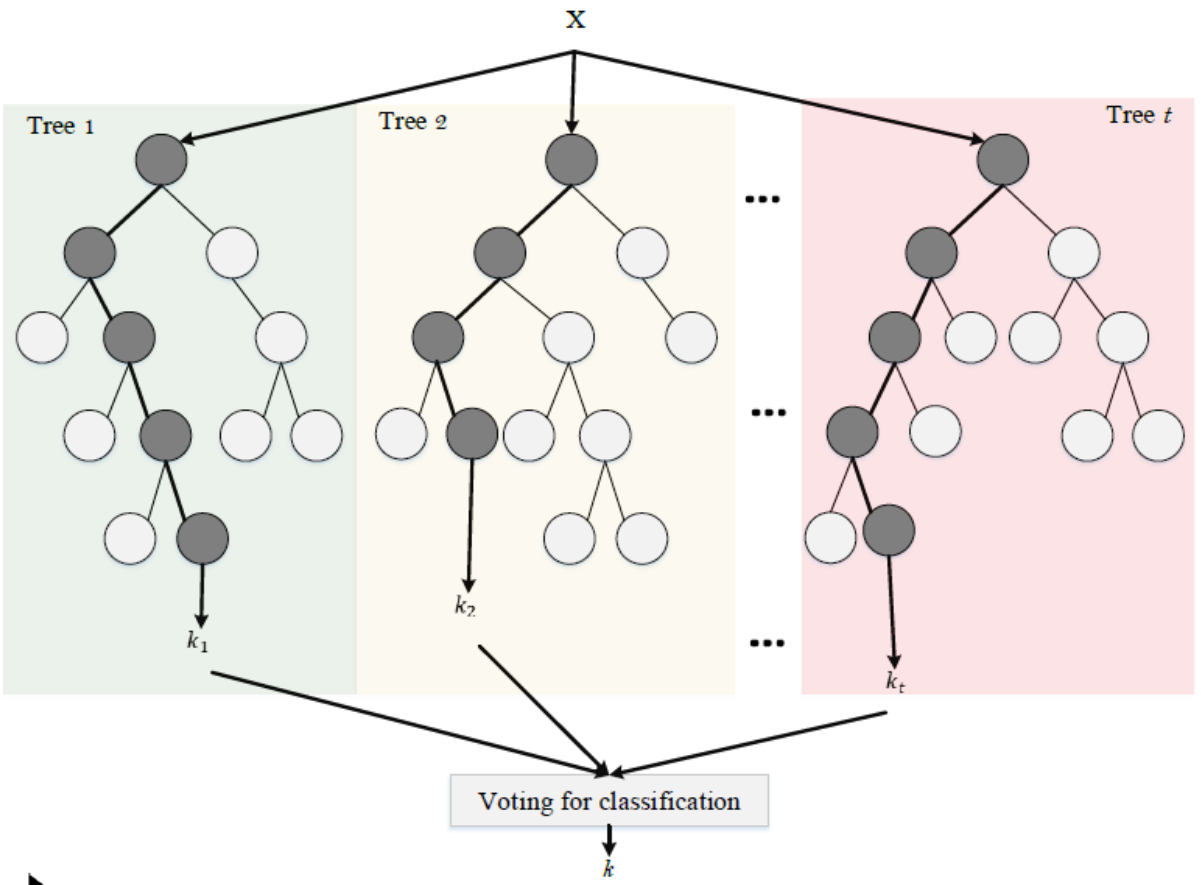


Figure 14. Random forest.

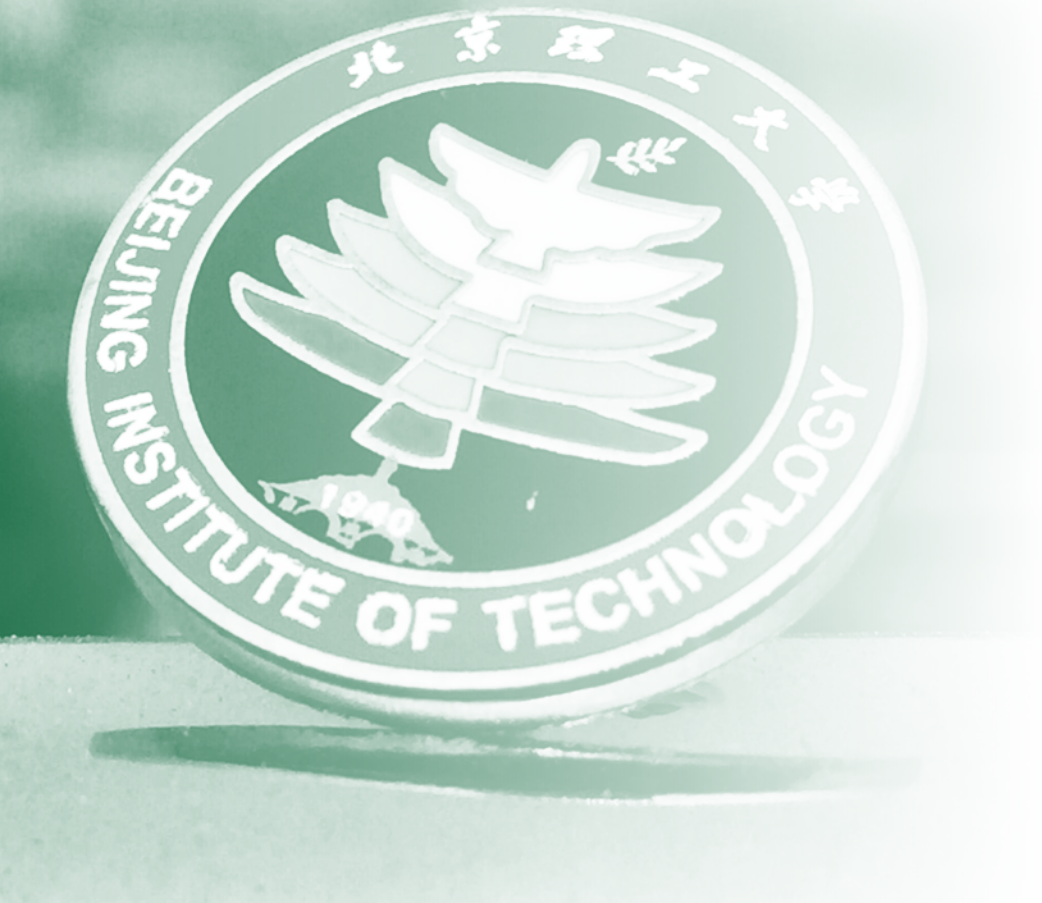
总结

- 基于传统机器学习的方法模型结构较为简单。
- 它学习的是数据预定义的特征表示，因此人工从原始文本中提取特征是这个问题的难点。
- 对于小数据集，在计算复杂度的限制下，传统机器学习学习模型通常比深度学习模型表现出更好的性能。

Reference
[1] Li Q , Peng H , Li J , et al. A Survey on Text Classification: From Shallow to Deep Learning[J]. 2020.
[2] Kowsari, Meimandi J , Heidarysafa, et al. Text Classification Algorithms: A Survey[J]. Information, 2019, 10(4).
[3] Aggarwal, Charu C . Mining Text Data[M]. Springer, 2015.

3 基于深度学习的文本分类

汇报人：主文浩、潘恋军、马翔雨



3-1

基于**普通卷积神经网络**的深度学习文本分类模型

- **TextCNN原理讲解**
- **TextCNN实验介绍**

TextCNN

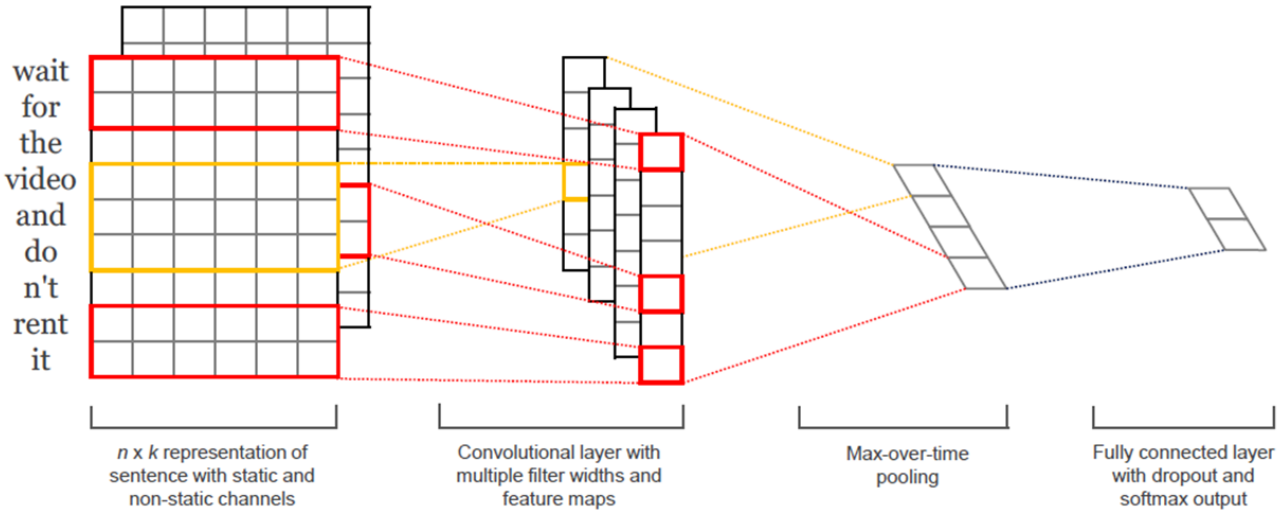


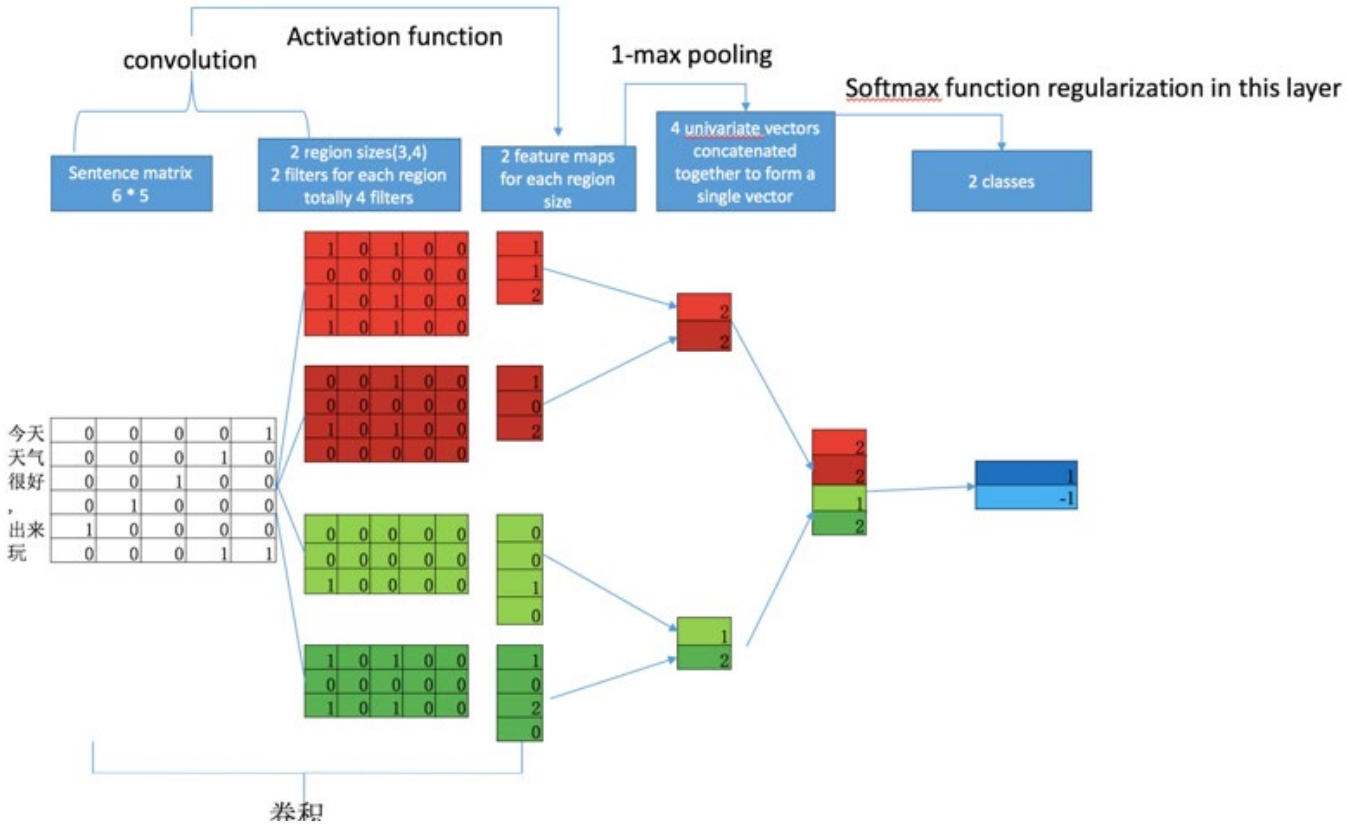
Figure 1: Model architecture with two channels for an example sentence.

与用CNN处理图像的方式类似。

TextCNN 其实只有一层卷积,一层max-pooling, 最后将输出外接softmax 来n分类。

TextCNN通过一维卷积来获取句子中的N-gram特征表示；
预训练词向量作为embedding layer。

TextCNN



TextCNN的成功, 不是网络结构的成功, 而是通过引入已经训练好的词向量来在多个数据集上达到了超越基准（benchmark）的表现，进一步证明了构造更好的映射（embedding）是提升NLP各项任务的关键能力。

- 分词构建词向量（word embedding）
- 卷积（convolution）
- 最大池化(max-pooling)
- softmax进行K分类

结束第一次训练

TextCNN

- Word Embedding

文本分类的重点

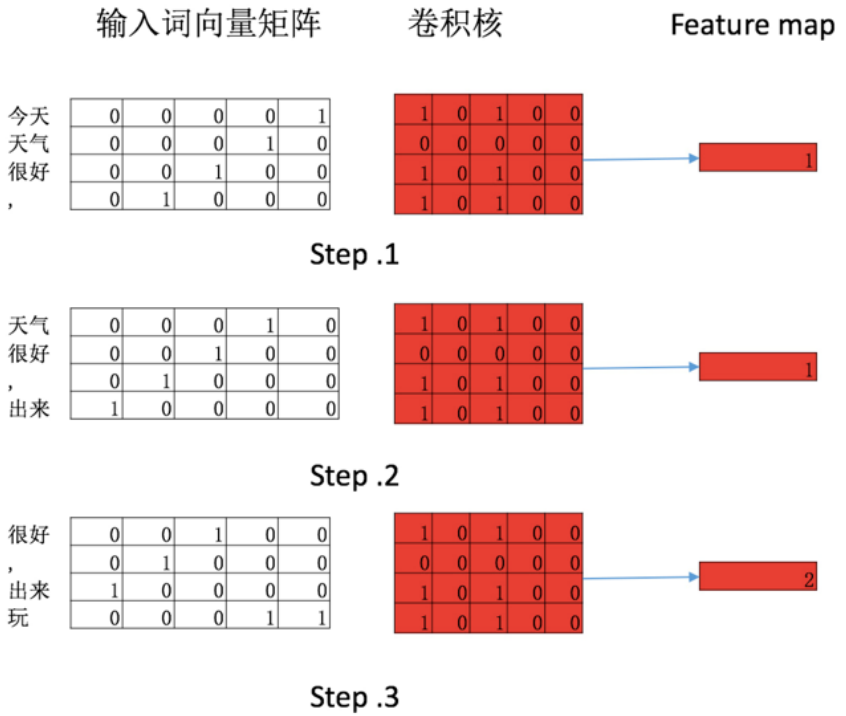
今天	0	0	0	0	1
天气	0	0	0	1	0
很好	0	0	1	0	0
,	0	1	0	0	0
出来	1	0	0	0	0
玩	0	0	0	1	1

