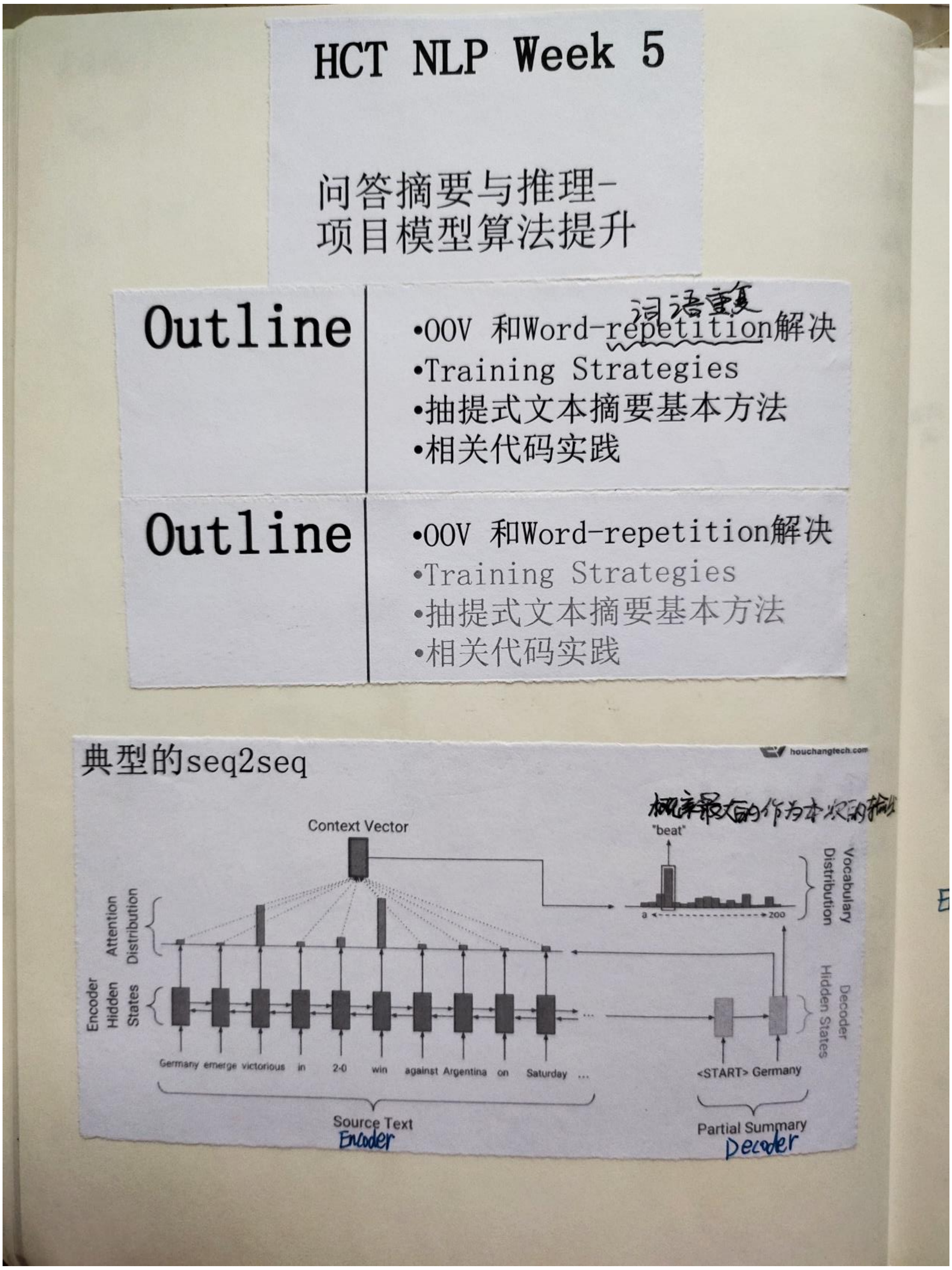
第五章 问答摘要与推理-项目模型算法提升



HCT NLP Week 5

问答摘要与推理-
项目模型算法提升

Outline

- OOV 和Word-repetition解决 词语重复
- Training Strategies
- 抽提式文本摘要基本方法
- 相关代码实践

Outline

- OOV 和Word-repetition解决
- Training Strategies
- 抽提式文本摘要基本方法
- 相关代码实践

典型的seq2seq

houchangtech.com
概率最大的作为本次的输出
Context Vector
"beat"
a
200
Vocabulary Distribution
Attention Distribution
Encoder Hidden States
Decoder Hidden States
Germany emerge victorious in 2-0 win against Argentina on Saturday ...
<START> Germany
Source Text
Encoder
Partial Summary
Decoder

