Machine translation

机器翻译

王彬沣、潘一辰、郭琪艺、田龙、许畅

目录

CONTENTS

01

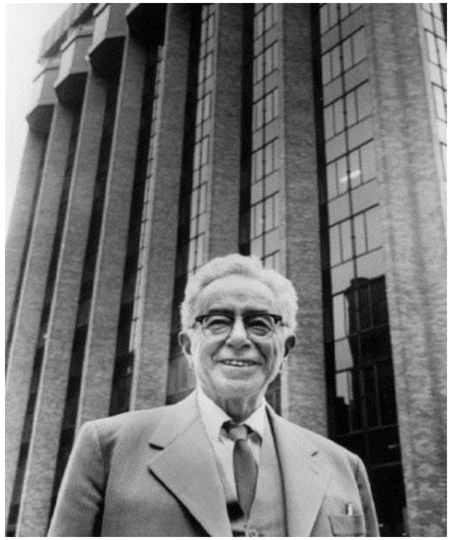
PART ONE

概机器翻译的历史发展

机器翻译的提出

提出者

瓦伦·韦弗(Warren Weaver) ,数学家，首次提出了分子生物学的概念，曾和香农一同撰写了《通信的数学原理》。



提出过程：

1. 1946年，韦弗受密码学和计算机启发，产生机器翻译想法
2. 1947年与控制论之父维纳探讨，遭到反对
3. 1948年，与Booth讨论。
4. 1949年，在《翻译》备忘录中提出了机器翻译

几个重要的事件

事件一

1954，Georgetown University和IBM；
250条俄语词汇， 6 条语法规则，将60个俄语词组翻译成了英语。

事件二

1964，ALPAC建立；
否定机器翻译，语义障碍严重

事件三

1990 ， IBM ， “ A Statistical Approach to MT”，5个word-based翻译系统

事件四

2013， 2014， encoder-decoder 与 seq2seq

理性主义

A 基于规则——RBMT

经验主义

A 基于实例——EBMT

B 基于统计——SMT

C 基于神经网络——NMT

基于规则的机器翻译

rule based machine translation

直接翻译法

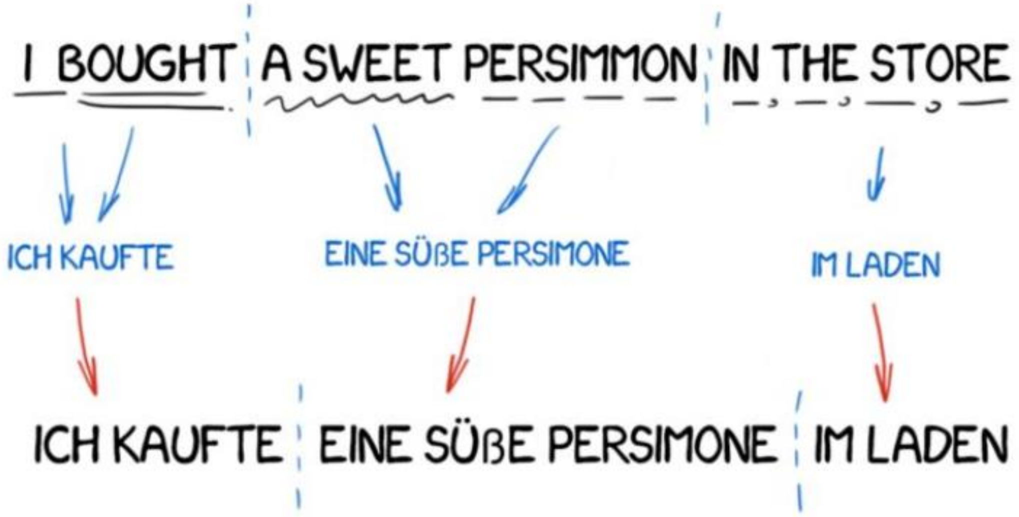
过程：源语输入、双语词典查询、词序调整、译语输出
效果：简单直接，但不够通顺流畅。