TextCNN

Convolution 卷积

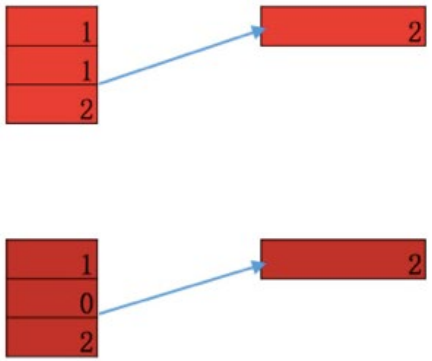
类比于图像处理中的卷积

因为没有padding，所以卷积结束后只有3×1的向量。

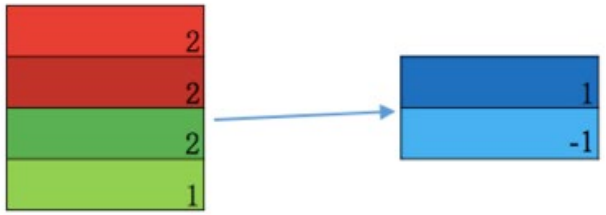


TextCNN

池化+softmax



Max-pooling，选取特征值最大的值



Softmax层的作用 是将**输入的预测向量转化为概率值**，也就是每个元素介于0和1之间，其和为1。

这里输出label为1和label为-1的类别的概率。

TextCNN

实验环境

版本： python3.8
平台语言： pytorch
IDE: pycharm/Anocanda
系统： Windows10
内存： 16G

数据集：THUCNews
准确率： 91.16%

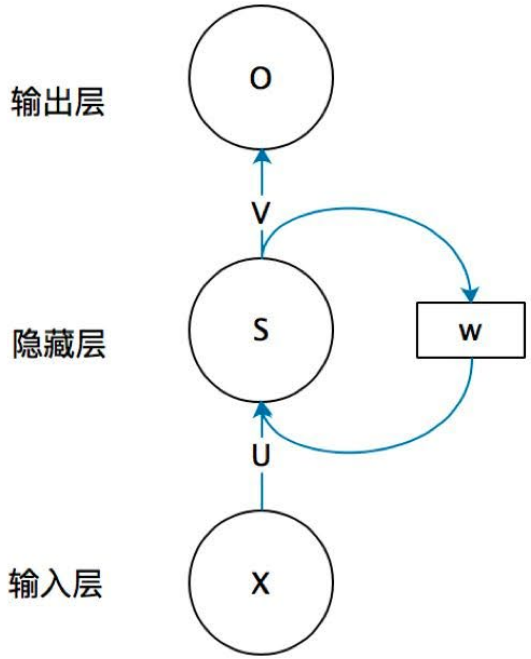
```
Iter:   3800,  Train Loss:  0.29,  Train Acc: 88.28%,  Val Loss:  0.32,  Val Acc: 90.22%,  Time: 0:01:10 *
Iter:   3900,  Train Loss:  0.32,  Train Acc: 88.28%,  Val Loss:  0.33,  Val Acc: 89.97%,  Time: 0:01:12
Iter:   4000,  Train Loss:  0.24,  Train Acc: 92.97%,  Val Loss:  0.33,  Val Acc: 90.23%,  Time: 0:01:14
Iter:   4100,  Train Loss:  0.26,  Train Acc: 89.06%,  Val Loss:  0.33,  Val Acc: 90.34%,  Time: 0:01:16
Iter:   4200,  Train Loss:  0.32,  Train Acc: 88.28%,  Val Loss:  0.33,  Val Acc: 89.95%,  Time: 0:01:18
Epoch [4/20]
Iter:   4300,  Train Loss:  0.23,  Train Acc: 92.97%,  Val Loss:  0.32,  Val Acc: 89.77%,  Time: 0:01:20
Iter:   4400,  Train Loss:  0.22,  Train Acc: 96.09%,  Val Loss:  0.32,  Val Acc: 90.08%,  Time: 0:01:21 *
Iter:   4500,  Train Loss:  0.24,  Train Acc: 92.19%,  Val Loss:  0.33,  Val Acc: 90.16%,  Time: 0:01:23
Iter:   4600,  Train Loss:  0.22,  Train Acc: 93.75%,  Val Loss:  0.32,  Val Acc: 90.09%,  Time: 0:01:25
Iter:   4700,  Train Loss:  0.36,  Train Acc: 90.62%,  Val Loss:  0.32,  Val Acc: 90.26%,  Time: 0:01:27 *
Iter:   4800,  Train Loss:  0.13,  Train Acc: 96.09%,  Val Loss:  0.32,  Val Acc: 90.17%,  Time: 0:01:29 *
Iter:   4900,  Train Loss:  0.23,  Train Acc: 91.41%,  Val Loss:  0.32,  Val Acc: 90.28%,  Time: 0:01:31
Iter:   5000,  Train Loss:  0.25,  Train Acc: 93.75%,  Val Loss:  0.32,  Val Acc: 90.29%,  Time: 0:01:32
Iter:   5100,  Train Loss:  0.26,  Train Acc: 92.97%,  Val Loss:  0.32,  Val Acc: 90.27%,  Time: 0:01:34
Iter:   5200,  Train Loss:  0.32,  Train Acc: 92.19%,  Val Loss:  0.32,  Val Acc: 90.54%,  Time: 0:01:36
Iter:   5300,  Train Loss:  0.19,  Train Acc: 91.41%,  Val Loss:  0.32,  Val Acc: 90.44%,  Time: 0:01:38
Iter:   5400,  Train Loss:  0.38,  Train Acc: 90.62%,  Val Loss:  0.33,  Val Acc: 90.32%,  Time: 0:01:40
Iter:   5500,  Train Loss:  0.21,  Train Acc: 92.97%,  Val Loss:  0.32,  Val Acc: 90.57%,  Time: 0:01:42
Iter:   5600,  Train Loss:  0.14,  Train Acc: 95.31%,  Val Loss:  0.33,  Val Acc: 90.22%,  Time: 0:01:43
Epoch [5/20]
Iter:   5700,  Train Loss:  0.25,  Train Acc: 89.84%,  Val Loss:  0.32,  Val Acc: 90.34%,  Time: 0:01:45
Iter:   5800,  Train Loss:  0.14,  Train Acc: 93.75%,  Val Loss:  0.33,  Val Acc: 90.18%,  Time: 0:01:47
No optimization for a long time, auto-stopping...
Test Loss:   0.3,  Test Acc: 91.16%
Precision, Recall and F1-Score...
               precision    recall  f1-score   support

      finance     0.9125    0.8970    0.9047      1000
       realty     0.9349    0.9330    0.9339      1000
       stocks     0.8690    0.8490    0.8589      1000
    education     0.9513    0.9580    0.9547      1000
      science     0.8816    0.8710    0.8763      1000
      society     0.8835    0.9250    0.9038      1000
     politics     0.8757    0.9020    0.8887      1000
       sports     0.9635    0.9490    0.9562      1000
         game     0.9436    0.9040    0.9234      1000
entertainment     0.9036    0.9280    0.9156      1000

     accuracy                         0.9116     10000
    macro avg     0.9119    0.9116    0.9116     10000
 weighted avg     0.9119    0.9116    0.9116     10000
```

3-2

基于**循环神经网络**的深度学习文本分类模型

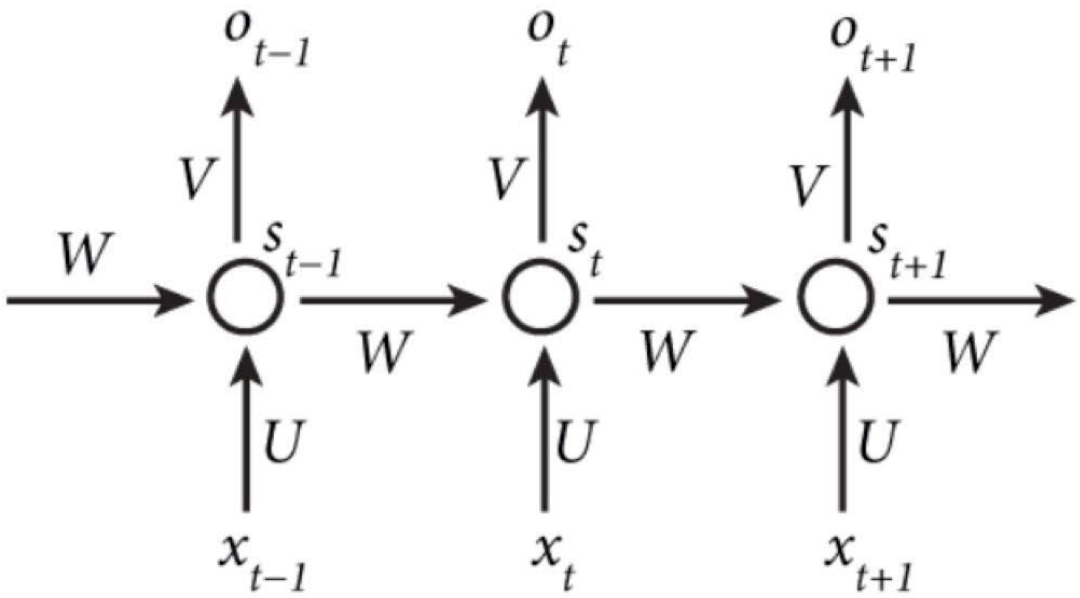
基于上下文机制的深度学习文本分类模型

1. 普通循环神经网络



循环层

按照时间线展开



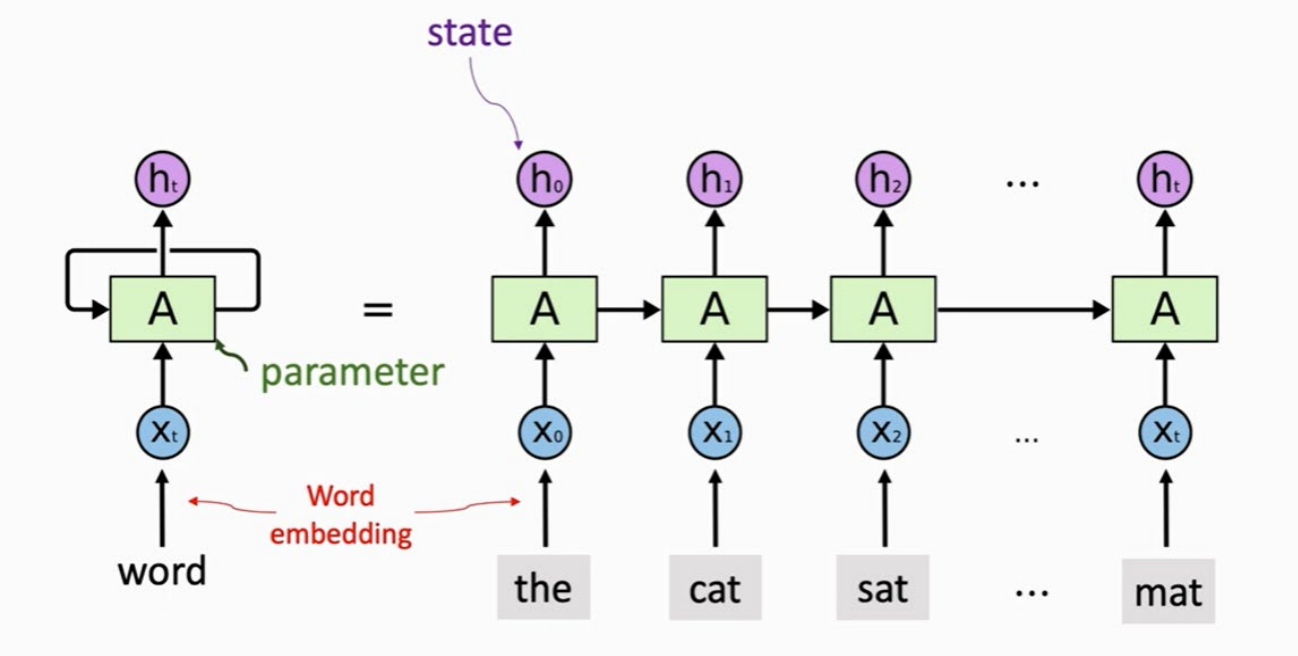
$$O_t = g(V \cdot S_t)$$

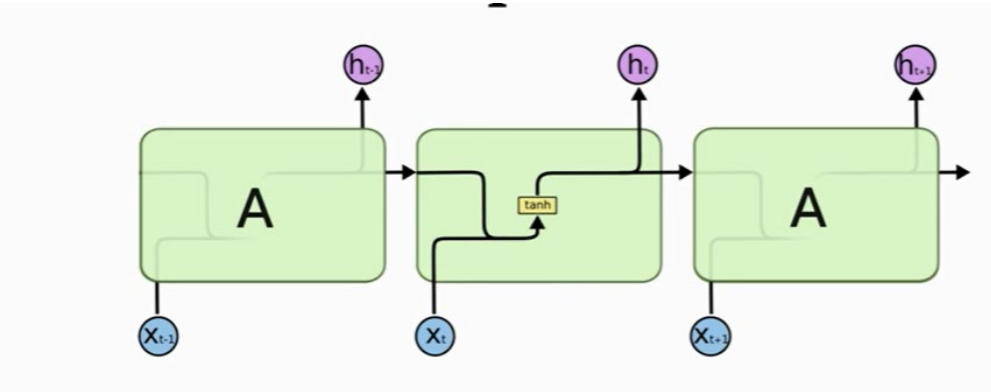
$$S_t = f(U \cdot X_t + W \cdot S_{t-1})$$