问题

The encoder is not well trained via back propagation through time

从模型的路径上看，encoder到实际输出有一定距离，从此限制了反向传播。

OOV（Out-of-vocabulary未登录词 ）

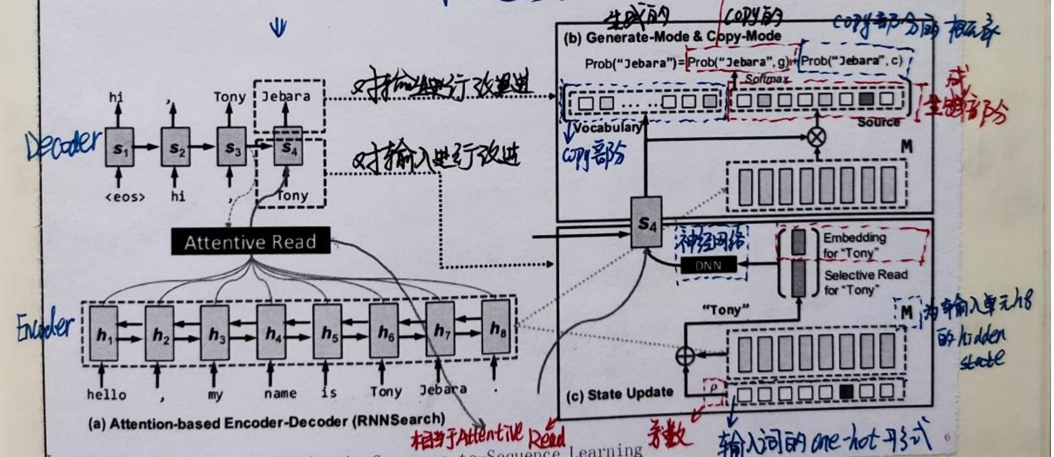
摘要总结的结果有的时候并不准确，比如摘要的结果可能输出德国队以2-1比分击败阿根廷，但是实际比分是2-0，出现这个的原因是out-of-vocabulary words（OOV）的出现

Word-repetition问题 词语级别的重复（不是句子重复）

摘要结果会出现repeat重复的信息，比如重复出现德国队击败阿根廷队

CopyNet

对S4单元的输入和输出进行改进

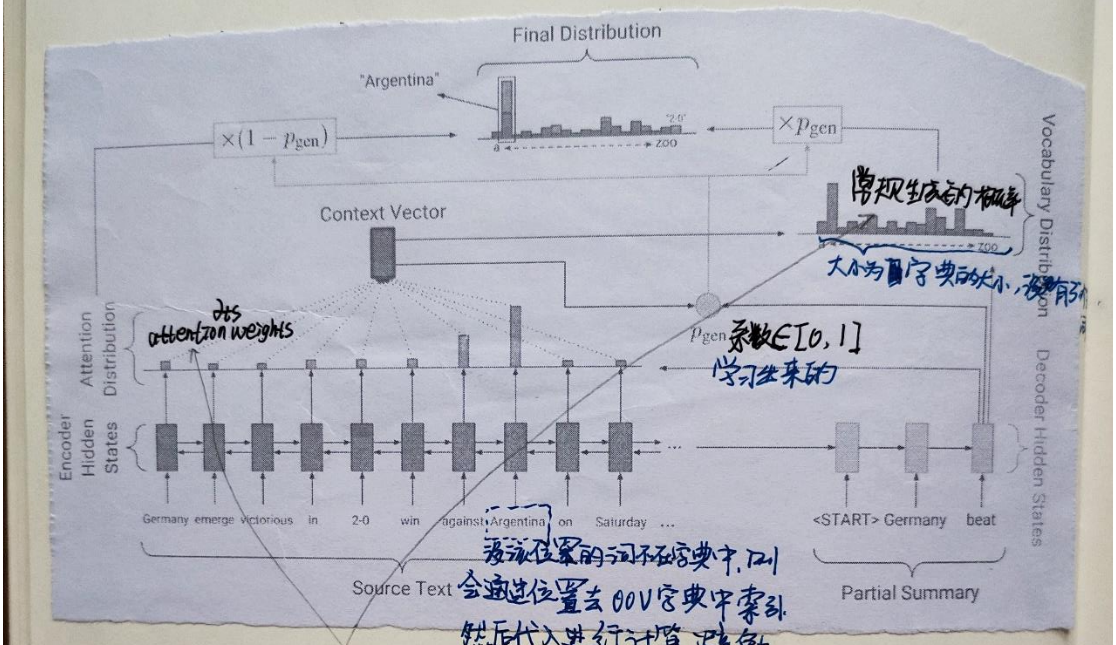


Incorporating Copying Mechanism in Sequence-to-Sequence Learning

输入词的one-hot形式乘以系数ρ，再与h8的hidden state相加

PGN (Pointer-Generator ~~Atten~~ Networks)