基于规则的机器翻译

rule based machine translation

转换法

过程：源语分析、源语转换、译语生成
效果：词序更合理，翻译更复杂。



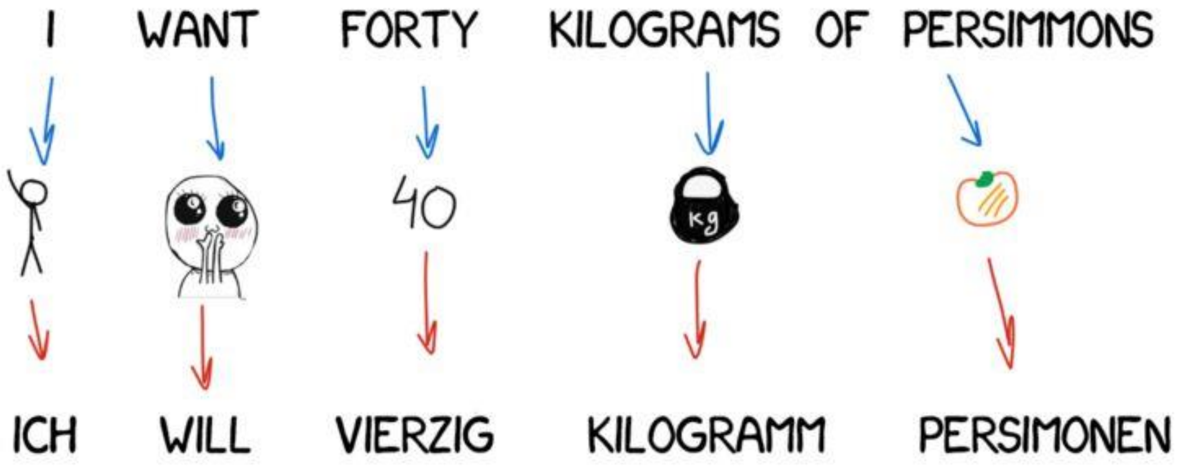
基于规则的机器翻译

rule based machine translation

中间语法

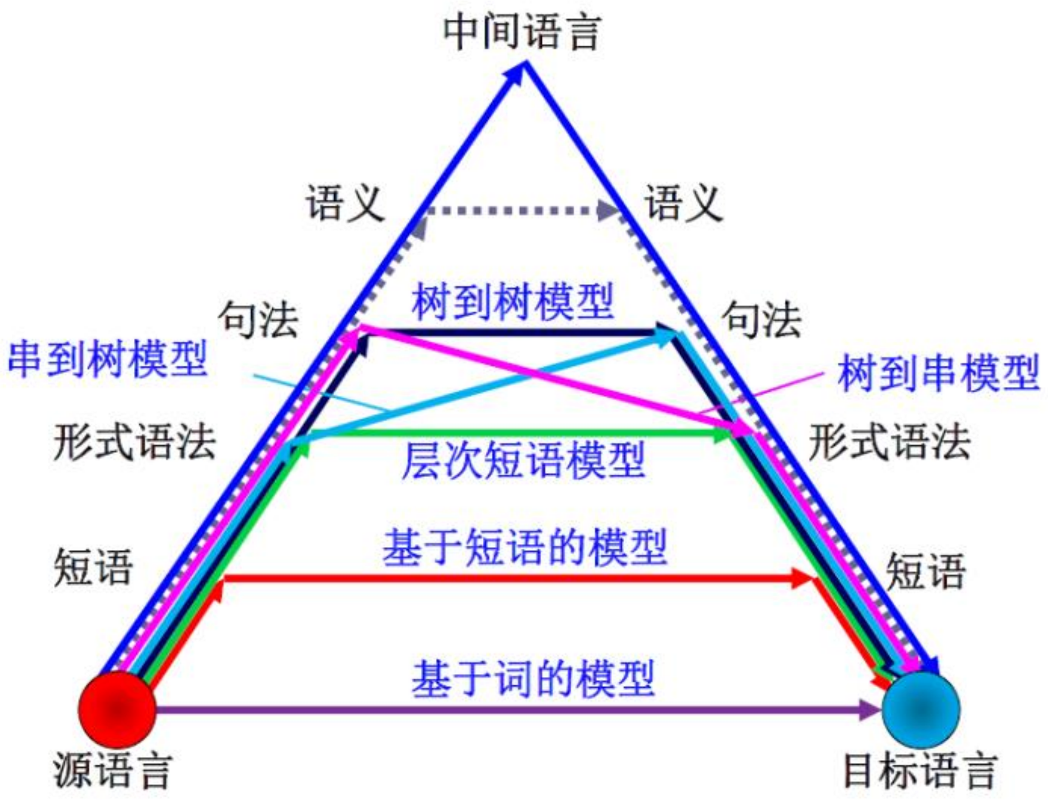
过程：把源语转换成一种无歧义、对任何语言都通用的中间语言表示，然后借助该中间语言表示生成译语