St的值不仅仅取决于Xt，还取决于St-1

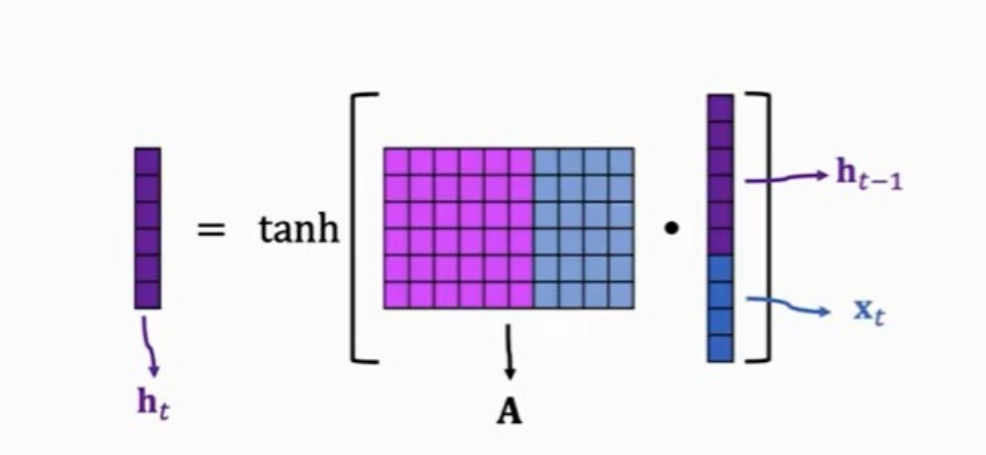
基于上下文机制的深度学习文本分类模型

1.普通循环神经网络



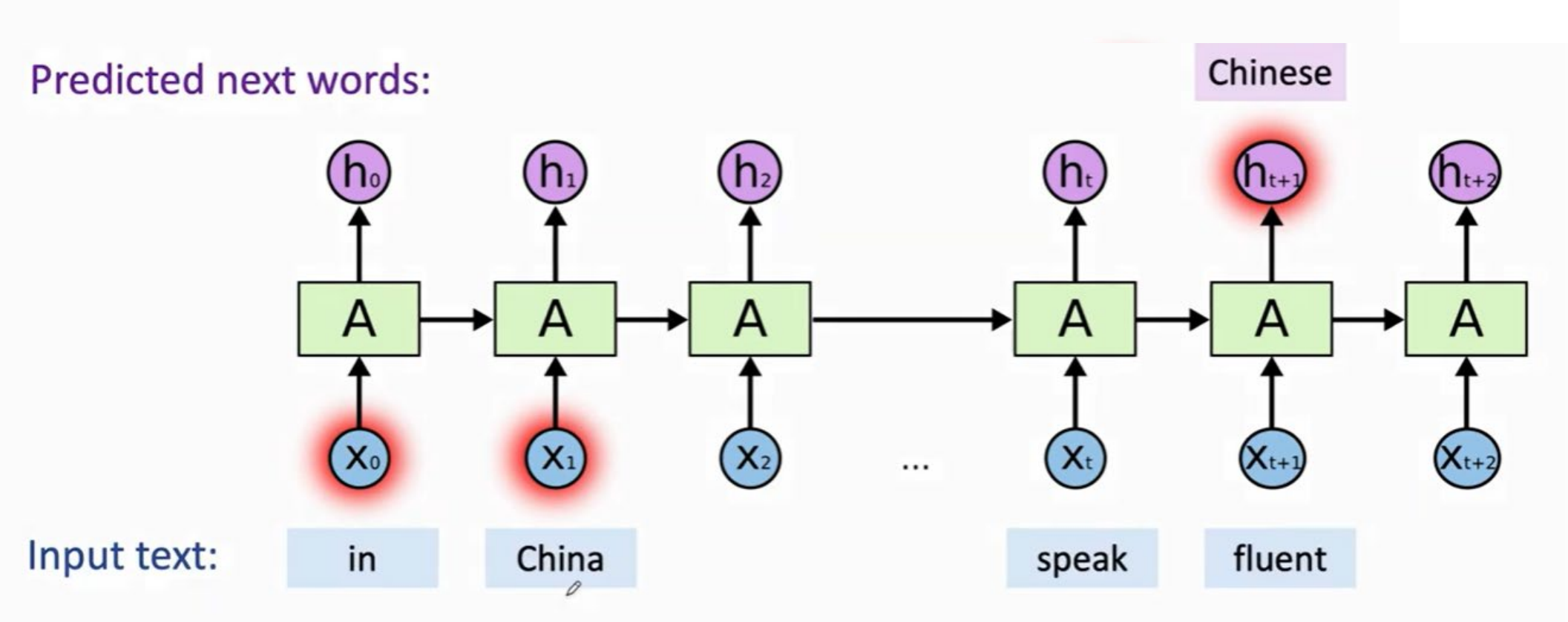


A：RNN的模型参数
Tanh：双曲正切函数



基于上下文机制的深度学习文本分类模型

1.普通循环神经网络-缺陷



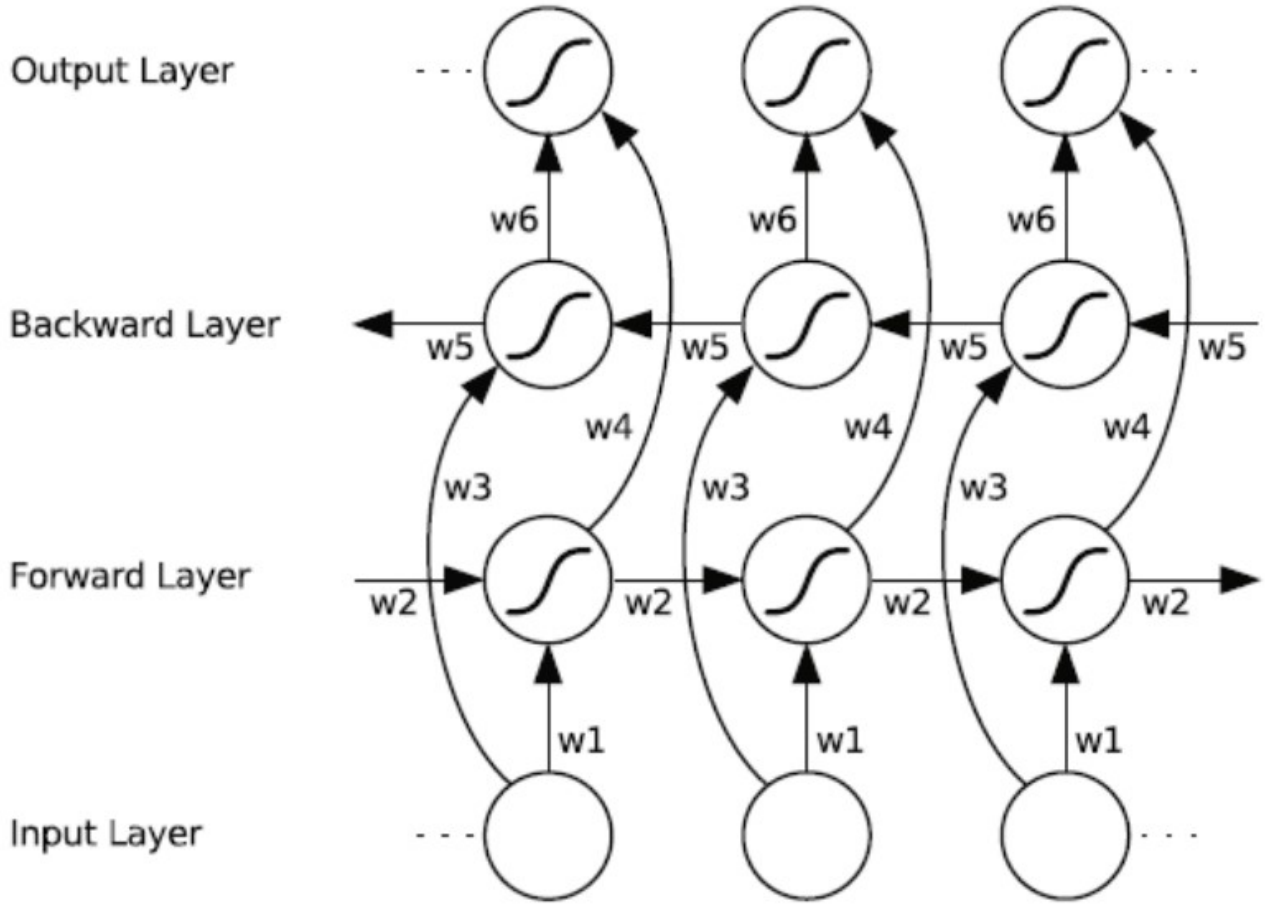
基于上下文机制的深度学习文本分类模型

```
Iter:   5500,  Train Loss:   0.2,  Train Acc: 92.19%,  Val Loss:  0.31,  Val Acc: 90.54%,  Time: 0:00:55
Iter:   5600,  Train Loss:  0.15,  Train Acc: 95.31%,  Val Loss:  0.31,  Val Acc: 90.69%,  Time: 0:00:56
Epoch [5/10]
Iter:   5700,  Train Loss:  0.24,  Train Acc: 91.41%,  Val Loss:  0.32,  Val Acc: 90.55%,  Time: 0:00:57
No optimization for a long time, auto-stopping...
Test Loss:   0.3,  Test Acc: 90.67%
Precision, Recall and F1-Score...
               precision    recall  f1-score   support

      finance     0.9141    0.9040    0.9090      1000
       realty     0.9217    0.9300    0.9258      1000
       stocks     0.9007    0.8070    0.8513      1000
    education     0.9151    0.9490    0.9318      1000
      science     0.8024    0.8850    0.8417      1000
      society     0.8741    0.9300    0.9012      1000
     politics     0.9035    0.8430    0.8722      1000
       sports     0.9701    0.9740    0.9721      1000
         game     0.9527    0.9070    0.9293      1000
entertainment     0.9260    0.9380    0.9319      1000

     accuracy                         0.9067     10000
    macro avg     0.9080    0.9067    0.9066     10000
 weighted avg     0.9080    0.9067    0.9066     10000
```

数据集：THUCNews（根据新浪新闻RSS订阅频道2005~2011年间的历史数据筛选过滤生成）

基于上下文机制的深度学习文本分类模型

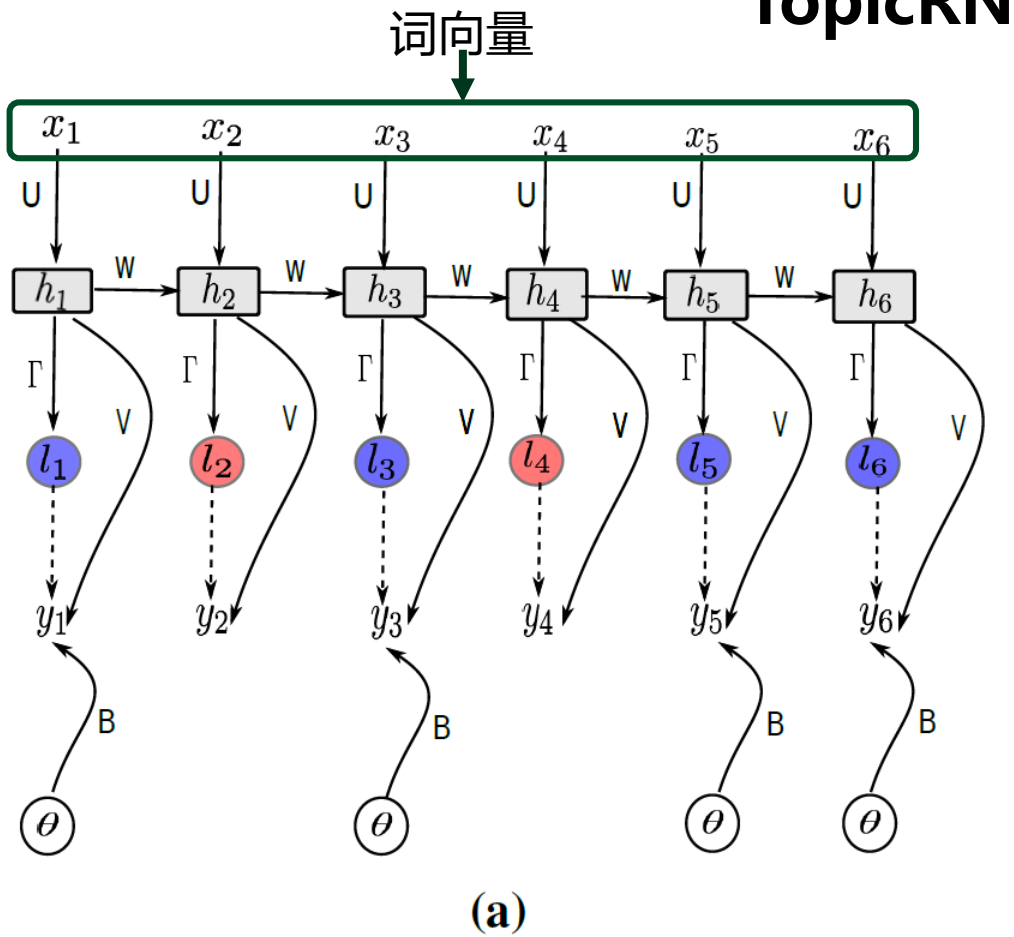
2. 双向循环神经网络



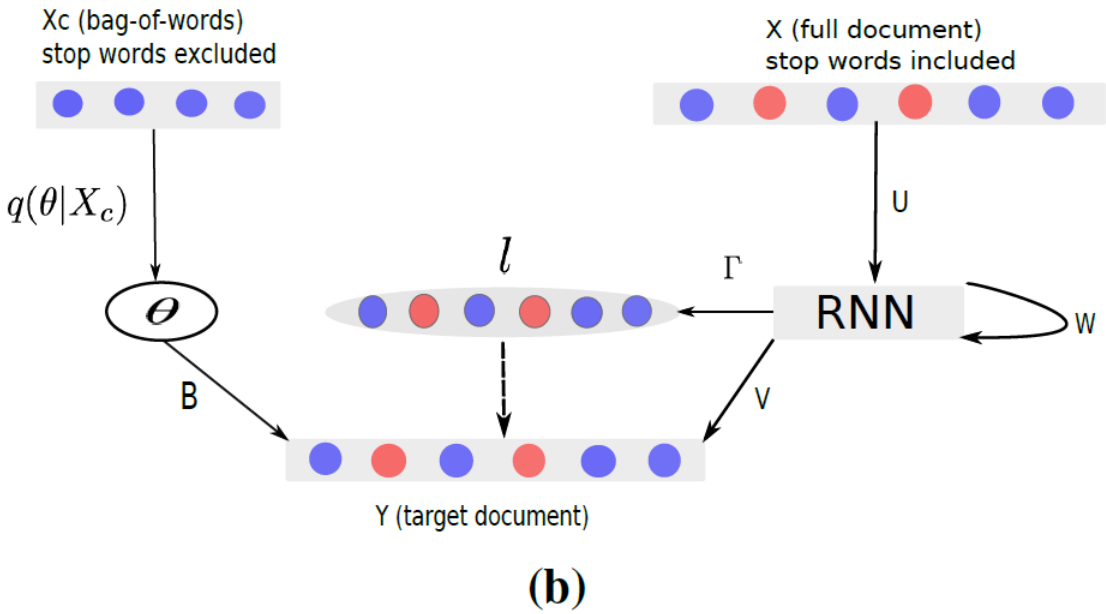
基于上下文机制的深度学习文本分类模型

3. 一个例子 —— TopicRNN[1]

TopicRNN 模型中同时兼顾了句法信息和语义信息



(a)



(b)

[1] Dieng A B, Wang C, Gao J, et al. Topicrnn: A recurrent neural network with long-range semantic dependency[J]. arXiv preprint arXiv:1611.01702, 2016.

基于上下文机制的深度学习文本分类模型

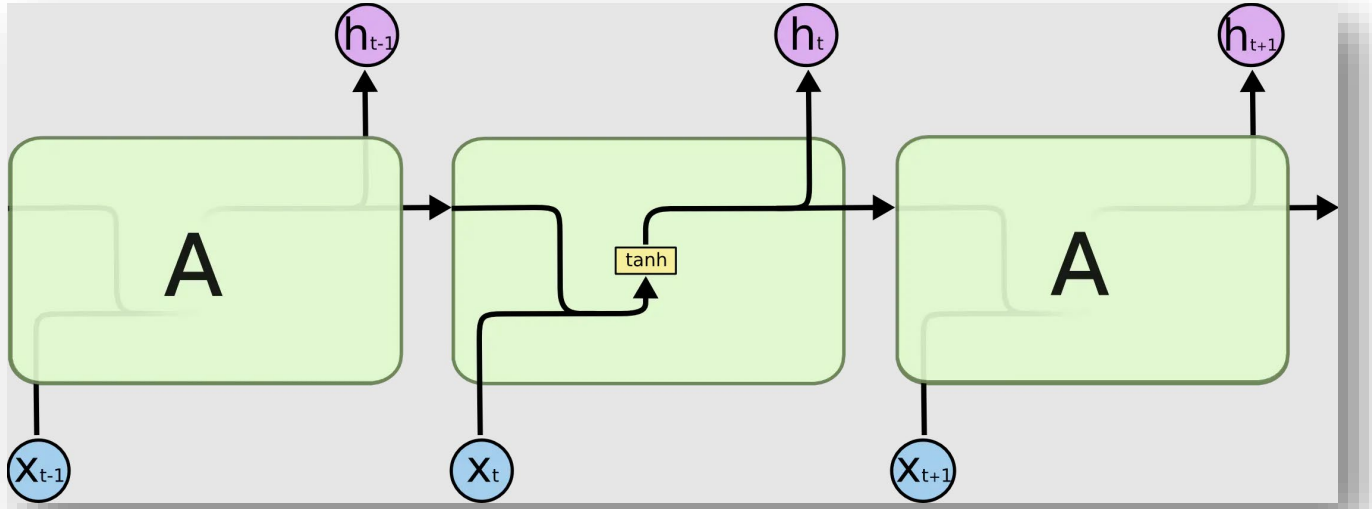
Model	Reported Error rate
BoW (bnc) (Maas et al., 2011)	12.20%
BoW ($b\Delta$ tć) (Maas et al., 2011)	11.77%
LDA (Maas et al., 2011)	32.58%
Full + BoW (Maas et al., 2011)	11.67%
Full + Unlabelled + BoW (Maas et al., 2011)	11.11%
WRRBM (Dahl et al., 2012)	12.58%
WRRBM + BoW (bnc) (Dahl et al., 2012)	10.77%
MNB-uni (Wang & Manning, 2012)	16.45%
MNB-bi (Wang & Manning, 2012)	13.41%
SVM-uni (Wang & Manning, 2012)	13.05%
SVM-bi (Wang & Manning, 2012)	10.84%
NBSVM-uni (Wang & Manning, 2012)	11.71%
seq2-bown-CNN (Johnson & Zhang, 2014)	14.70%
NBSVM-bi (Wang & Manning, 2012)	8.78%
Paragraph Vector (Le & Mikolov, 2014)	7.42%
SA-LSTM with joint training (Dai & Le, 2015)	14.70%
LSTM with tuning and dropout (Dai & Le, 2015)	13.50%
LSTM initialized with word2vec embeddings (Dai & Le, 2015)	10.00%
SA-LSTM with linear gain (Dai & Le, 2015)	9.17%
LM-TM (Dai & Le, 2015)	7.64%
SA-LSTM (Dai & Le, 2015)	7.24%
Virtual Adversarial (Miyato et al. 2016)	**5.91%**
TopicRNN	**6.28%**

数据集：IMDB

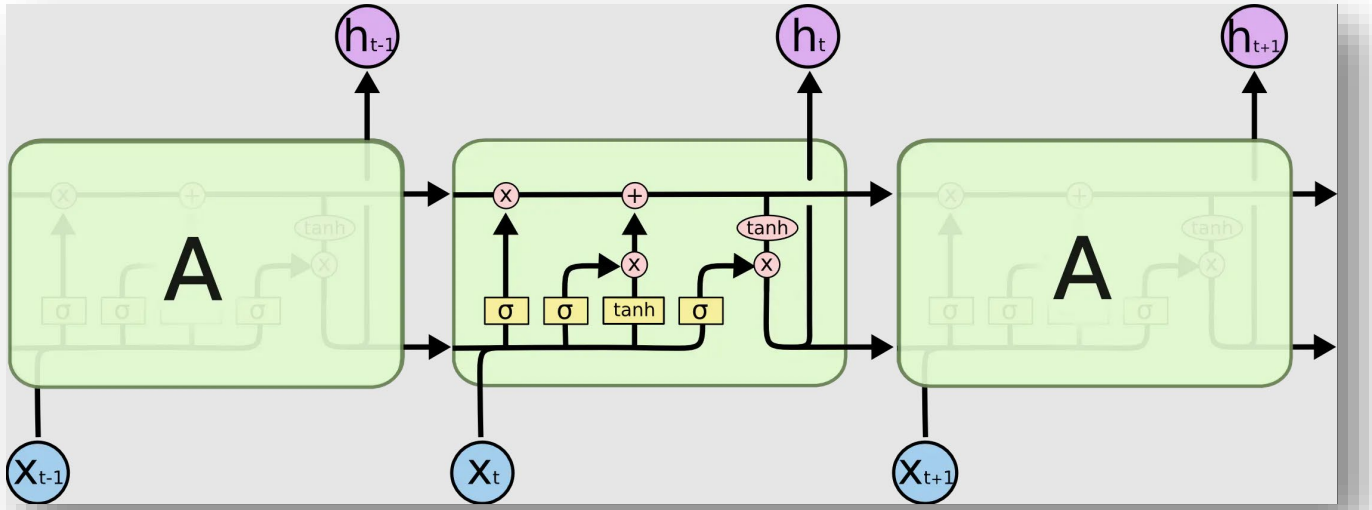
3-3

基于**长短期记忆网络**的深度学习文本分类模型

基于记忆存储机制的深度学习文本分类模型

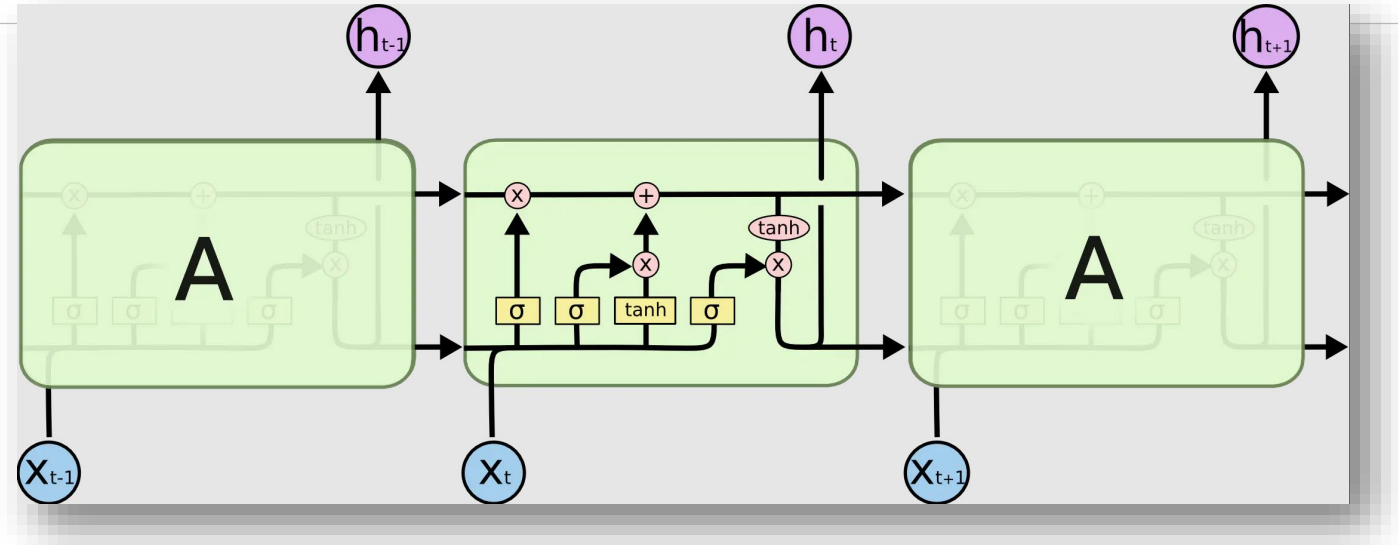


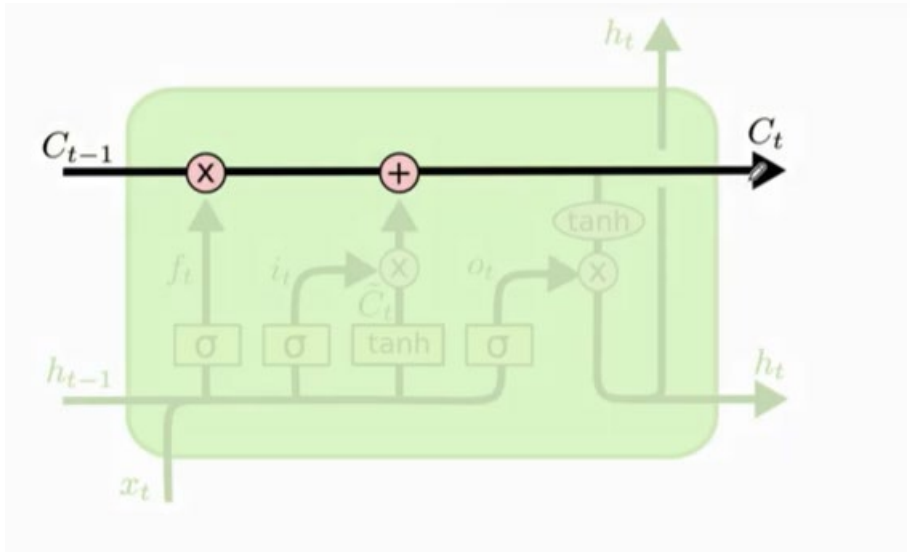
循环神经网络具有短期记忆，在处理较长的序列数据时，很难将信息传递到较远层.