用于解决OOV问题



常规生成的概率

大小为字典的大小，没有

p_{gen} 系数$\in[0,1]$

学习出来的

α_{ts} attention weights

设该位置的词不在字典中，则会通过位置去OOV字典中索引，然后代入进行计算，避免该位置为<unk>

$$P_{gen} = \sigma(W_h^T \times h_t + W_s^T \times S_t + W_x^T \times x_t + b)$$

σ → sigmoid $\in(0,1)$

h_t → Context vector

S_t → decoder hidden

x_t → 输入词的词向量embedding

b → 偏置

写为W_1、W_2、W_3也行，只是写法问题

W_h^T、W_s^T、W_x^T为神经网络的权重。

$$P(w) = P_{gen} \cdot P_{vocab}(w) + (1-P_{gen})\sum_{i:w_j=w} \boxed{\alpha_{ts}^{w}}$$

$P_{vocab}(w)$ → 常规输出生的概率

→ 即各输入词对应的attention weight

PGN 特点

1. pointer-generator network能够很容易的复制输入的文本内容，可以通过Pgen 来调节。

2. pointer-generator network能够从输入的文本内容中复制OOV词汇，这是最大的优点，这个也可以采用更小的词汇表vocabulary ，较少计算量和存储空间。

3. pointer-generator network训练会更快，在seq2seq训练过程中用更少的迭代次数就能取得一样的效果。

Get To The Point: Summarization with Pointer-Generator Networks

Repetition Handling 解决词语的重复问题

model generated summaries suffer from both word-level and sentence-level repetitions.

Temporal Attention

Intra-decoder Attention

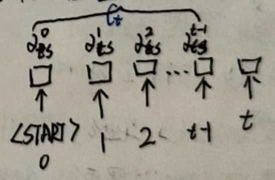
} 这两种方法已用的很少

☆Coverage(机器翻译中用得多) 减少生成重复词，对已生成的词做一个惩~~罚~~罚

Coverage：

① $\overset{\text{coverage}}{C_i^t} = \sum_{t=0}^{t-1} \alpha_{ts}$ 即 t位置的 coverage 等于从 0位置到 t-1位置的 attention weights 的~~[illegible]~~和

其中 $C_0 = 0$，$C_1 = 1$；C_t为一个非标准的分布



② $e_i^t = U^T \cdot \tanh(W_s h_s + W_t h_t + W_c C_i^t + b)$

e_i^t → 分数(score)

U^T → 权重、<全连接层 Dense()>

tanh → tanh 能记录更多东西，用概率控制时用sigmoid

$W_s h_s$ → encoder hidden state

$W_t h_t$ → decoder hidden state

C_i^t → coverage

b → 偏置

注：一层的全连接层为一个矩阵[参数]

trick：在预测时加 tri-gram block

③ $CovLoss_t = \sum \min(\alpha_{ts}^t, C_i^t)$ <coverage loss>

$Loss_t = -\log P(W_t) + \lambda \sum \min(\alpha_{ts}^t, C_i^t)$ <最终loss>

λ ↑ 固定的系数值，如 0.5，1

Outline

- OOV 和Word-repetition解决
- Training Strategies
- 抽提式文本摘要基本方法
- 相关代码实践

TRAINING STRATEGIES 训练策略

A. Word-Level Training 词语级别的训练

two different methods for avoiding the problem of exposure bias.

1) Cross-Entropy Training (XENT) 交叉熵（loss函数采用交叉熵）

teacher forcing 方法 会造成 exposure bias

2) Scheduled Sampling 避免 exposure bias

是一种解决训练和生成时输入数据分布不一致的方法。在训练早期该方法主要使用目标序列中的真实元素作为解码器输入，可以将模型从随机初始化的状态快速引导至一个合理的状态。随着训练的进行，该方法会逐渐更多地使用生成的元素作为解码器输入，以解决数据分布不一致的问题。该方法应用在模型的训练阶段，生成阶段不使用。

Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks

TRAINING STRATEGIES

B. Sequence-Level Training 序列（句子）级别的训练、

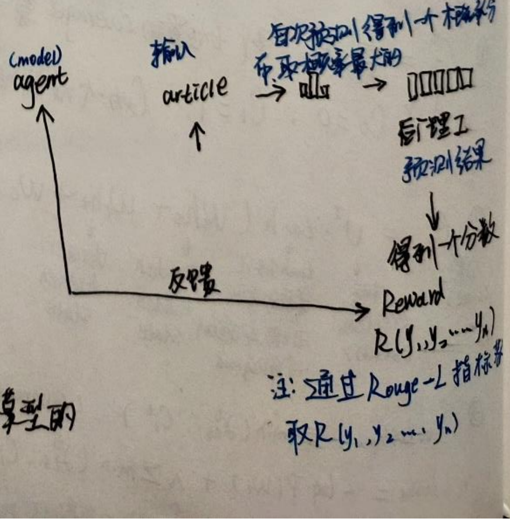
RL algorithms reinforcement learning 增强学习（强化学习）

强化学习过程如下：

① 定义一个模型 agent

② 初始化，并进行预测

③ 用预测结果更新 agent



注：通过Rouge-L指标来取R(y1,y2……yn)