效果：理论上可行，但无成功案例。



基于规则的机器翻译

rule based machine translation



基于实例的机器翻译

example based machine translation

01 / 提出者

1984，京都大学长尾真

02 / 方法

实例泛化，将一类语言现象用统一的模板来表示。翻译时对已有的模板进行检索

03 / 例子

翻译实例：

How much is that **red umbrella**?	Ano **akai kasa** wa ikura desu ka.
How much is that **small camera**?	Ano **chiisai kamera** wa ikura desu ka.

提取到的模板：

模板1：
How much is that X == Ano X wa ikura desu ka
模板2：
Red umbrella == Akai kasa
模板3：
Small camera ==Chiisai kamera

02

PART TWO

统计机器翻译

1990年代初IBM首次开展统计机器翻译研究

1999年JHU夏季研讨班复现IBM的工作并推出开放源代码的工具

2001年IBM提出机器翻译自动评测方法BLEU

2002年NIST开始举行每年一度的机器翻译评测

2002年Franz Josef Och提出统计机器翻译的对数线性模型

发展历程

发展历程

2003

Franz Josef Och提出对数线性模型的最小错误率训练方法

2004

Philipp Koehn推出Pharaoh(法老)

2005

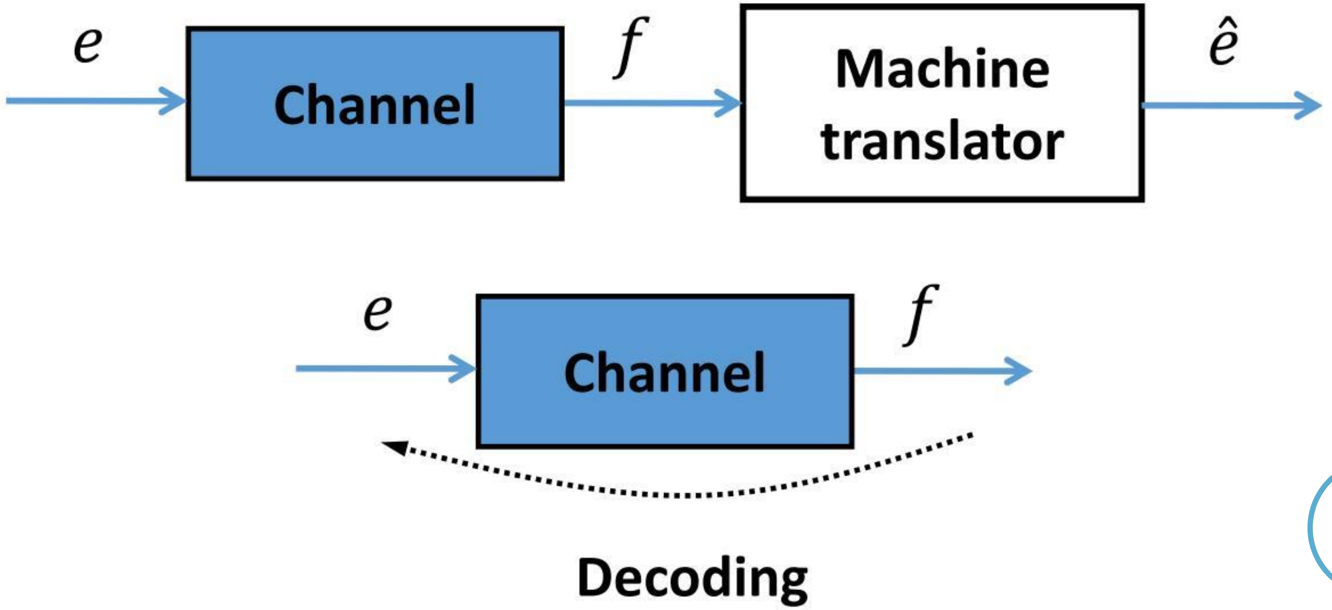
Google在NIST评测中大获全胜

2006

NIST评测中USC-ISI的树到串句法模型第一次超过Google

参考《机器翻译原理与方法》课程PPT P59

噪声信道模型



核心思想

$$e^{*} = \arg\max_{e} P(e)P(f \mid e)$$

解码算法

语言模型

翻译模型

$$P(e|f) = P_{\lambda_1,\dots\lambda m}(e|f) = \frac{\exp[\sum_{i=1}^{m}\lambda_i h_i(e,f)]}{\sum_{e'}\exp[\sum_{i=1}^{m}\lambda_i h_i(e',f)]}$$

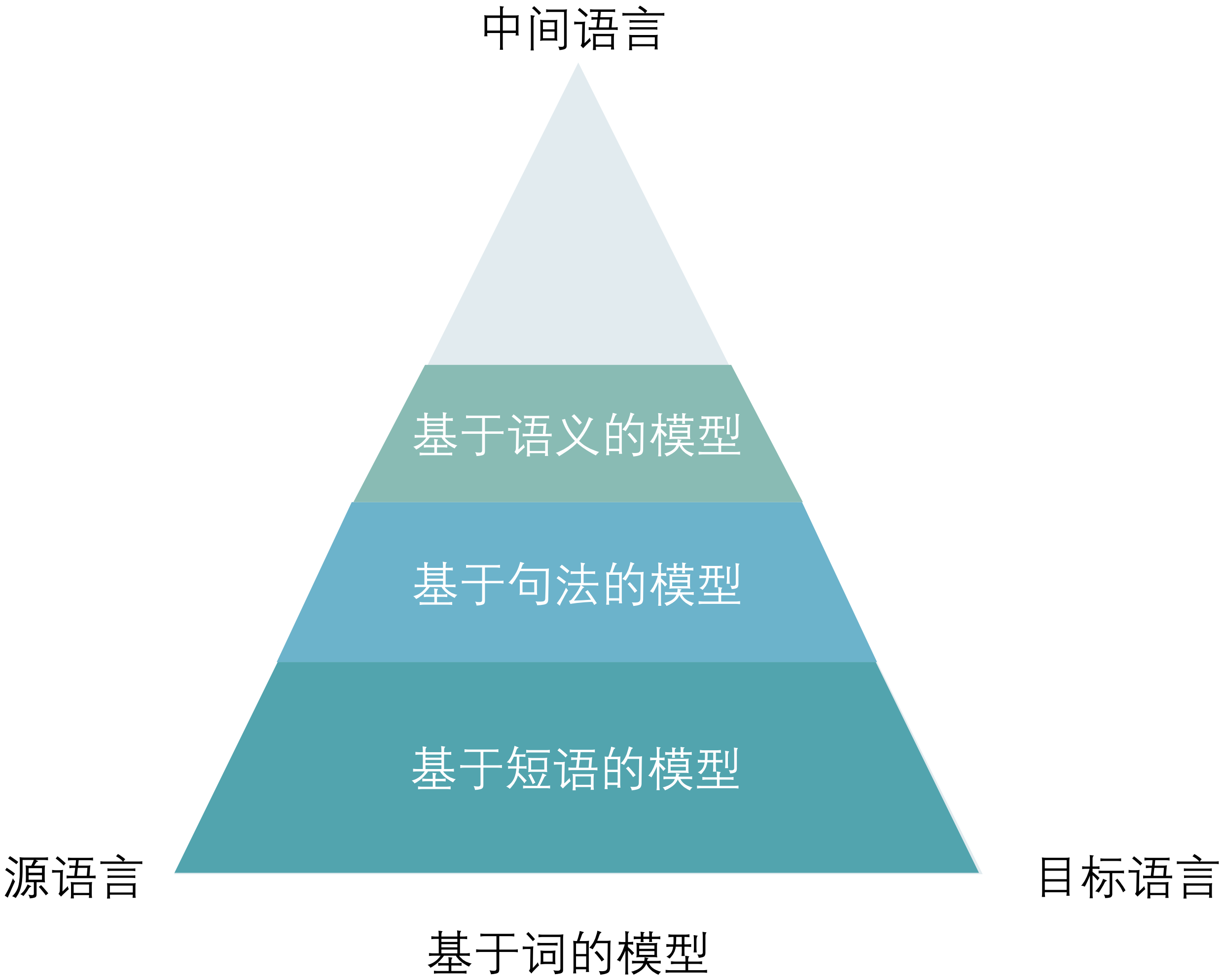
$$e^* = \arg\max_e P(e|f) = \arg\max_e \sum_{i=1}^{m}\lambda_i h_i(e,f)$$