长短期记忆神经网络具有长期短期记忆，在处理较长的序列数据时，比较有优势.

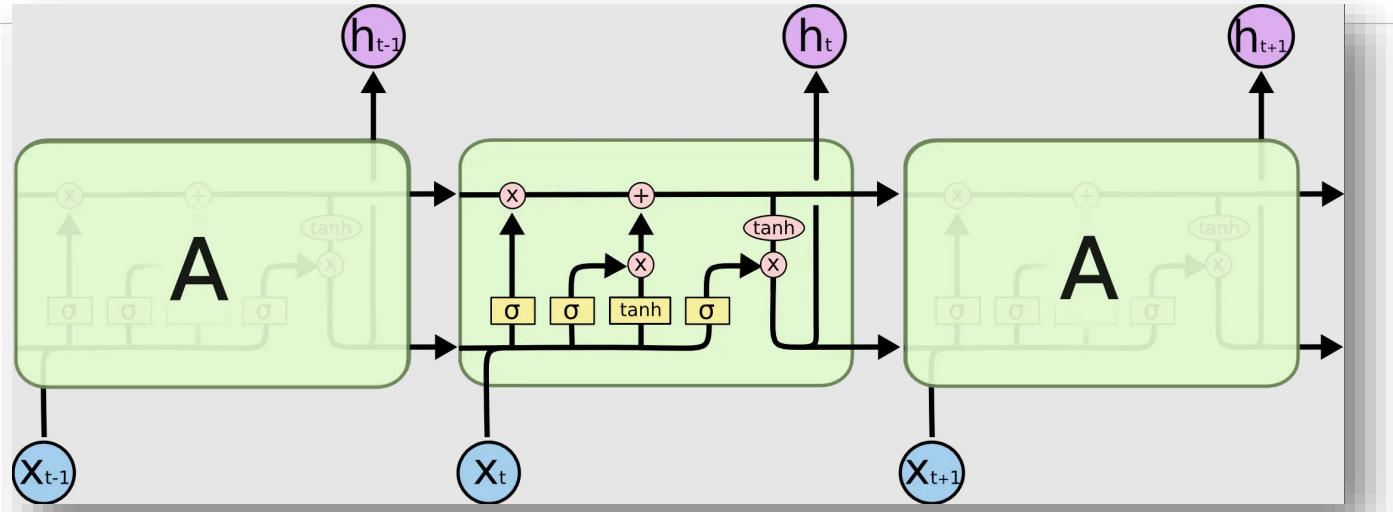
LSTM



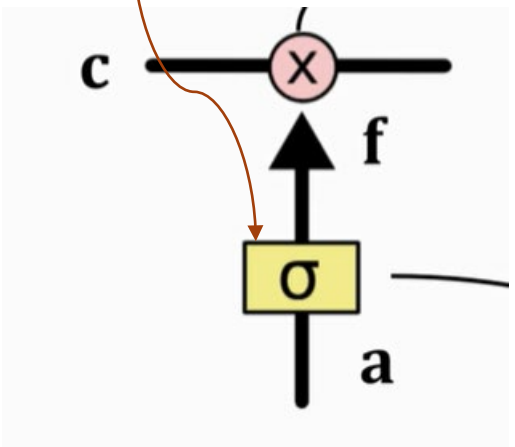


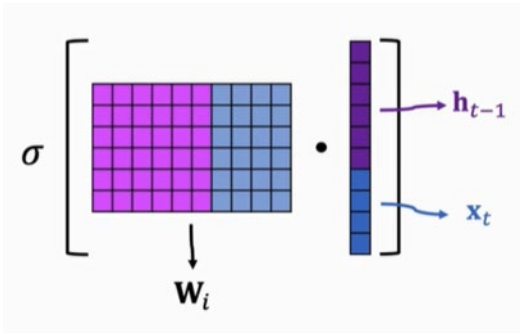
C_t：控制参数（传送带），决定什么样的信息会被保留，什么样的信息会被遗忘。

LSTM



遗忘门

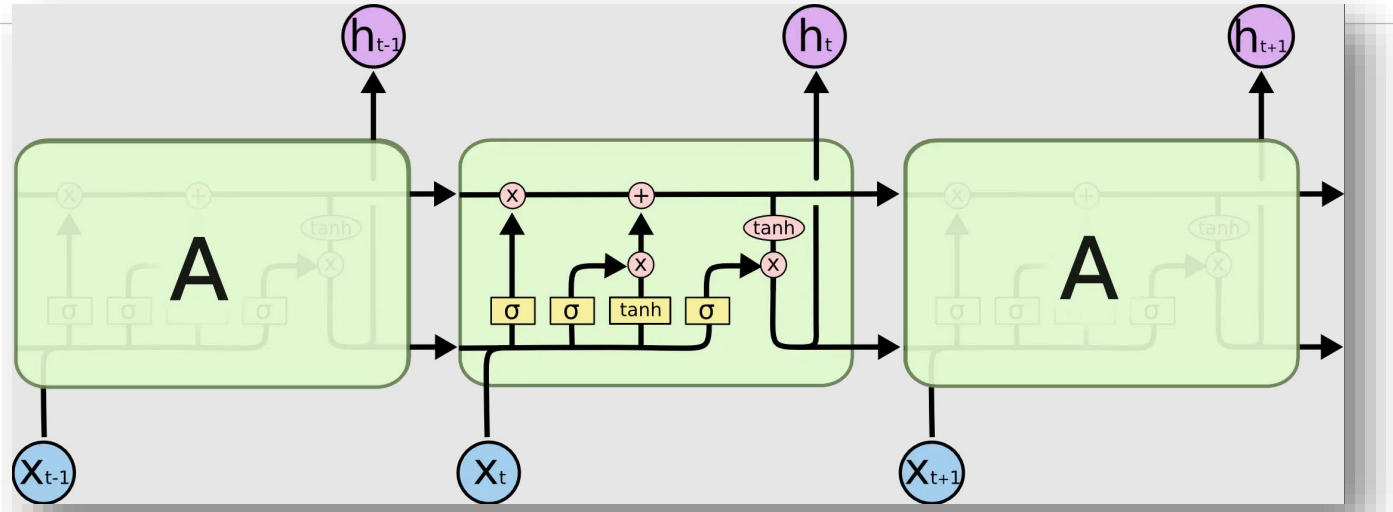




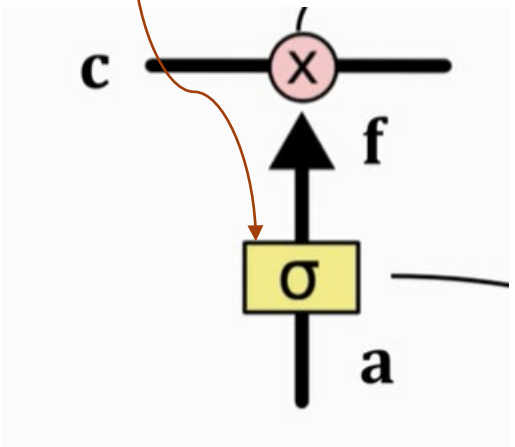
$$\sigma\left(\begin{bmatrix}1\\3\\0\\-2\end{bmatrix}\right) = \begin{bmatrix}0.73\\0.95\\0.5\\0.12\end{bmatrix}$$

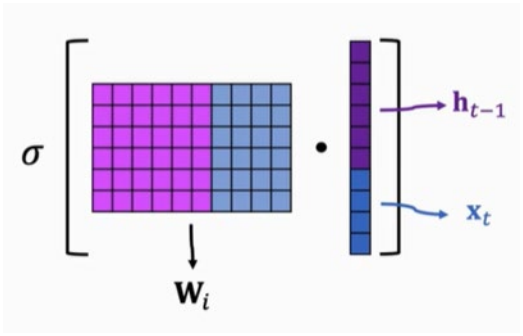
a　　f

LSTM



遗忘门

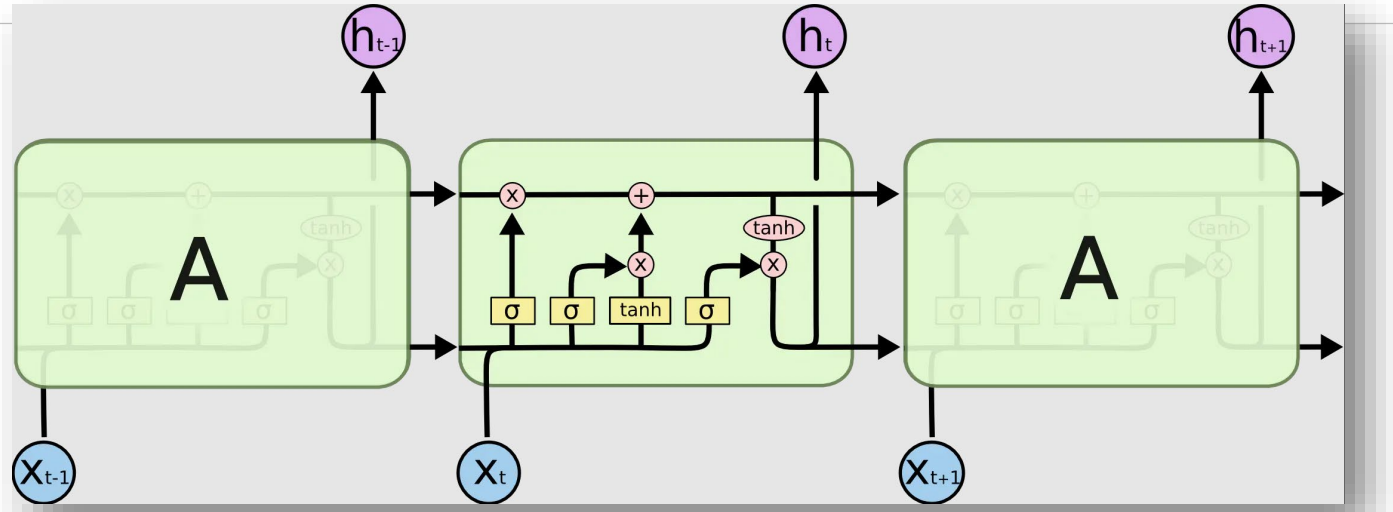




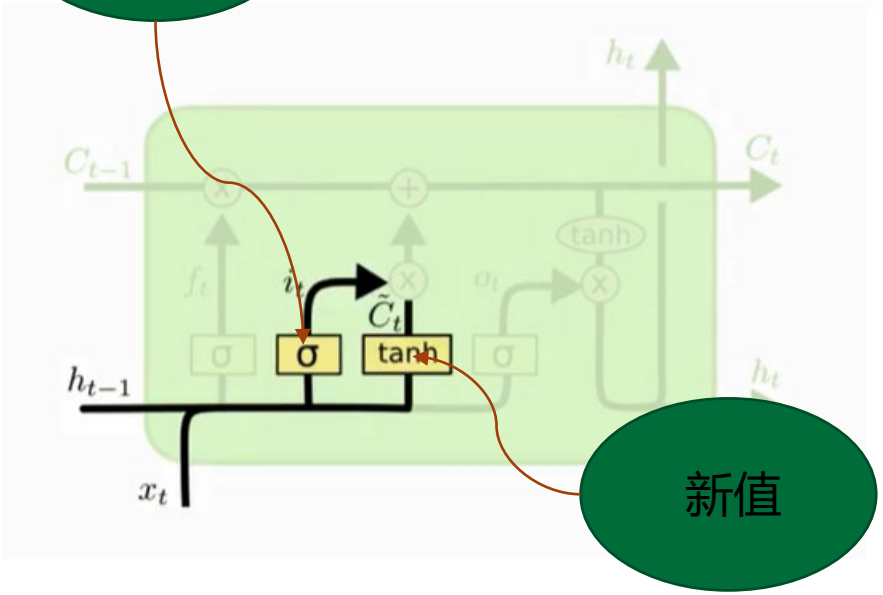
$$\begin{bmatrix} 0.9 \\ 0.2 \\ -0.5 \\ -0.1 \end{bmatrix} \circ \begin{bmatrix} 0.5 \\ 0 \\ 1 \\ 0.8 \end{bmatrix} = \begin{bmatrix} 0.45 \\ 0 \\ -0.5 \\ -0.08 \end{bmatrix}$$

c f output

LSTM



输入门

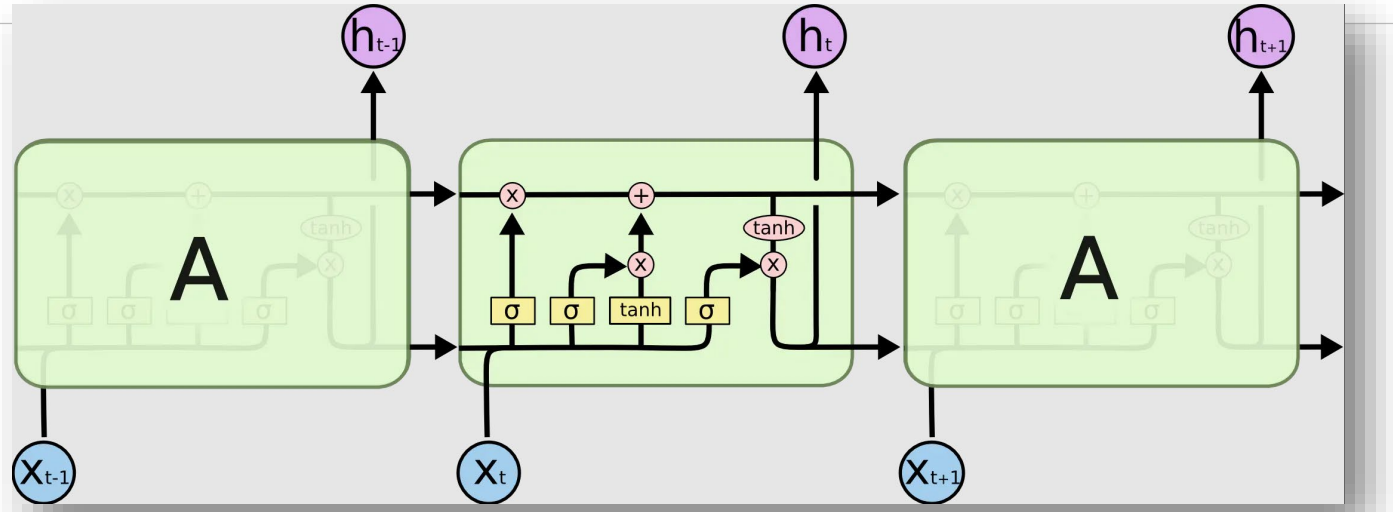


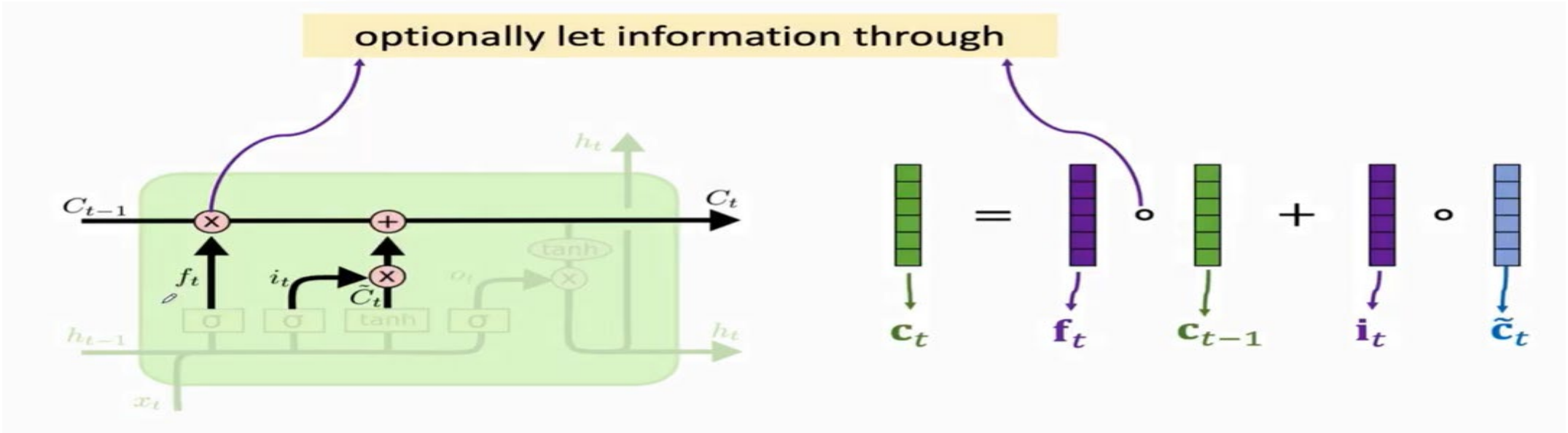
新值

$$i_t = \sigma\left[W_i \cdot \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}\right]$$