注：强化学习资料：

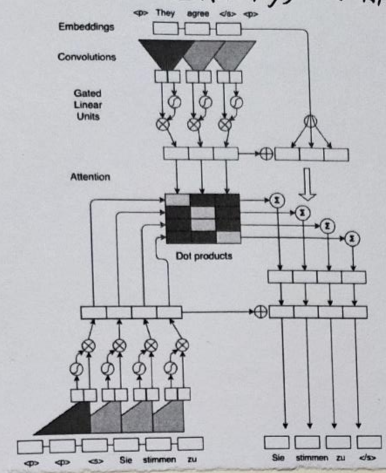
b站：周博磊

RL不是通过反向梯度求导来更新模型的

（没有交叉熵，没有loss概念）

Beyond RNN（现在用的不多了）（之前在翻译网络中用的较多）

CNN



Position Embedding

层叠CNN构成了hierarchical representation表示

融合了Residual connection、liner mapping的多层attention

采用GLU做为gate mechanism

进行了梯度裁剪和精细的权重初始化，加速模型训练和收敛

Convolutional Sequence to Sequence Learning

Beyond RNN为了解决GRU和LSTM不能并行计算的问题

CNN: Convolutional Neural Networks 卷积神经网络

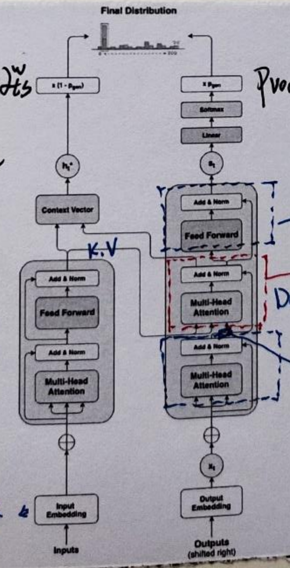
Beyond RNN

用Transformer替换 LSTM.gru等神经单元

注:视频中有transformer的基础介绍

$(1-P_{gen})\sum a_{ts}^{w}$

$P_{vocab} \times P_{gen}$



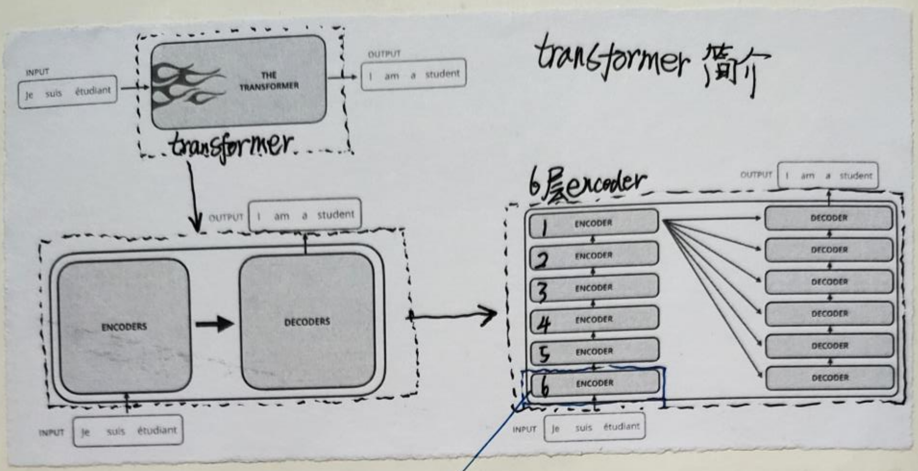
与Encoder中的结构相同

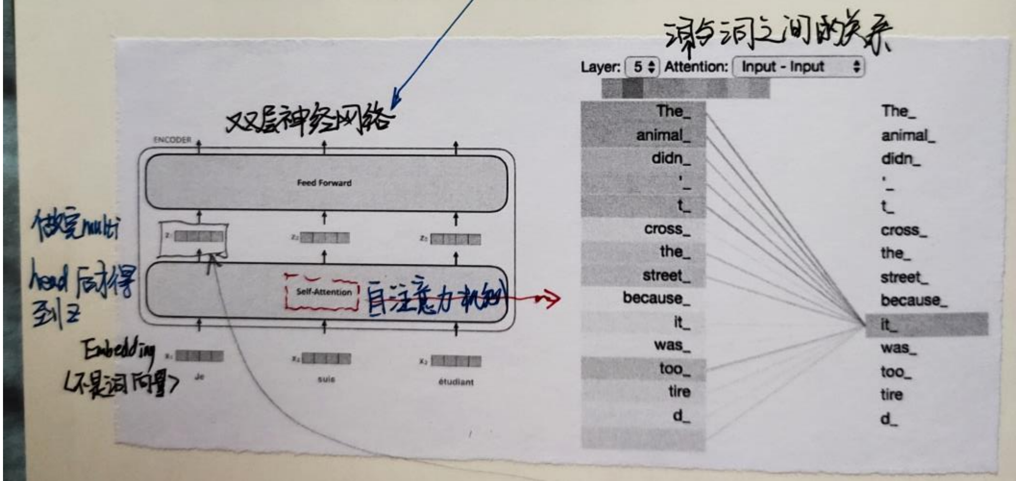
与Encoder中的结构有差异

此处的embedding不是词向量

Transformers and Pointer-Generator Networks f

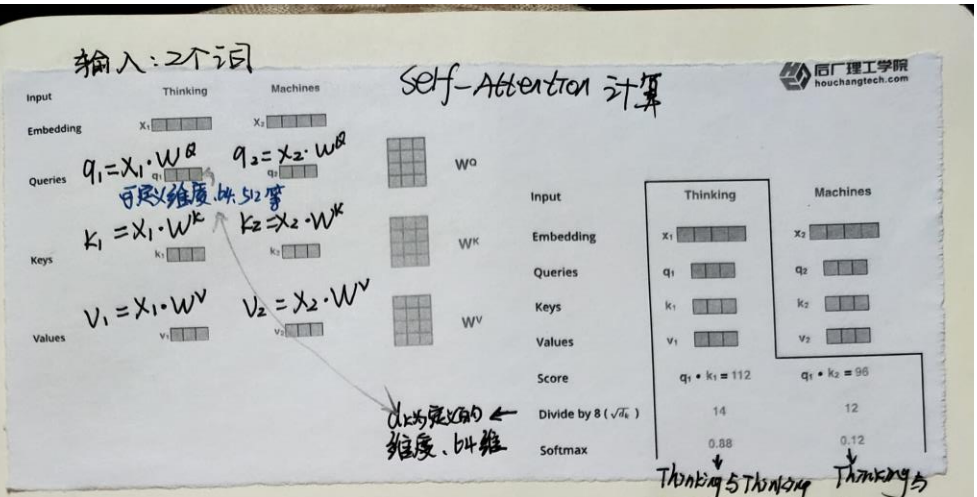
Abstractive Summarization


transformer 简介
INPUT
Je suis étudiant
THE TRANSFORMER
OUTPUT
I am a student
transformer
OUTPUT
I am a student
ENCODERS
DECODERS
INPUT
Je suis étudiant
6层encoder
OUTPUT