核心思想——特征

1. 句子长度特征
2. 附件的语言模型特征
3. 词典特征

对数线性模型

刘群.统计机器翻译综述[J].中文信息学报,2003(04):1-12.



统计翻译模型的发展

基于短语的模型是最为成熟的模型，而基于句法的模型是一个研究热点。在这个金字塔上，越往塔尖的方向走，对语言的分析也越深入。

金字塔

基于词的统计翻译模型

（1） 根据双语平行语料库，在无人工参与的情况下确定词语对齐；

（2） 形成带概率的翻译词典；

（3） 通过已知的概率（词语翻译和调序）计算两个句子互为翻译的概率。

Model 1

仅考虑词与词的互译概率，翻译效果与次序无关

Model 2

增加了词的位置变化的概率

Model 3

增加繁衍率模型，考虑一个单词可以翻译为多个单词的情况

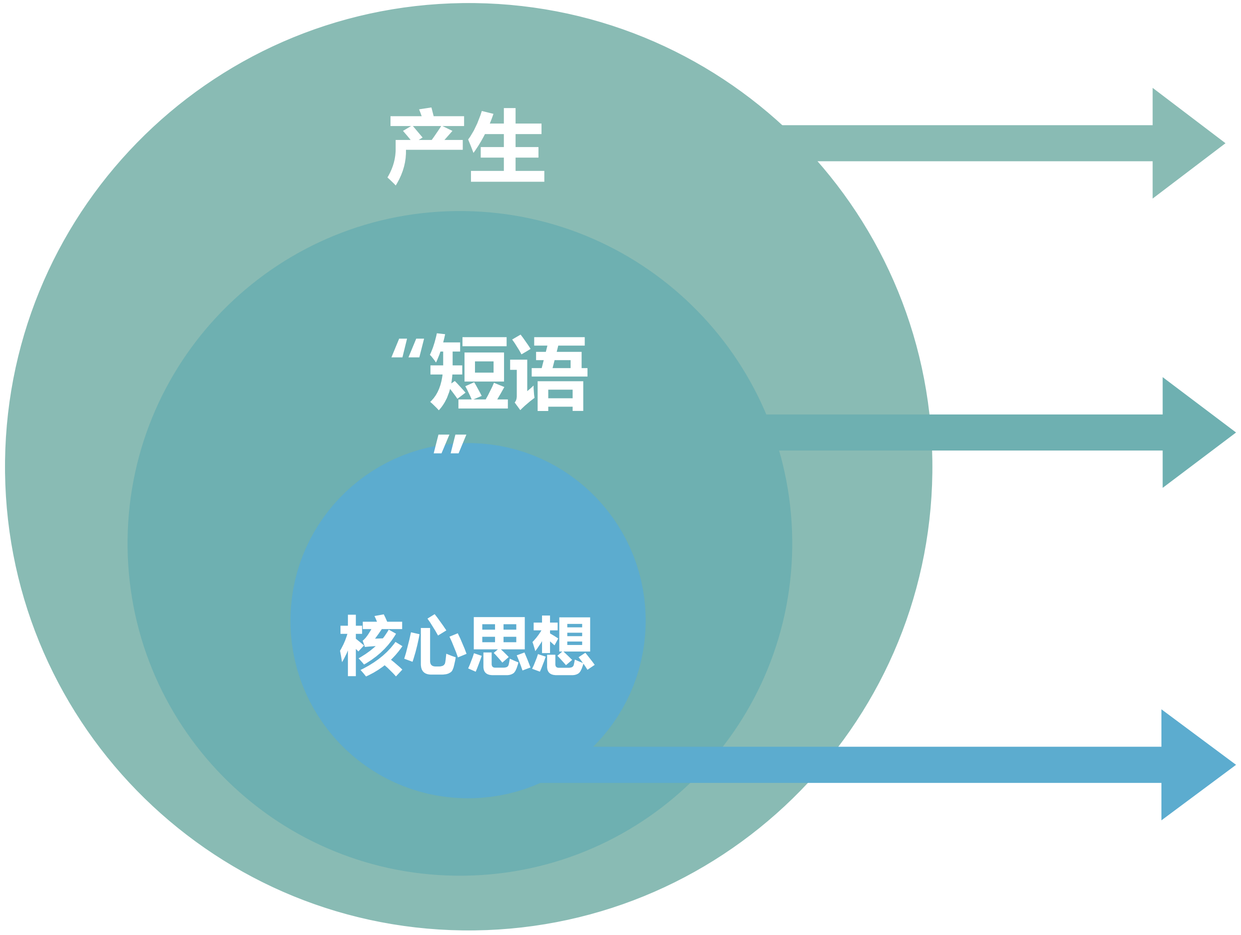
Model 4