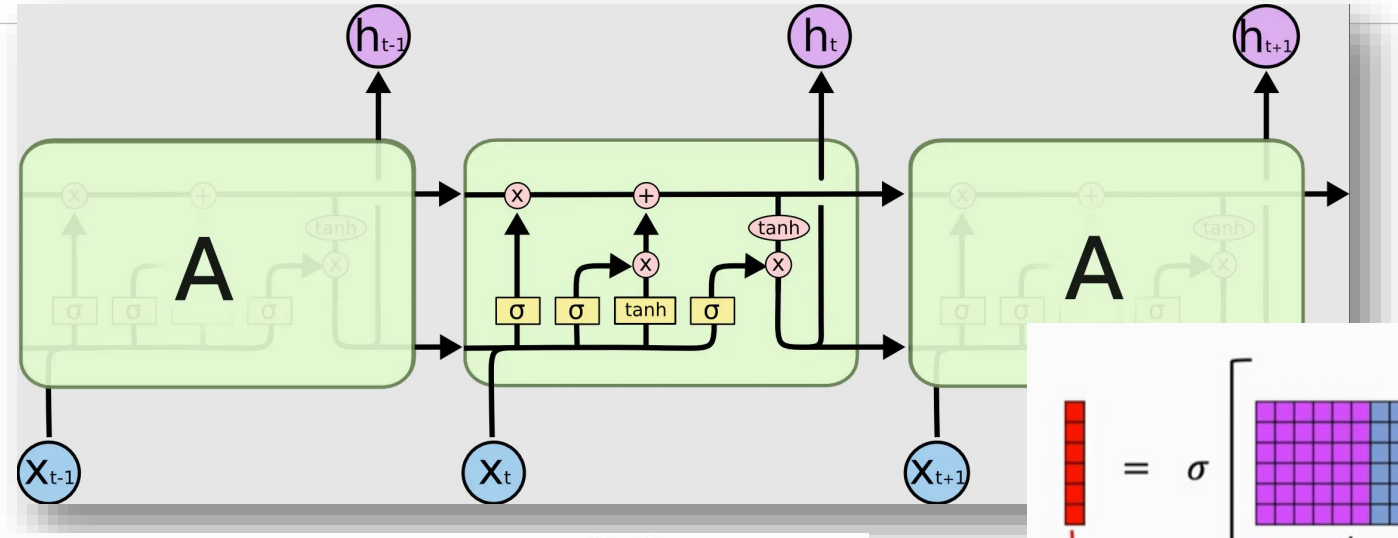
$$\tilde{c}_t = \tanh\left[W_c \cdot \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}\right]$$

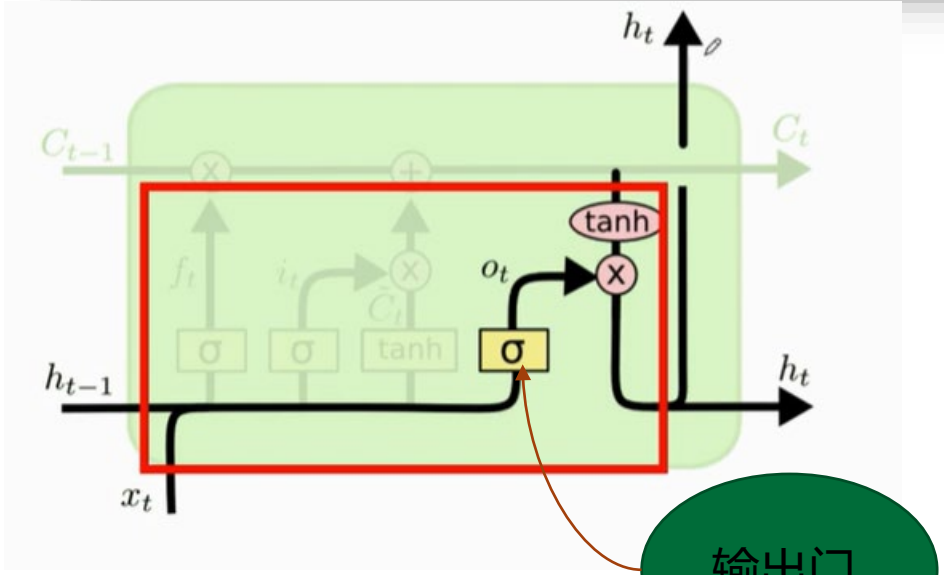
LSTM


h_{t-1}
h_t
h_{t+1}
A
A
tanh
σ
σ
tanh
σ
X_{t-1}
X_t
X_{t+1}

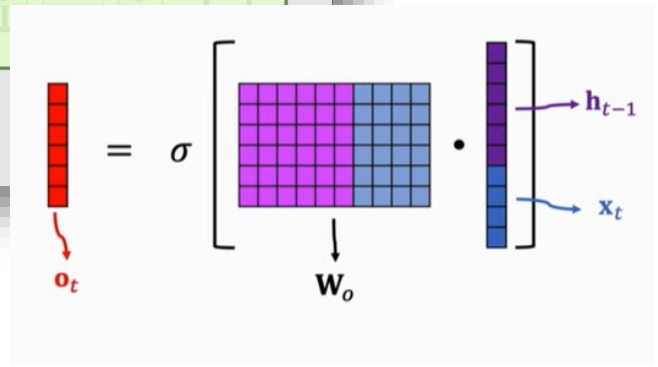

optionally let information through
C_{t-1}
C_t
f_t
i_t
$\tilde{C}_t$
h_{t-1}
h_t
x_t
tanh
$\mathbf{c}_t$ = $\mathbf{f}_t$ ∘ $\mathbf{c}_{t-1}$ + $\mathbf{i}_t$ ∘ $\tilde{\mathbf{c}}_t$