I am a student
1 ENCODER
2 ENCODER
3 ENCODER
4 ENCODER
5 ENCODER
6 ENCODER
DECODER
DECODER
DECODER
DECODER
DECODER
DECODER
INPUT
Je suis étudiant


双层神经网络
ENCODER
Feed Forward
做完multi head 后得到z
z_1
z_2
z_3
Self-Attention
自注意力机制
Embedding 〈不是词向量〉
x_1
x_2
x_3
Je
suis
étudiant
词与词之间的关系
Layer: 5
Attention: Input - Input
The_
animal_
didn_
'_
t_
cross_
the_
street_
because_
it_
was_
too_
tire
d_
The_
animal_
didn_
'_
t_
cross_
the_
street_
because_
it_
was_
too_
tire
d_

输入：2个词

Self-Attention 计算



$q_1 = x_1 \cdot W^Q$　$q_2 = x_2 \cdot W^Q$

可定义维度，64，512等

$k_1 = x_1 \cdot W^k$　$k_2 = x_2 \cdot W^k$

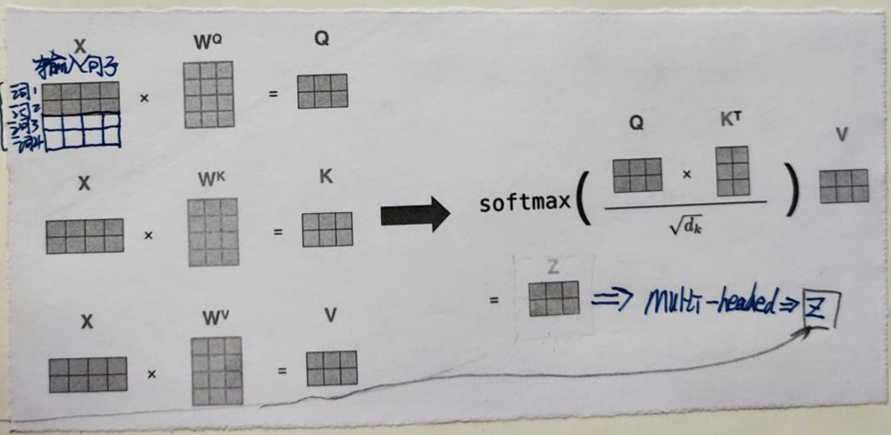
$v_1 = x_1 \cdot W^v$　$v_2 = x_2 \cdot W^v$

d_k为定义的维度、64维

Thinking与Thinking的关系

Thinking与Machine的关

由词构成句子



输入句子

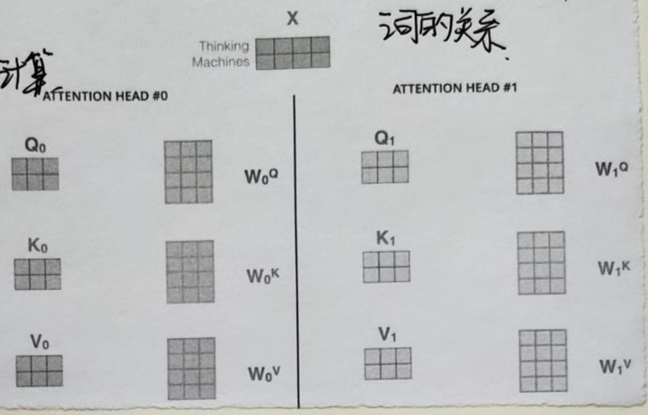
词1
词2
词3
词4

$$\mathrm{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right) V = Z \Rightarrow \text{Multi-headed} \Rightarrow \boxed{Z}$$

Multi-headed

重复多次self-Attention 操作，计算一个词与多个词的关系.