增加相对对齐模型

Model 5

修正Model 4中的缺陷

基于短语的统计翻译模型



基于词的SMT：没有采用任何句法结构信息，词序的调整是完全盲目的

比传统语言学中“短语”的概念更为宽泛，可以称之为“词序列”，即一串连续的词的组合

在基于词对齐的语料库中，寻找并记录互为翻译的双语短语并统计概率，形成一个双语的短语库。

基于句法的统计翻译模型

形式化语法

语言学语法

基于短语结构树

侧重对句子各部分及整体结构的描述，更多体现了对句法结构的关照

基于依存树

侧重对句子内部词与词之间关系的描述，更体现了对语义结构的关照

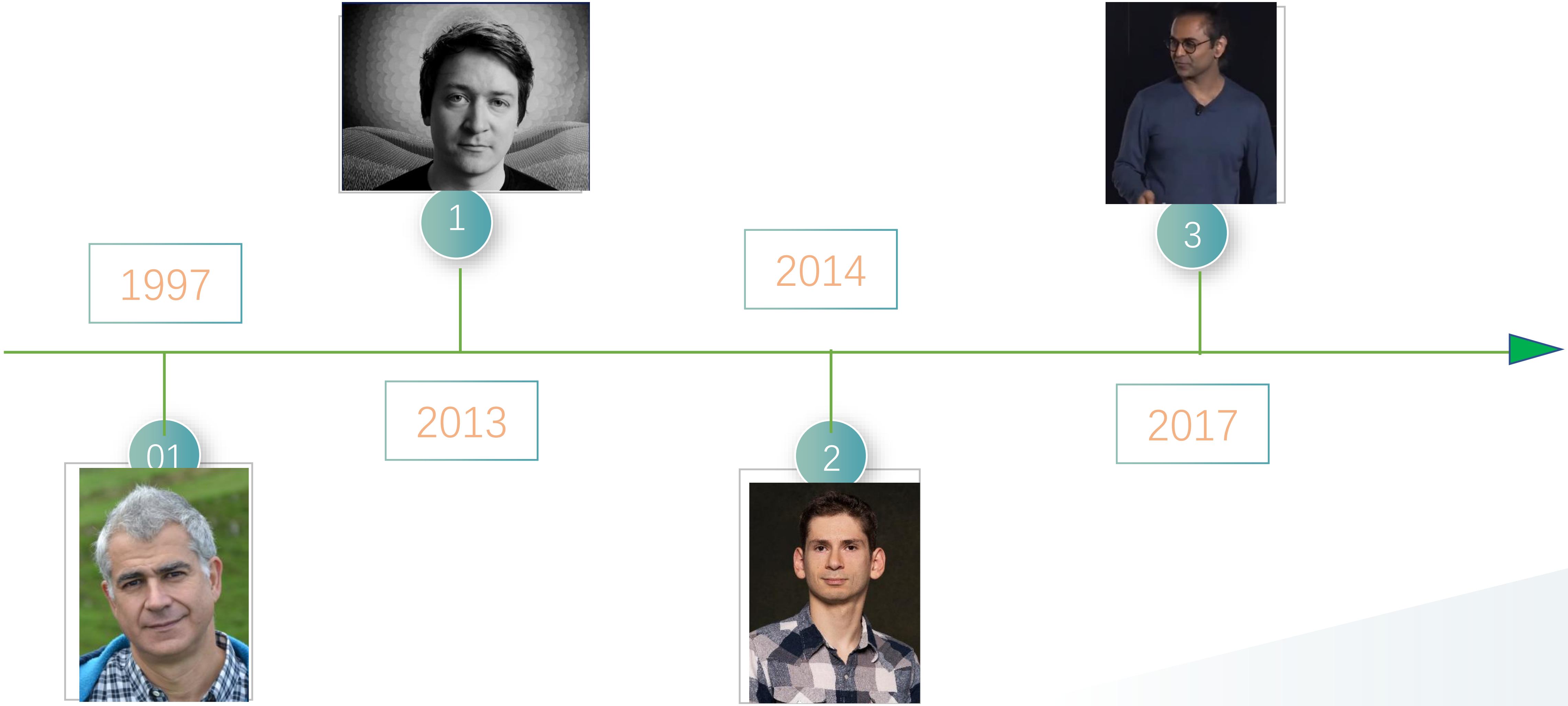
高璐璐,赵雯.机器翻译研究综述[J].中国外语,2020,17(06):97-103.DOI:10.13564/j.cnki.issn.1672-9382.2020.06.015.