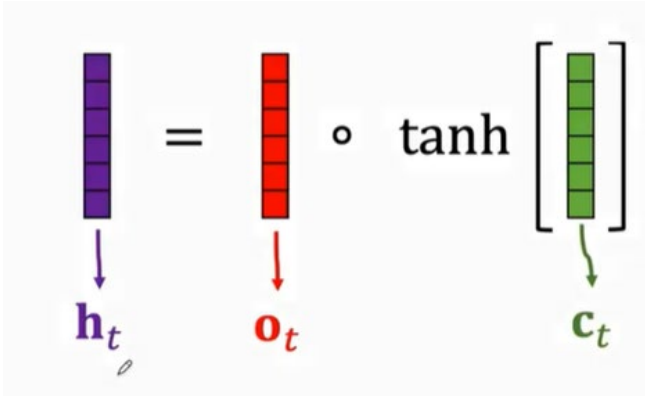
LSTM


h_{t-1}
h_t
h_{t+1}
A
A
tanh
σ
σ
tanh
σ
x_{t-1}
x_t
x_{t+1}


h_t
C_{t-1}
C_t
tanh
o_t
f_t
i_t
$\tilde{C}_t$
σ
σ
tanh
σ
h_{t-1}
h_t
x_t

输出门


o_t
=
σ
W_o
h_{t-1}
x_t

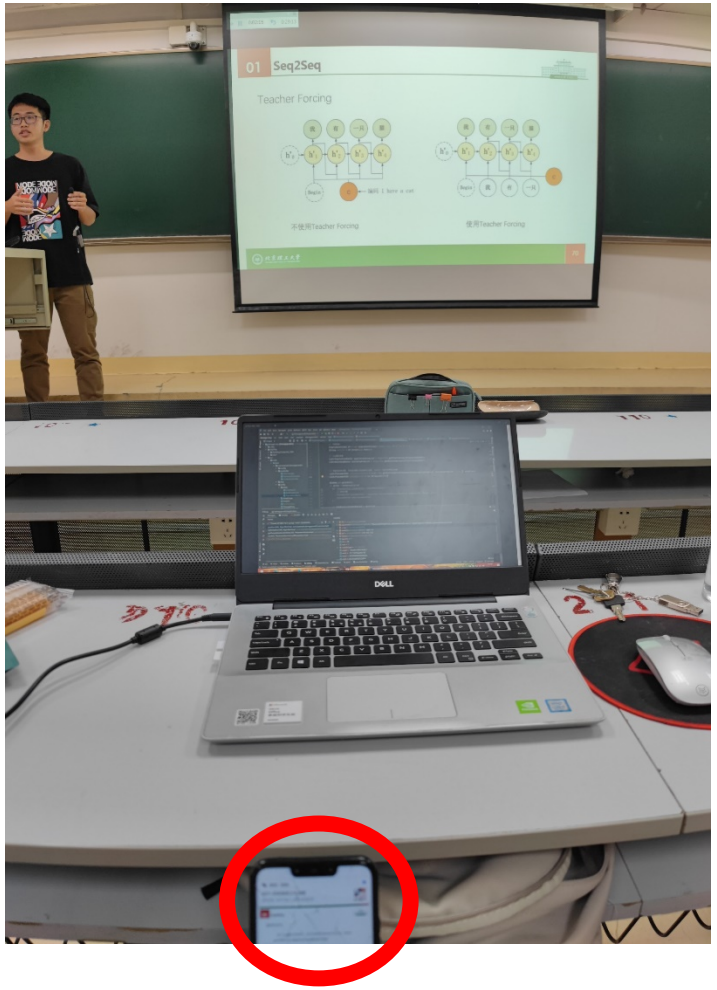

h_t
=
o_t
∘
tanh
c_t

3-4

基于**注意力机制**的深度学习文本分类模型

基于注意力机制的深度学习文本分类模型

1. Attention机制

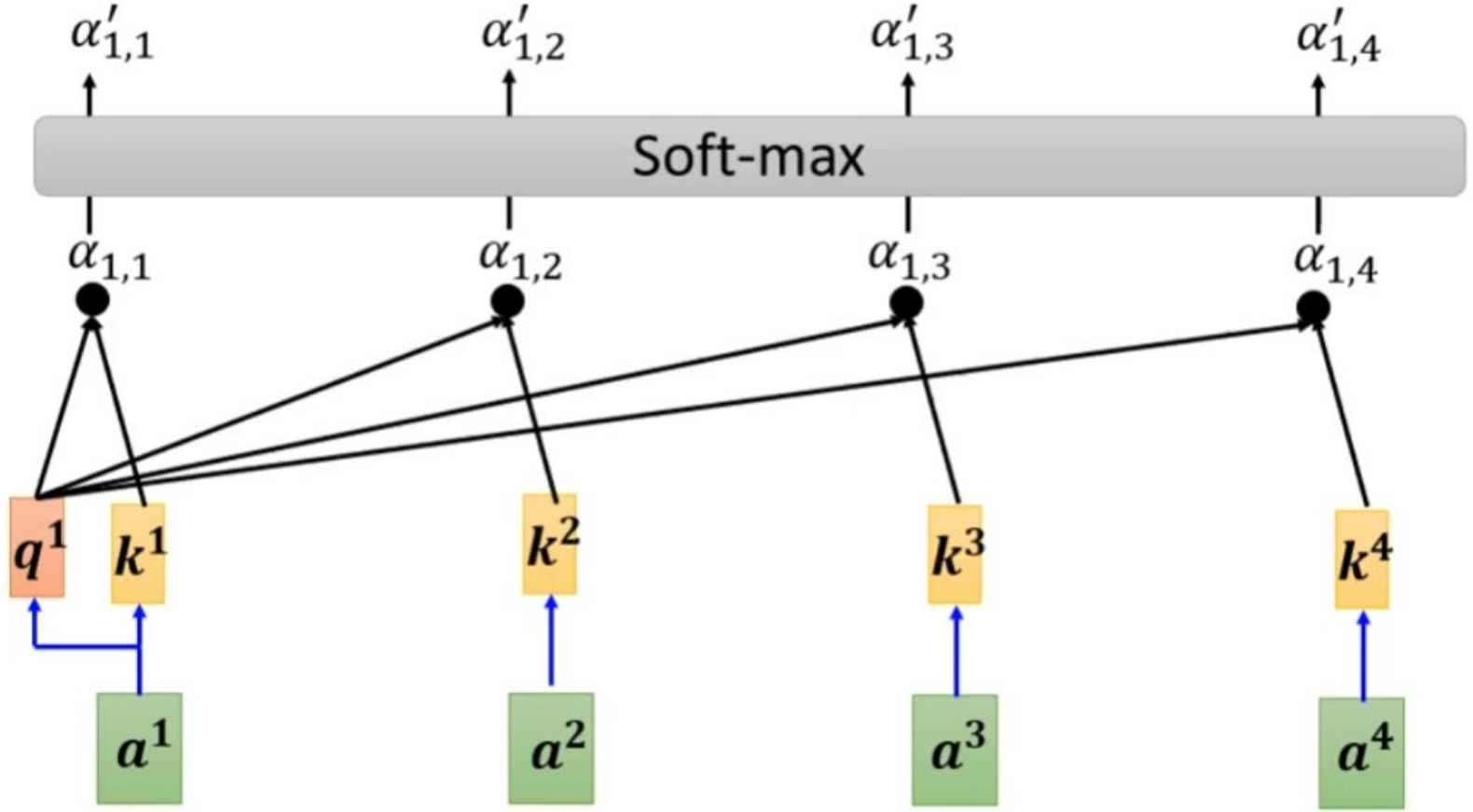


注意力（Attention）机制最先应用在图像处理中，后来逐步被引入到自然语言处理领域。

通过引入注意力机制来提取具有重要意义的词汇来对句子进行表示，并将这些信息词汇的表征聚合起来形成句子向量。

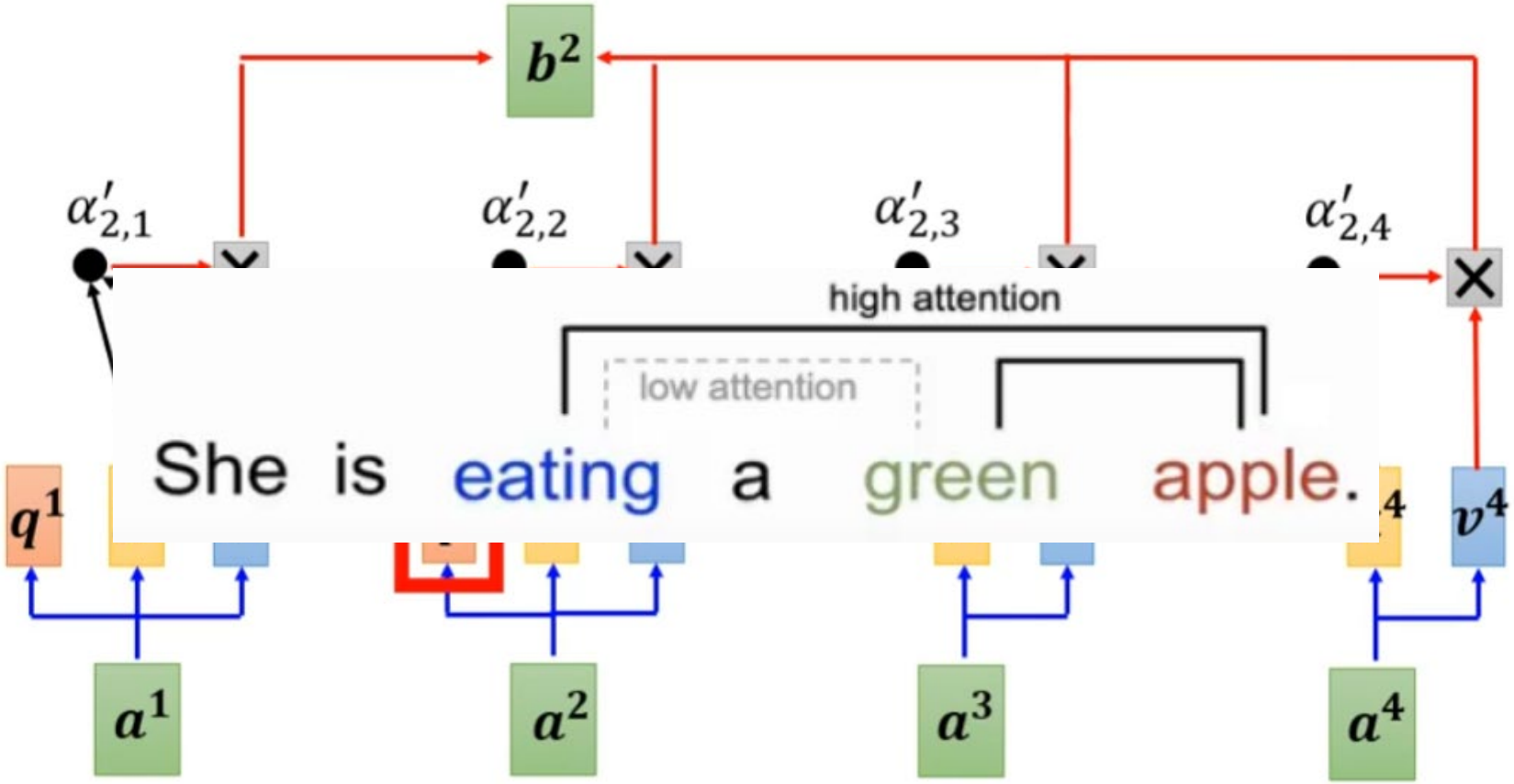
基于注意力机制的深度学习文本分类模型

2. self-attention



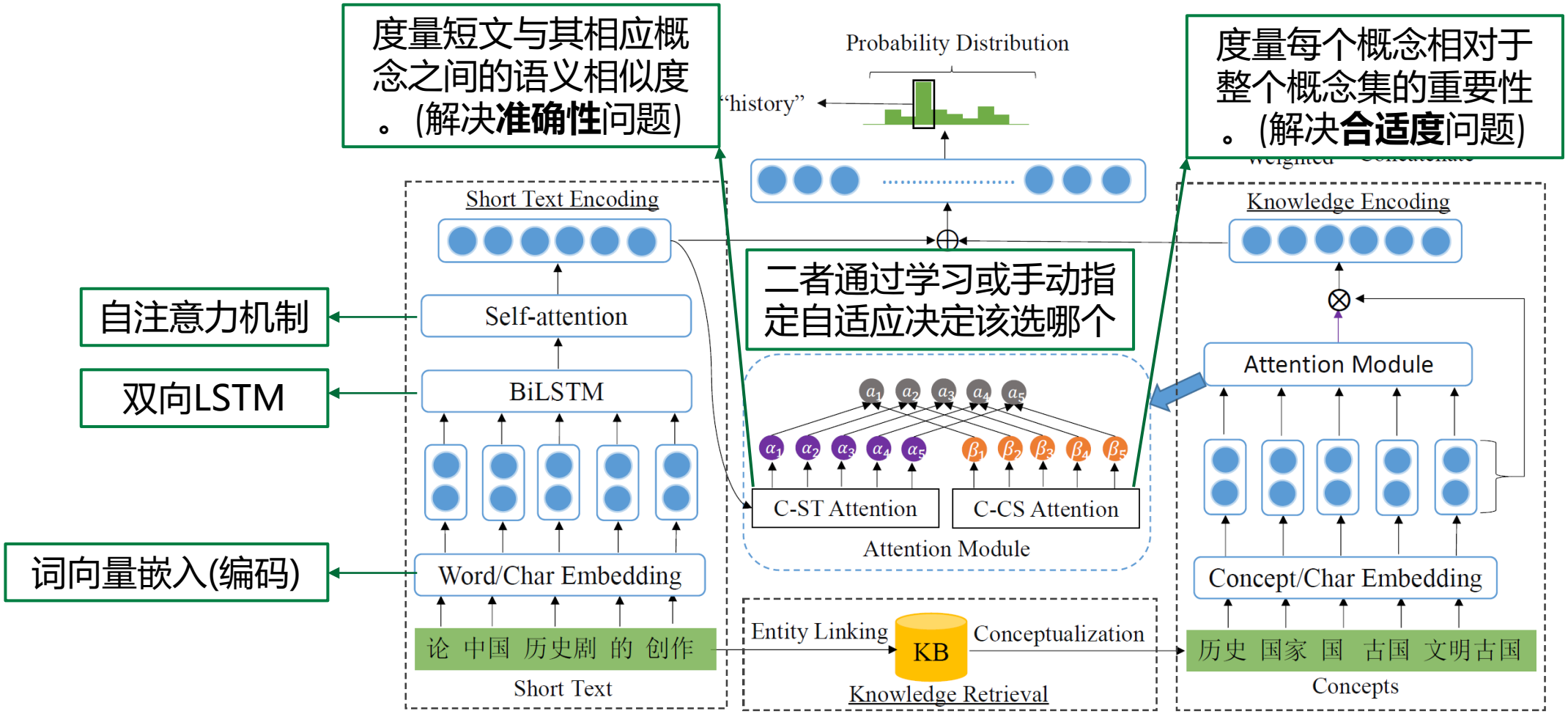
基于注意力机制的深度学习文本分类模型

2. self-attention



基于注意力机制的深度学习文本分类模型

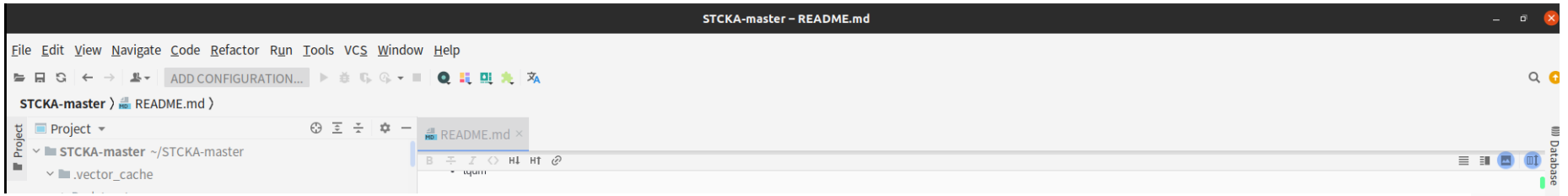
3. STCKA——基于 **知识驱动 注意力** 的深度 短文本 分类方法



Chen J, Hu Y, Liu J, et al. Deep short text classification with knowledge powered attention[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33(01): 6252-6259.

基于注意力机制的深度学习文本分类模型

3. STCKA——DEMO



使用数据集名称:

TagMyNews

测试集准确率:81.47%

Model	Weibo	Topic	Product Review	News Title
CNN	0.3900	0.8243	0.7290	0.7706
RCNN	0.4040	0.8257	0.7280	0.7853
CharCNN	0.4100	0.8500	0.7010	0.7493
BiLSTM-MP	0.4160	0.8186	0.7290	0.7719
BiLSTM-SA	0.4120	0.8200	0.7310	0.7802
KPCNN	0.4240	0.8643	0.7340	0.7878
STCKA	**0.4320**	**0.8814**	**0.7430**	**0.8011**

```
100%|
10/28/2021 20:04:57 - INFO - __main__ -  Epoch:99, Training Loss:0.0123
10/28/2021 20:04:57 - INFO - __main__ -  Epoch:99, Eval Loss:1.3292, Eval Acc:0.7826, Eval P:0.7473, Eval R:0.7393, Eval F1:0.7425
10/28/2021 20:04:57 - INFO - __main__ -  Test Loss:0.6093, Test Acc:0.8147, Test P:0.7858, Test R:0.7862, Test F1:0.7859
(stcka) qdmxy@qdmxy-G7:~/STCKA-master$
```

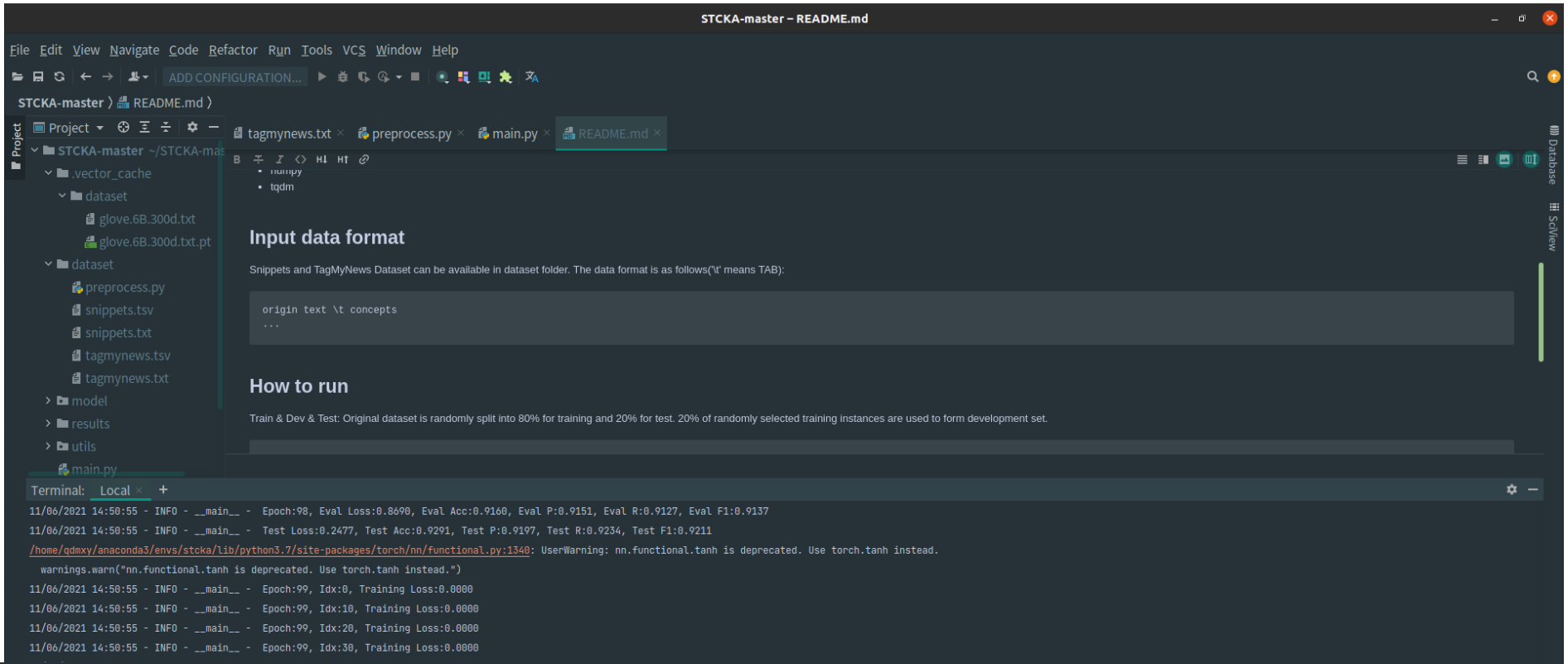
基于注意力机制的深度学习文本分类模型

3. STCKA——DEMO

使用数据集名称:

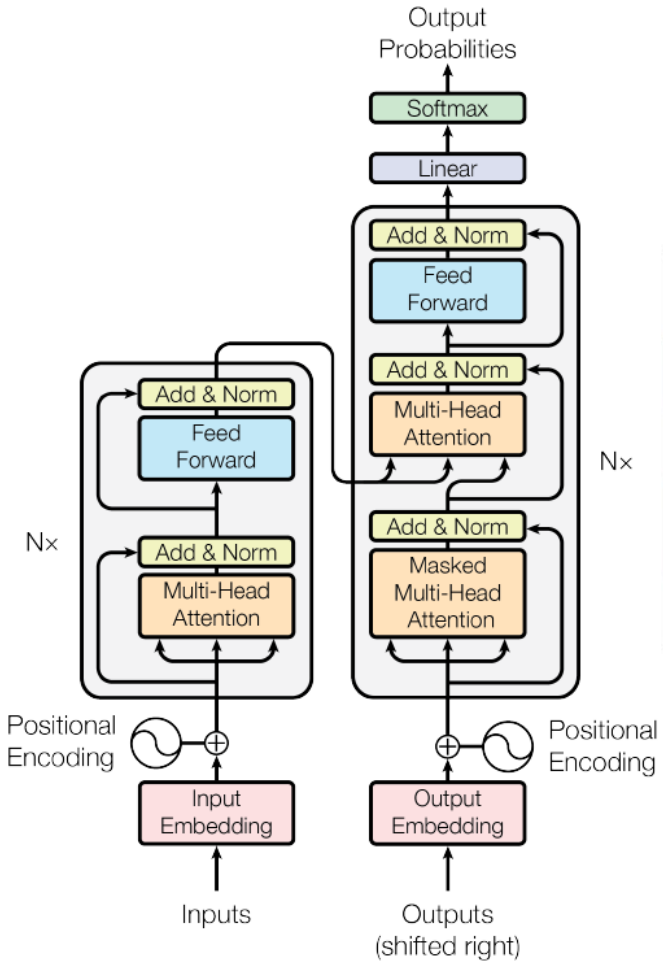
Snippets

测试集准确率:92.91%



基于注意力机制的深度学习文本分类模型

4. Transformer

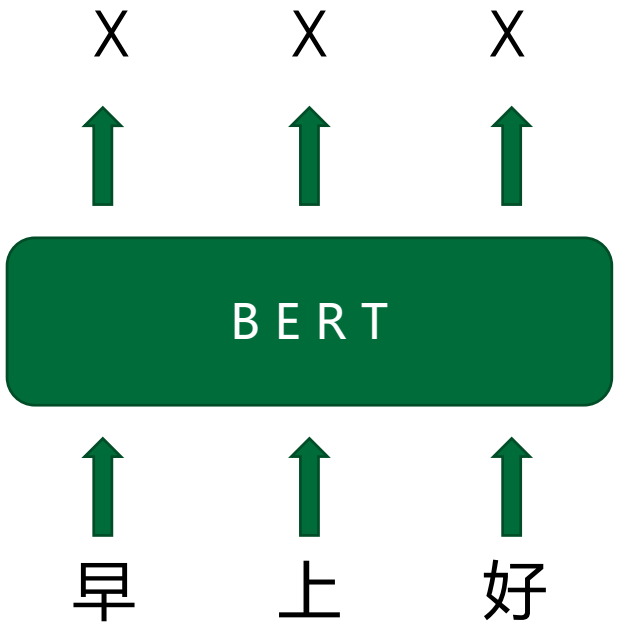


	ARIMA	Prophet	LSTM	Conv	Transformer
Long Input	✓	✓	×	✓	×
Long Output	×	×	×	×	×
Complexity / layer	$O(L)$	Not clear	$O(L * d^2)$	$O(k * L * d)$	$\boldsymbol{O(L^2 * d)}$
Max Path			$O(L)$	$O(log_k L)$	$\boldsymbol{O(1)}$ ✓

3-5

基于**语言模型**的深度学习文本分类模型

1. BERT —— 双向Transformer编码表达

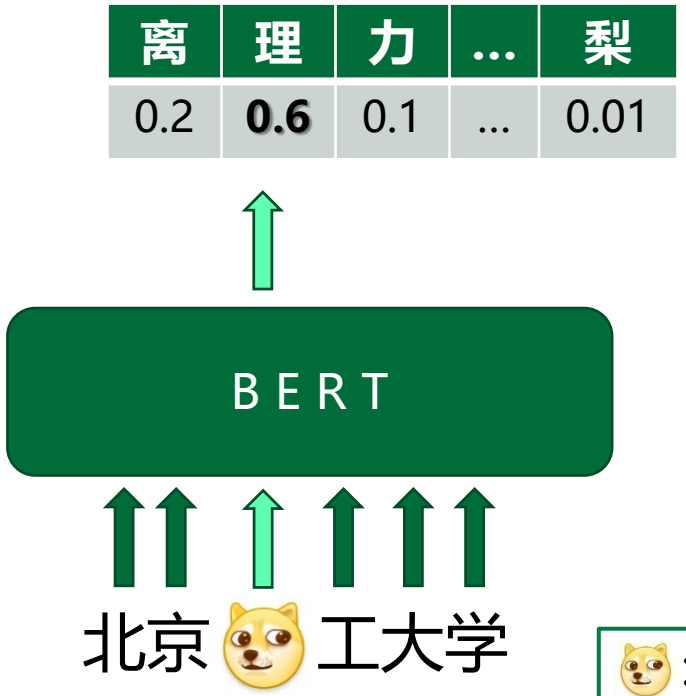


1. 这是一个**预训练**好的模型，输入和输出长度一样

2. 这个模型本身并不是专门用来做文本分类的

3. 将它的输出层进行一些**微调**即可用于文本分类等下游任务， 效果还不错！

如何训练?

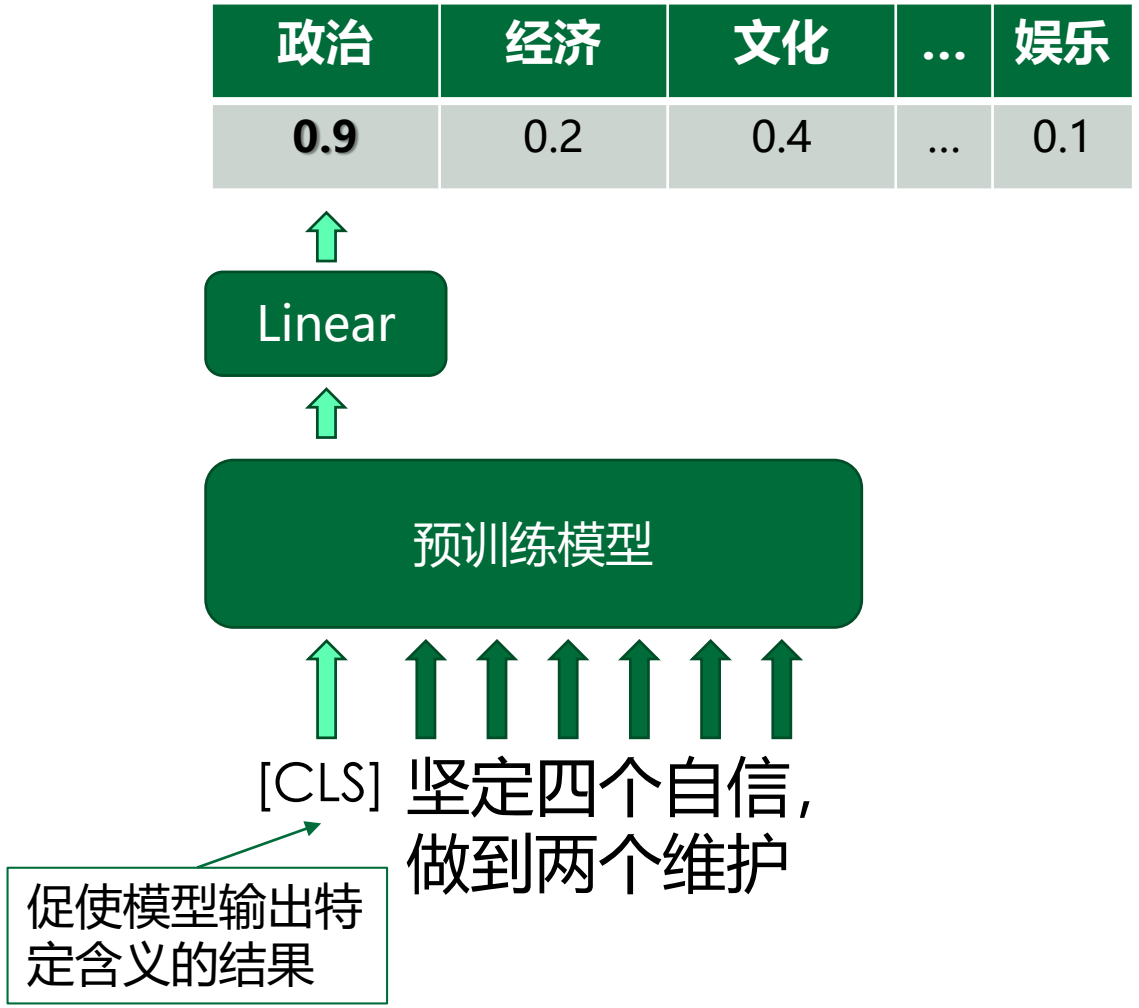


方法1：通过让BERT模型进行文本预测，即给定一段话，随机遮盖或替换其中的一些词语，让模型预测对应正确的词语是什么。(完形填空)