利用~~[illegible]~~多份进行计算
因为初始化 =>
矩阵不同.
计算结果也不同



Multi-headed

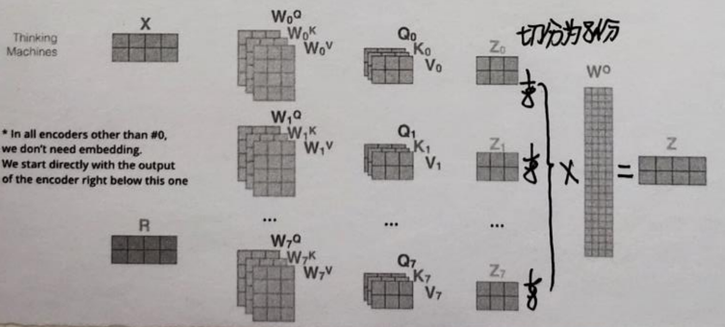
1) This is our input sentence*

2) We embed each word*

3) Split into 8 heads. We multiply X or R with weight matrices

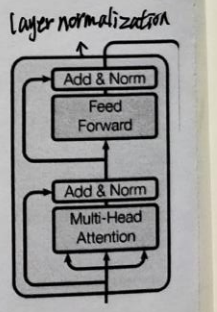
4) Calculate attention using the resulting Q/K/V matrices

5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer



切分为8份

* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one



Other Studies

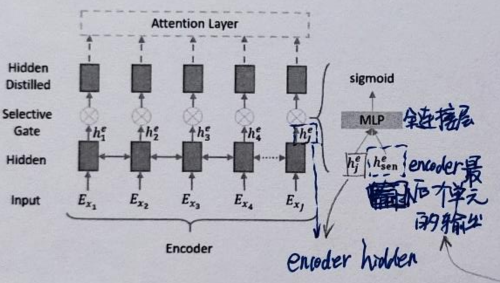
1) *Network Structure and Attention*

2) *Extraction + Abstraction*

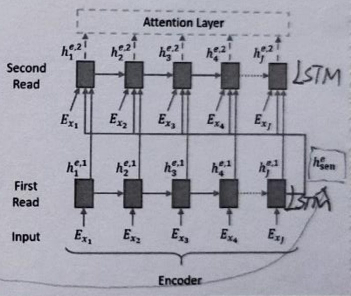
3) *Long Documents*

Improving Encoded Representations 对Encoder的改进

Selective Encoding



Read-Again Encoding



Improving Decoder 对Decoder的改进 思想.

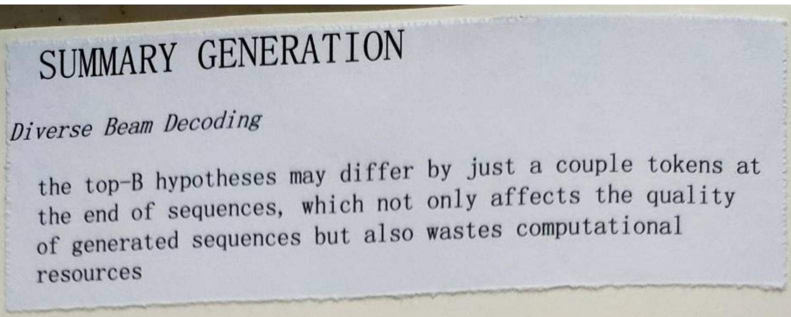
Embedding Weight Sharing 共享权重

Extraction + Abstraction

Extractor + Pointer-Generator Network

Key-Information Guide Network (KIGN)

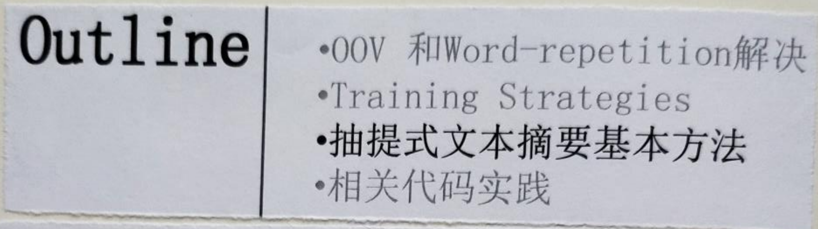
Reinforce-Selected Sentence Rewriting



SUMMARY GENERATION

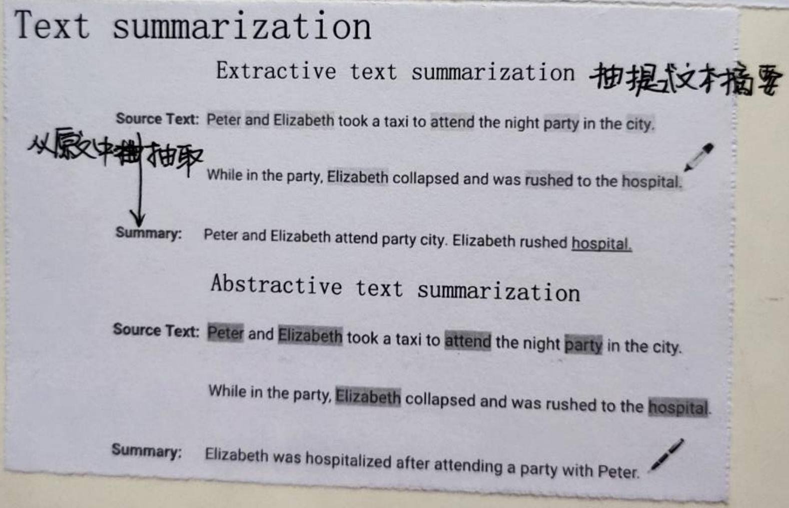
Diverse Beam Decoding

the top-B hypotheses may differ by just a couple tokens at the end of sequences, which not only affects the quality of generated sequences but also wastes computational resources



Outline

- •OOV 和Word-repetition解决
- •Training Strategies
- •抽提式文本摘要基本方法
- •相关代码实践



Text summarization

Extractive text summarization 抽提式文本摘要

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

从原文中~~抽~~抽取

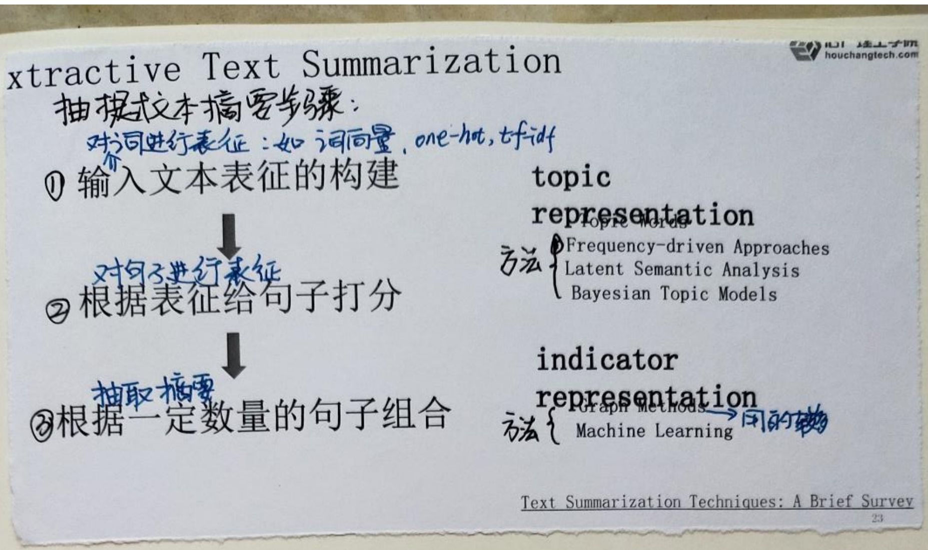
Summary: Peter and Elizabeth attend party city. Elizabeth rushed hospital.

Abstractive text summarization

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Elizabeth was hospitalized after attending a party with Peter.



xtractive Text Summarization

houchangtech.com

抽取式文本摘要步骤：

对词进行表征：如 词向量，one-hot，tf-idf

① 输入文本表征的构建

对句子进行表征

② 根据表征给句子打分

抽取摘要

③ 根据一定数量的句子组合

topic representation

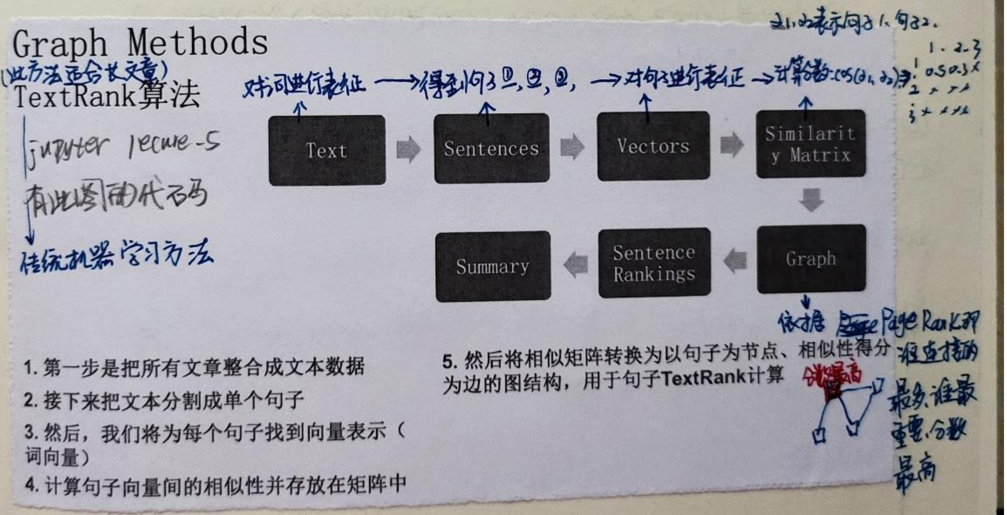
方法
- Topic Words
- Frequency-driven Approaches
- Latent Semantic Analysis
- Bayesian Topic Models