方法2：给定两个句子，让BERT模型判断这两个句子是否关联，即B这句话是否为A这句话的下一句。

：既可以是固定遮挡字符也可以是任意一个字符

基于语言模型的深度学习文本分类模型



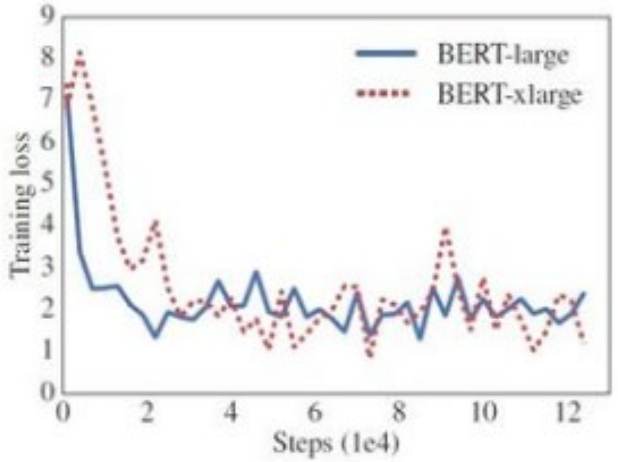
基于语言模型的深度学习文本分类模型 —— ALBERT

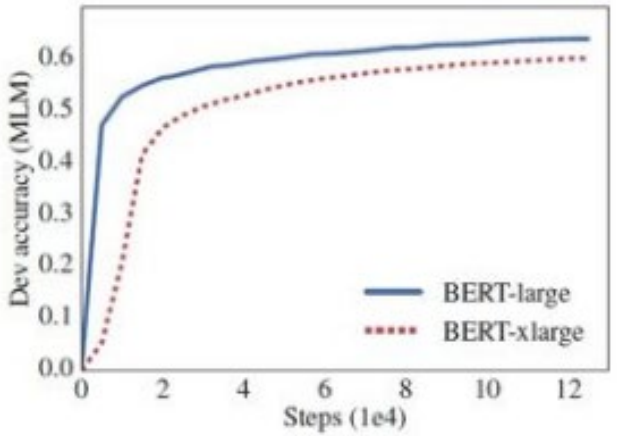
1. 介绍：一个精简的 BERT

BERT存在的问题：

参数量过高，对算力要求也高

参数量过高并不一定能带来性能的提升！





ALBERT当中的改进点：

一任务、两方法

一任务： SOP任务代替BERT中的NSP任务

SOP：不仅预测后面句子是不是前一个句子的下一个句子，还预测前面句子是不是后面句子的上一个句子。

NSP：仅预测后面的句子是不是前一个句子的下一个句子。

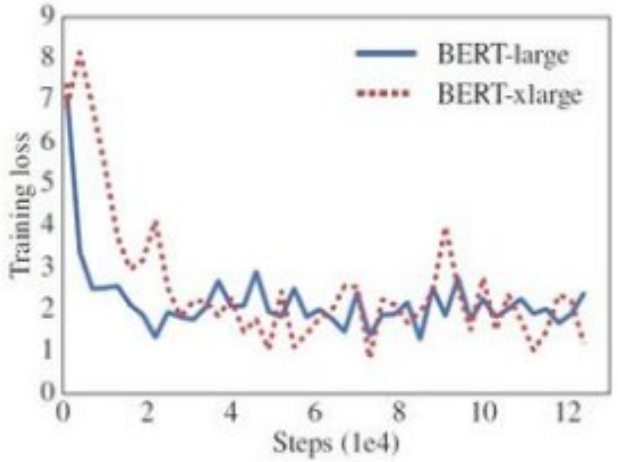
基于语言模型的深度学习文本分类模型 —— ALBERT

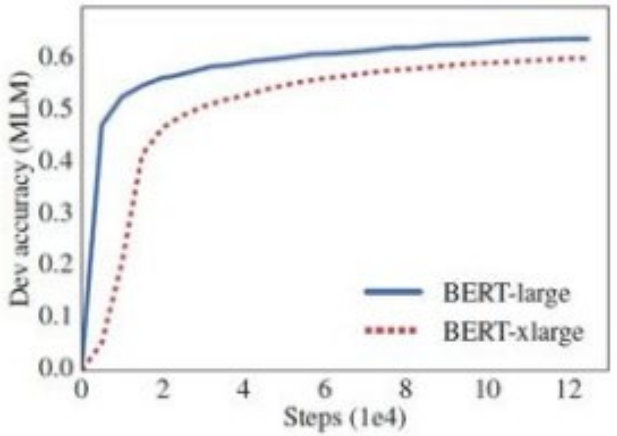
1. 介绍：一个精简的 BERT

BERT存在的问题：

参数量过高，对算力要求也高

参数量过高并不一定能带来性能的提升！





ALBERT当中的改进点：

—任务、两方法

两方法（减少参数）：

1. 对嵌入参数化进行因式分解。先将**大量**one-hot向量映射到低维度空间，再映射到隐藏空间
2. 参数共享。共享所有层的所有参数。

基于语言模型的深度学习文本分类模型 —— ALBERT

ALBERT——DEMO

```
***** Eval results
/home/lab303/qdmxy/CLUE/baselines/models/albert/tnews_output/model.ckpt-
157000 *****eval_accuracy = 0.1089eval_loss = 2.5990453global_step =
157000loss = 2.5990453***** Eval results
/home/lab303/qdmxy/CLUE/baselines/models/albert/tnews_output/model.ckpt-
158000 *****eval_accuracy = 0.1089eval_loss = 2.5990145global_step =
158000loss = 2.5990145***** Eval results
/home/lab303/qdmxy/CLUE/baselines/models/albert/tnews_output/model.ckpt-
159000 *****eval_accuracy = 0.1089eval_loss = 2.5990093global_step =
159000loss = 2.5990093***** Eval results
/home/lab303/qdmxy/CLUE/baselines/models/albert/tnews_output/model.ckpt-
160000 *****eval_accuracy = 0.1089eval_loss = 2.5987792global_step =
160000loss = 2.5987792***** Eval results
/home/lab303/qdmxy/CLUE/baselines/models/albert/tnews_output/model.ckpt-
160080 *****eval_accuracy = 0.1089eval_loss = 2.5987782global_step =
160080loss = 2.5987782
```

准确率仅为10%，远低于官方训练的结果（57.36%）
分析原因可能是因为受到显存限制，batch_size被我调整为1，因此降低了训练效果

基于语言模型的深度学习文本分类模型 —— ALBERT

ALBERT——DEMO

为什么batch_size低了效果会差?

- batch_size太低每次梯度下降修正方向均以每一个样本为准，这导致收敛困难，震荡严重;

batch_size越高越好吗?

- 不是的，太高的batch_size会导致显存占用过高，且对参数的修正会变得缓慢

```
--train_batch_size=1 \
--learning_rate=2e-5 \
--num_train_epochs=3.0 \
```

```
--train_batch_size=1 \
--learning_rate=2e-7 \
--num_train_epochs=10.0 \
```

基于语言模型的深度学习文本分类模型 —— ALBERT

ALBERT——DEMO

***** Eval results /home/lab303/qdmxy/CLUE/baselines/models/albert/tnews_output/model.ckpt-533600 *****

eval_accuracy = 0.5491

eval_loss = 2.3788922

global_step = 533600

loss = 2.3788922

基于语言模型的深度学习文本分类模型 —— UNILM

1.UNLIM ——自然语言理解与生成的统一预训练语言模型：

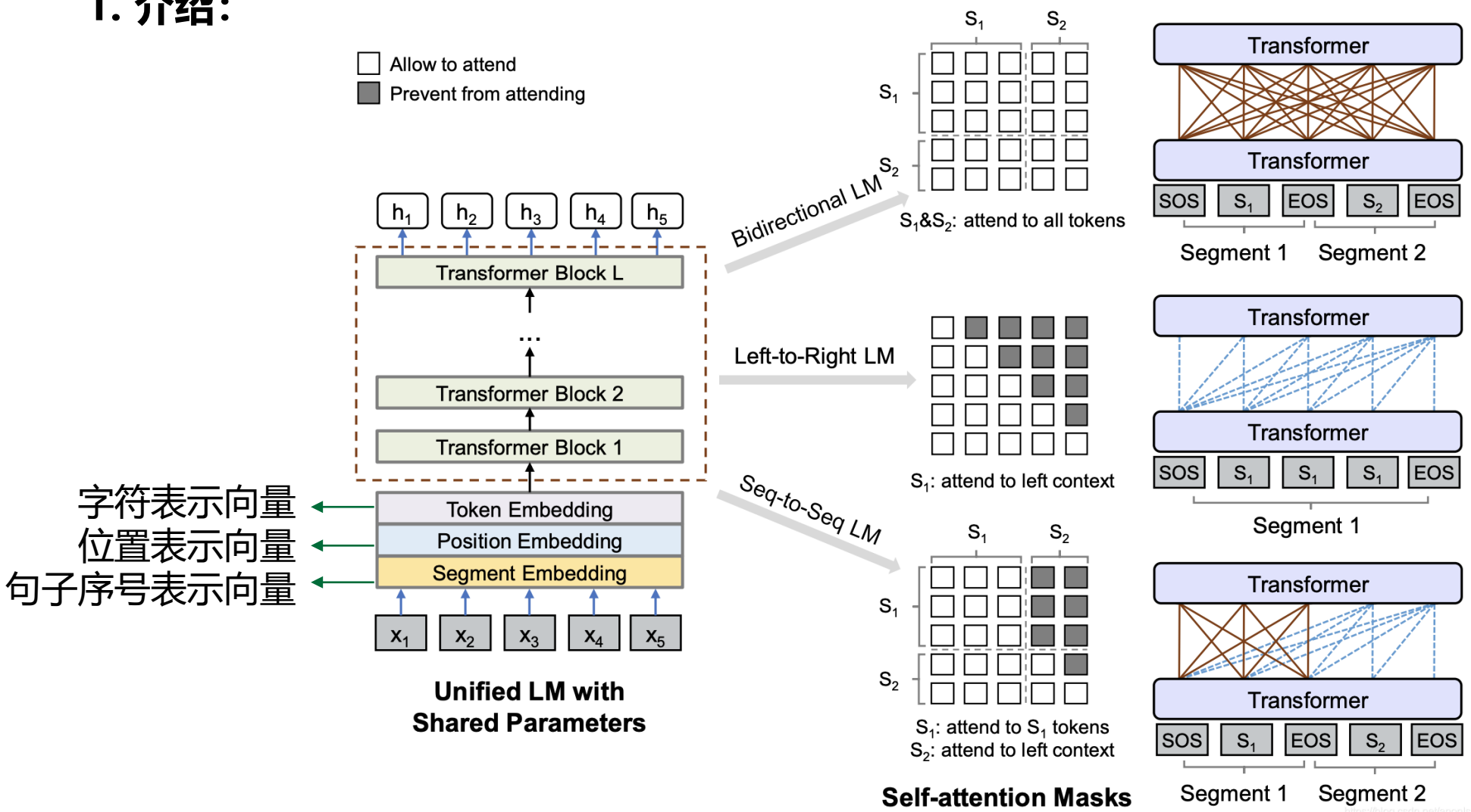
① 统一的预训练过程,使用一个transformer语言模型囊括了了不同类型的语言模型的的参数和结构(bi, left-t-right,right-t-left,seq-t-seq),从而不需要分开训练多个语言模型。

② 参数共享使得他学习到的文本表征更加通用化了,针对不同的语言建模目标进行联合优化,上下文信息会以不同的方式去使用,所以可以减少自然语言任务训练中的过拟合。

③ 因为UNILM有seq-to-seq的用法,所以天生适用自然语言生成任务,比如摘要提取,问题生成。

基于语言模型的深度学习文本分类模型 —— UNILM

1. 介绍:



字符表示向量
位置表示向量
句子序号表示向量

训练时:
三分之一的时间用**Bidirectional LM**
三分之一的时间用**Sequence-to-Sequence LM**
从左到右的单向和从右到左的单向各占六分之一的时间

训练建议

1. 根据规模和算力限制选择合适的预训练模型。
2. 选择与本次任务领域相近的预训练数据训练的模型。如果相差较大可能要重新进行预训练。
3. 采用多任务训练有助于提升模型泛化能力。多任务学习就是在各个任务具有相关性时，同时学习多个任务，来提高单任务学习的效果。
4. 对模型适当压缩。

4 实验探究

汇报人：周伊凡

实验探究

- THUCNews

  文本长度：20-30

  训练集数量：16w

  测试集数量：4w

  类别（10类）：财经、房产、股票、教育、科技、社会、时政、体育、游戏、娱乐。

- TNEWS'今日头条中文新闻（短文）分类

  文本长度：20-30

  训练集数量：26.6w

  测试集数量：5.7w

  类别（15类）

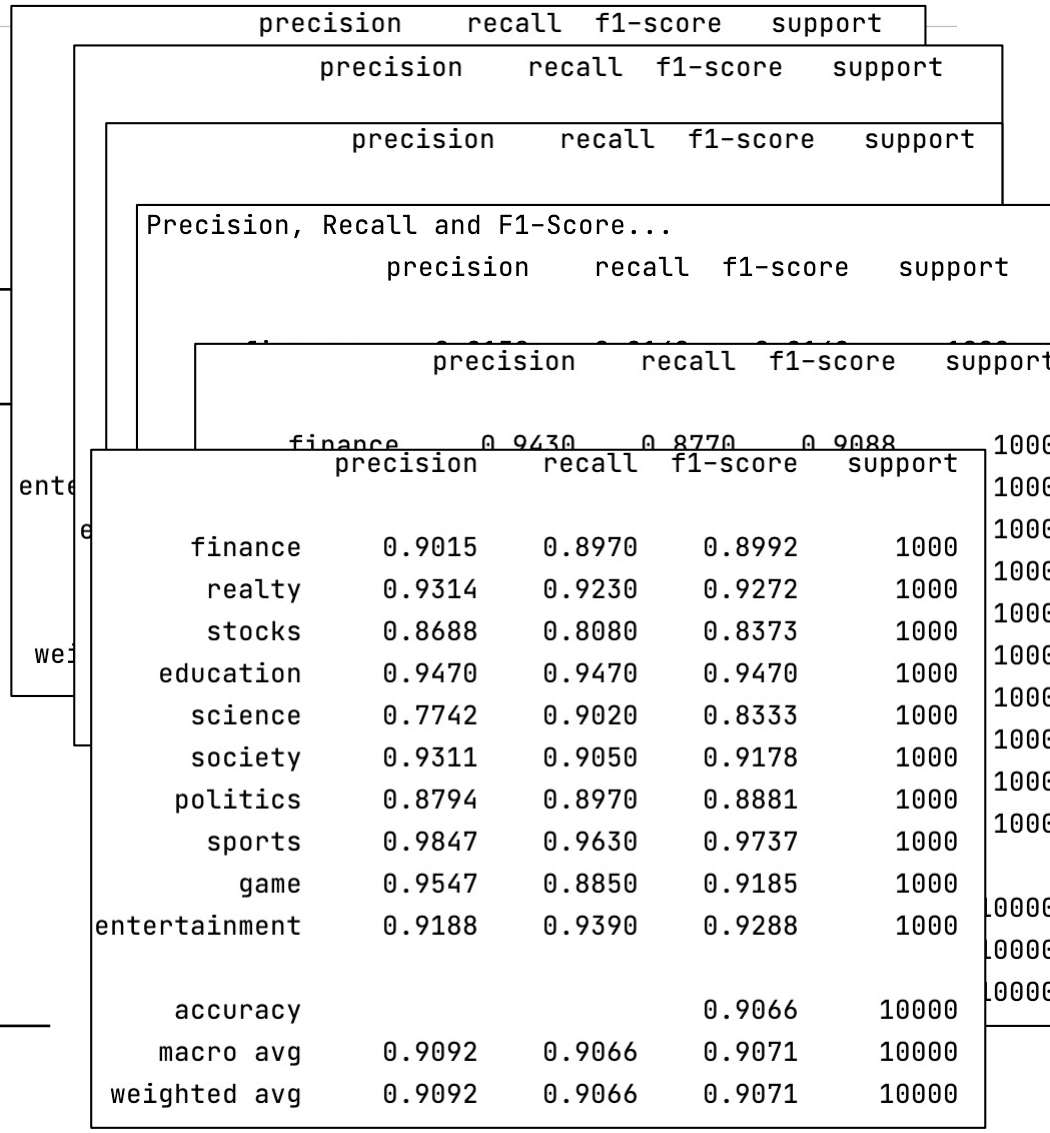
词汇阅读是关键 08年考研暑期英语复习全指南 3
中国人民公安大学2012年硕士研究生目录及书目 3
日本地震：金吉列关注在日学子系列报道 3
名师辅导：2012考研英语虚拟语气三种用法 3
自考经验谈：自考生毕业论文选题技巧 3
本科未录取还有这些路可以走 3
2009年成人高考招生统一考试时间表 3
去新西兰体验舌尖上的饕餮之旅(组图) 3
四级阅读与考研阅读比较分析与应试策略 3
备考2012高考作文必读美文50篇(一) 3
名师详解考研复试英语听力备考策略 3
热议：艺考合格证是高考升学王牌吗(组图) 3
研究生办替考网站续：幕后老板年赚近百万(图) 3
2011年高考文科综合试题(重庆卷) 3
56所高校预估2009年湖北录取分数线出炉 3
公共英语(PETS)写作中常见的逻辑词汇汇总 3

{"label": "106", "label_desc": "news_house", "sentence": "通过中介公司买了二手房，首付都付了，现在卖家不想卖了。怎么处理？", "keywo
{"label": "112", "label_desc": "news_travel", "sentence": "2018年去俄罗斯看世界杯得花多少钱？", "keywords": "莫斯科,贝加尔湖,世界
{"label": "109", "label_desc": "news_tech", "sentence": "剃须刀的个性革新，雷明登天猫定制版新品首发", "keywords": "剃须刀,绝地求生
{"label": "103", "label_desc": "news_sports", "sentence": "再次证明了“无敌是多么寂寞”—逆天的中国乒乓球队！", "keywords": "世乒赛
{"label": "109", "label_desc": "news_tech", "sentence": "三农盾SACC-全球首个推出：互联网+区块链+农产品的电商平台", "keywords": "
{"label": "116", "label_desc": "news_game", "sentence": "重做or新英雄？其实重做对暴雪来说同样重要", "keywords": "暴雪,重做,新英雄
{"label": "103", "label_desc": "news_sports", "sentence": "如何在商业活动中不受人欺骗？", "keywords": ""}
{"label": "101", "label_desc": "news_culture", "sentence": "87版红楼梦最温柔的四个丫鬟，娶谁都是一生的福气", "keywords": "欧阳奋强
{"label": "109", "label_desc": "news_tech", "sentence": "凌云研发的国产两轮电动车怎么样，有什么惊喜？", "keywords": ""}
{"label": "106", "label_desc": "news_house", "sentence": "房地产税迟迟无法出台？央行研究局局长徐忠这样说", "keywords": "土地市场,
{"label": "107", "label_desc": "news_car", "sentence": "我四千一个月，老婆一千五一个月，存款八万且有两小孩，是先买房还是先买车？", "k
{"label": "104", "label_desc": "news_finance", "sentence": "“产地办展”模式为“东莞制造”送创新情报", "keywords": "深圳国际,展览会,
{"label": "104", "label_desc": "news_finance", "sentence": "全国首个央地融合平台在沪落地", "keywords": "中国电建,世博地区,太平洋,
{"label": "100", "label_desc": "news_story", "sentence": "故事：刘主任建猪场", "keywords": "刘大柱,青蛇,打谷场,表姐家,那条大蛇"}
{"label": "102", "label_desc": "news_entertainment", "sentence": "什么是人情，什么是世故？", "keywords": ""}