indicator representation

方法
- Graph Methods → 图的结构
- Machine Learning

Text Summarization Techniques: A Brief Survey

23



Graph Methods

(此方法适合长文章)

TextRank算法

jupyter lecture-5

有此图的代码

传统机器学习方法

对词进行表征 ⟶ 得到句子①，②，③， ⟶ 对句子进行表征 ⟶ 计算分数：cos(s1, s2)

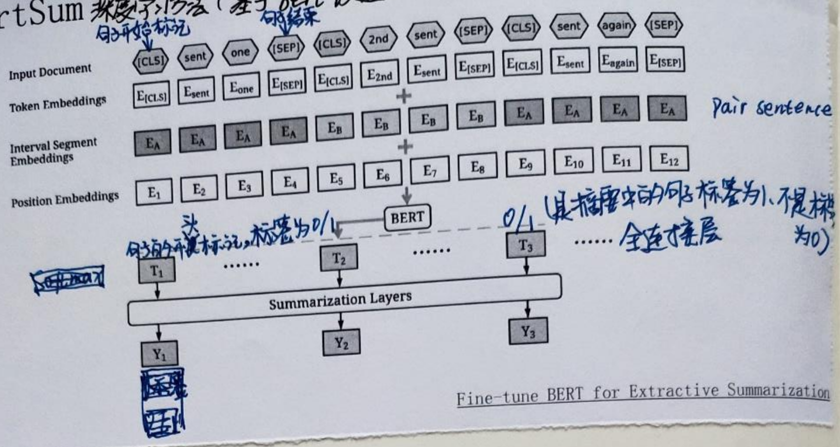
s1, s2表示句子1、句子2。

1. 2. 3
1 0.5 0.3 x
2 x x x
3 x x x x

依据 PageRank的 谁连接的最多，谁最重要，分数最高

1. 第一步是把所有文章整合成文本数据
2. 接下来把文本分割成单个句子
3. 然后，我们将为每个句子找到向量表示（词向量）
4. 计算句子向量间的相似性并存放在矩阵中
5. 然后将相似矩阵转换为以句子为节点、相似性得分为边的图结构，用于句子TextRank计算

BertSum 深度学习方法（基于bert改造的）


句子开始标记
句子结束
Input Document
[CLS] sent one [SEP] [CLS] 2nd sent [SEP] [CLS] sent again [SEP]
Token Embeddings
E[CLS] Esent Eone E[SEP] E[CLS] E2nd Esent E[SEP] E[CLS] Esent Eagain E[SEP]
Interval Segment Embeddings
EA EA EA EA EB EB EB EB EA EA EA EA
Pair sentence
Position Embeddings
E1 E2 E3 E4 E5 E6 E7 E8 E9 E10 E11 E12
BERT
句子的开头标记，标签为0/1
0/1（是摘要中的句子标签为1、不是标签为0）
全连接层
T1 T2 T3
Summarization Layers
Y1 Y2 Y3

Fine-tune BERT for Extractive Summarization

注：(1) bert是无监督学习

(2) bert是自编码结构

(3) bert是一个字一个字切分的（切分的单位为字 如 今/天/星/期/一）

(4) bert 模型缺点：模型大

bert在训练时，15%的词会被mask（掩盖）掉，其中（这15%中）的80%会完全被mask（类似完型填空），10%的给错误词 ~~[illegible]~~，10%的给正确词

bert就是很多层transformer的编码器，做相邻语句的判断任务（编码）实现无监督训练

给定上一个词，预测下一个词为自回归问题，bert为自编码问题

bert只有encoder，没有decoder.

Outline

- OOV 和Word-repetition解决
- Training Strategies
- 抽提式文本摘要基本方法
- 相关代码实践

附：

（1）transformer 模型 github：https://github.com/huggingface/transformers

（2）文章：BERT-Pre-training of Deep Bidirectional Transformers for Language Understanding：https://arxiv.org/pdf/1810.04805.pdf

（翻译博文：https://blog.csdn.net/sinat_33741547/article/details/86311310）

（3）bert 模型 github：https://github.com/jihun-hong/Bert-Classifier/tree/master/src/models