实验探究

模型对比

模型	THUCNews	备注
TextCNN	90.73%	CNN文本分类（Kim 2014
TextRNN	90.68%	BiLSTM
TextRNN_Att	90.08%	BiLSTM+attention
TextRCNN	91.53%	BiLSTM+池化
FastText	91.88%	bow+bigram+trigram
DPCNN	91.25%	深层金字塔CNN
Transformer	90.66%	self-attention
Bert	94.83%	bert + fc



实验探究

- 模型对比

模型	参数	TNEWS'	IFLYTEK'
BERT-base	108M	56.58	60.29
BERT-wwm-ext	108M	56.84	59.43
ERNIE-base	108M	58.33	58.96
RoBERTa-large	334M	57.86	62.55
XLNet-mid	200M	56.24	57.85
ALBERT-xxlarge	235M	**59.46**	62.89
ALBERT-xlarge	60M	57.36	59.50
ALBERT-large	18M	55.16	57.00
ALBERT-base	12M	55.06	56.58
ALBERT-tiny	**4M**	53.35	48.71
RoBERTa-wwm-ext	108M	56.94	60.31
RoBERTa-wwm-large	330M	58.61	**62.98**

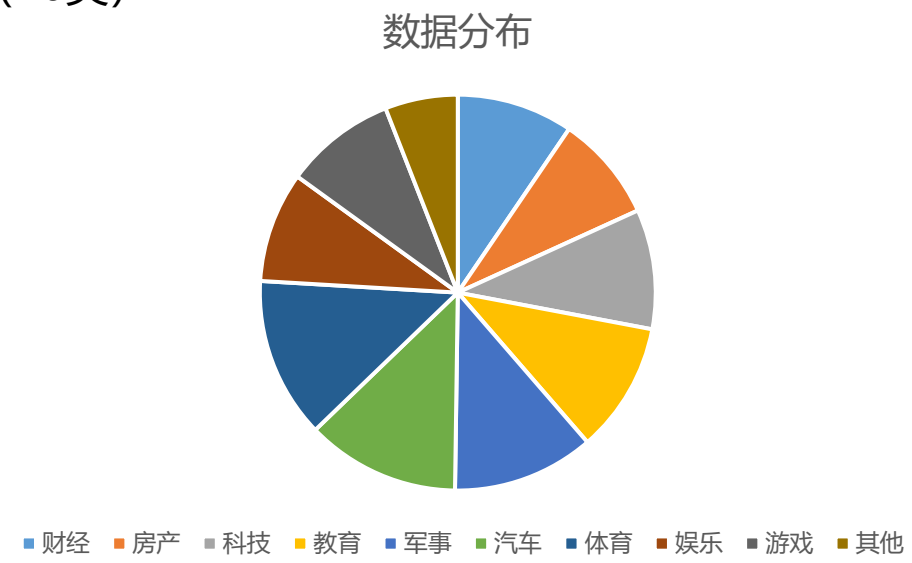
“中国软件杯”大学生软件设计大赛新闻文本分类赛题数据集

文本长度：不定长度，数据处理后为512

训练集数量：1w

测试集数量：1w

类别（10类）

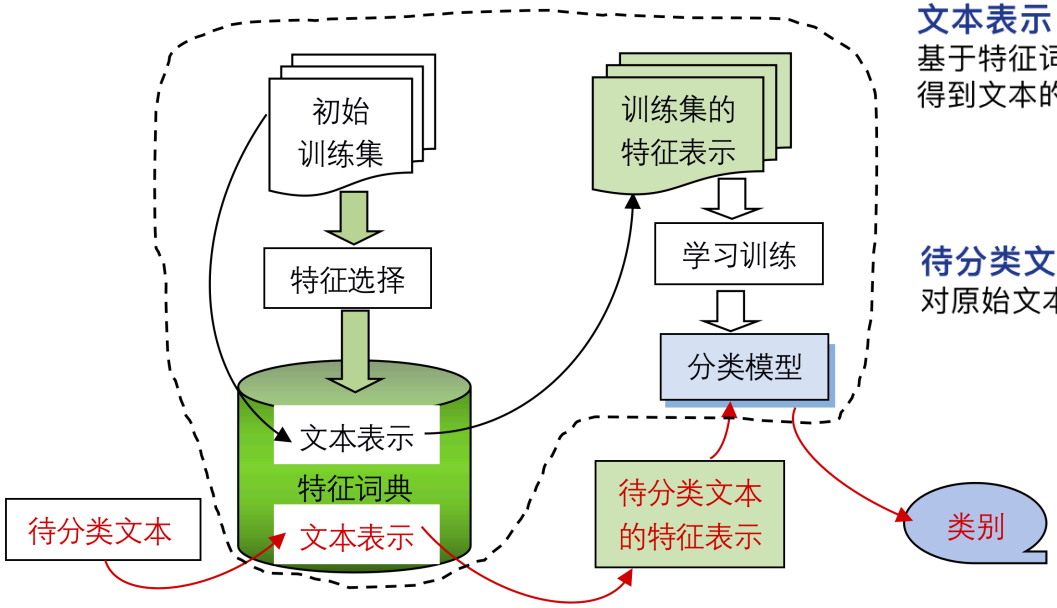


数据处理流程：

- 删除：删除重复数据
- 筛选：筛选语义相似的数据，仅保留代表性数据样本
- 添加：在央视网、百度新闻、新华网、腾讯新闻等各大新闻网站上爬取最新数据，添加到训练集中
- 转化：合并文本标题与内容，截取文本长度（512字符），权衡标题与内容权重，将Excel数据转换成txt格式

实验探究

系统模型图及模块划分



文本原始特征提取
分词、去停用词、英文的词根还原等处理，得到文本的原始特征表示

1

2

特征词典建立
词频统计，特征选择以降低特征向量的维度，建立文本数据的特征空间

文本表示
基于特征词典进行词频统计/权值计算，得到文本的特征表示

3

4

分类模型的学习训练
利用SVM分类算法进行学习训练，得到分类模型

待分类文本预处理
对原始文本数据进行数据清洗等处理

5

6

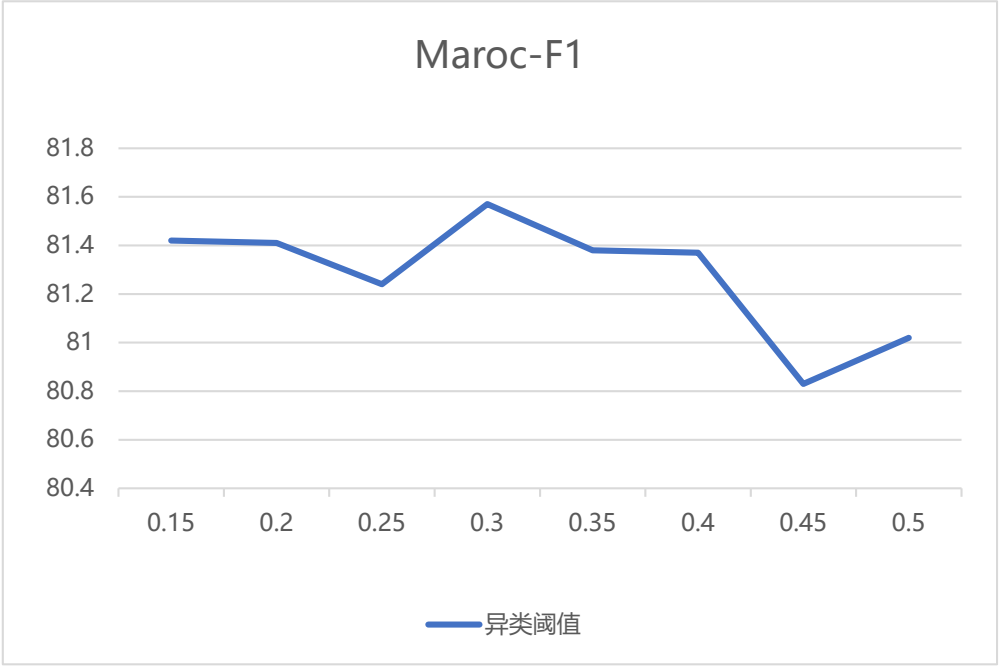
文本分类
使用分类模型分类，得到分类结果

单条新闻文本分类平均时间不超过5s

长文本分类F1值达到85.8%

问题探究一：异分类

通过设定阈值的办法，实现了利用SVM进行其他类分类即异分类



问题探究二：集成学习思想

将集成学习的思想融入分类系统中；分类前利用有放回随机采样策略建立不同的子特征空间；分类后利用投票机制决定最终结果

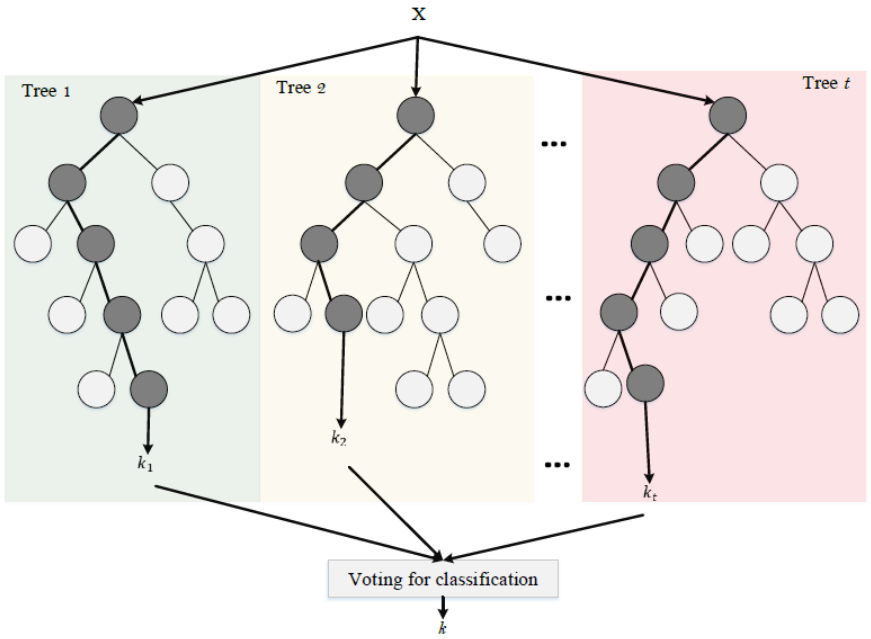
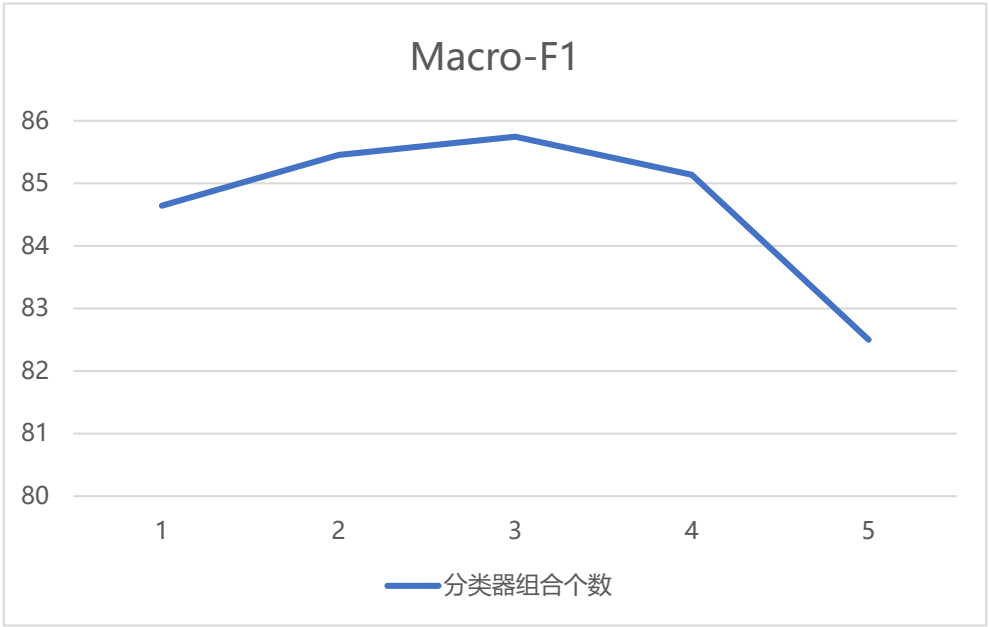


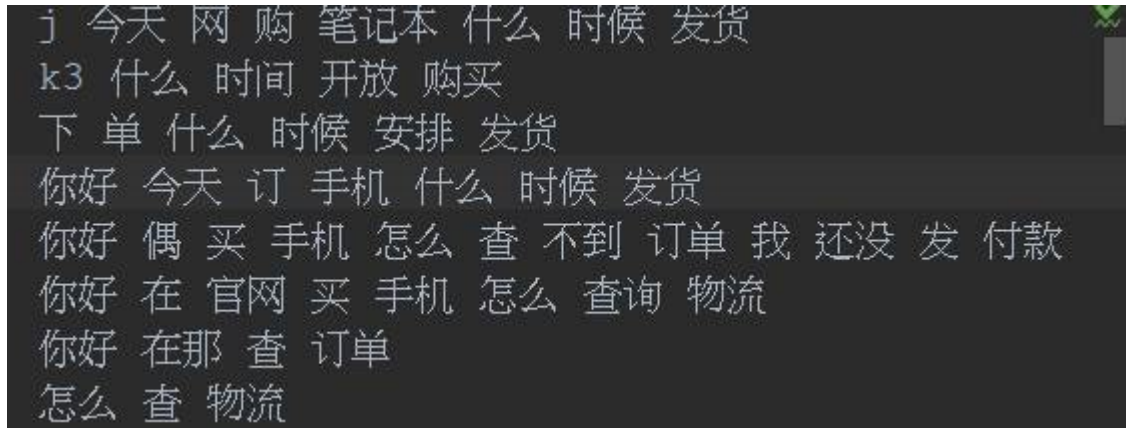
Figure 14. Random forest.



5 文本分类应用与商业价值

汇报人：李雪薇

文本分类应用与商业价值



- 客服/聊天话题分类：自动识别客服跟用户聊天过程中用户反馈的问题类别，如是退货问题、物流问题、商品质量问题等。
- 新闻分类：自动判断新闻类别，如政治类、经济类、民生类、体育类等。
- 邮件自动打标签：如自动识别邮件是不是垃圾邮件。

文本分类应用与商业价值

- 内容审核之灌水评论检测：检测评论是否是灌水评论。
- 购买意愿识别：根据用户发表的微博等信息判
是否有购买某商品的意愿。
- 自杀倾向预测：根据用户的社交媒体发
息，自动识别用户是否有抑郁或自杀倾
- 事件类型分类：自动判断事件的类型，
域的新闻报道中涉及到的事件类别自动
任、辞职、增持、减持、会议召开。

前沿技术：数据

- 零样本/少样本数据：半监督学习+文本增强

半监督学习中一致性正则能够充分利用大量未标注数据，同时能够使输入空间的变化更加平滑，从另一个角度来看，降低一致性损失实质上也是将标签信息从标注数据传播到未标注数据的过程

文本增强提供了原有标注数据缺少的归纳偏差，在少样本场景下通常会取得稳定、但有限的性能提升

- 引入外部知识：

将预训练模型引入知识图谱

面向任务的精调阶段引入知识

文本匹配中引入知识图谱

Query
新冠肺炎可以通过完好的皮肤传播吗
IPhone**手机**多少钱

前沿技术：模型

- XLNet：谷歌的最新模型，XLNet在NLP的主要任务上比如文本分类，情感分析，问答，以及自然语言推理上都达到最先进的水平。核心思想是：语言理解的广义自回归预训练和Transformer-XL结构。
- ERNIE：语言理解模型。解决了外部知识纳入语言表示的两个挑战：**构化知识编码**和**异构信息融合。**
- Text-to-Text Transfer Transformer (T5)：把所有的问题都套进去一个统一的范式，从而可以采用同样的模型架构、同样的训练策略、同样的损失函数、同样的解码手段。为整个 NLP 预训练模型领域提供了一个通用框架，把所有任务都转化成一种形式，语料库的大小仍然高达750GB。
- Binary Partitioning Transfomer (BPT)：将Transformer 视为一个图神经网络，旨在提高自注意力机制的效率。
- Rethinking Complex Neural Network Architectures：简单的调优的模型。不使用注意力机制。结合LSTM +正则化方法进行文档分类的论文

前沿技术：性能

- 模型的语义鲁棒性：

语言多样性、口语特征和噪声扰动

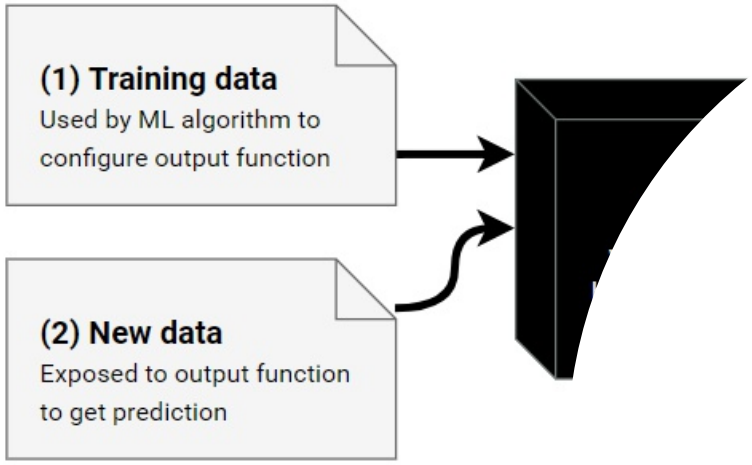
采用数据增广（Augmentation）的方式来作为测试手段：词扰动，同义复述，模拟语音识别，口语不流畅。

- 模型的可解释性:

理解模型部件的功能属性

解释模型预测的行为: 梯度方法, 注意力方法

生成支撑决策的依据



谢谢聆听

请各位老师批评指正