03

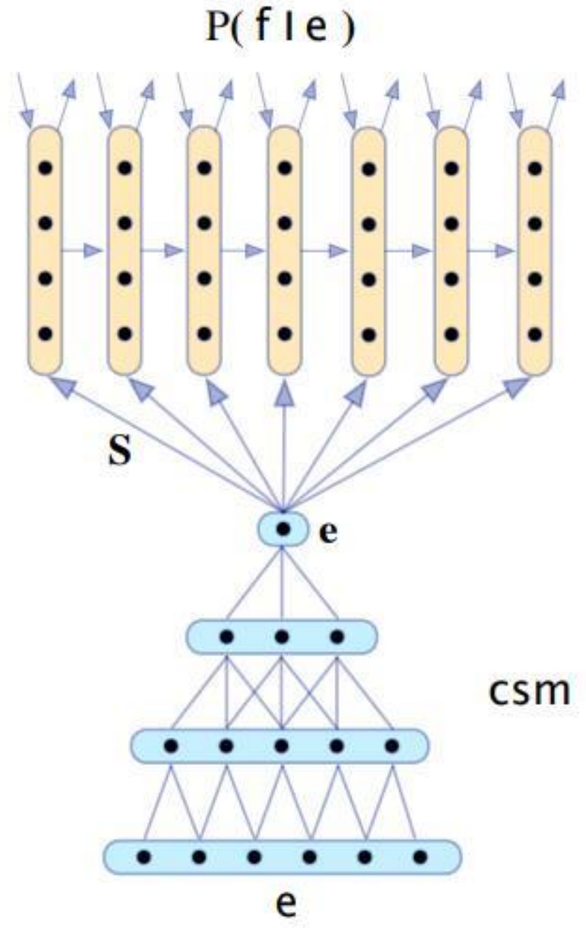
PART THREE

神经机器翻译

A brief history of NMT



Recurrent Continuous Translation Models



RCTM I

WMT-NT	*2009*	*2010*	*2011*	*2012*
KN-5	218	213	222	225
RLM	178	169	178	181
IBM 1	207	200	188	197
FA-IBM 2	153	146	135	144
RCTM I	143	134	140	142
RCTM II	**86**	**77**	**76**	**77**

Table 1: Perplexity results on the WMT-NT sets.

WMT-NT	*2009*	*2010*	*2011*	*2012*
RCTM I + WP	19.7	21.1	22.5	21.5
RCTM II + WP	19.8	21.1	22.5	21.7
`cdec` (12 features)	19.9	21.2	22.6	21.8

Table 4: Bleu scores on the WMT-NT sets of each RCTM linearly interpolated with a word penalty WP. The `cdec` system includes WP as well as five translation models and two language modelling features, among others.

Kalchbrenner, Nal, and Phil Blunsom. "Recurrent continuous translation models." *Proceedings of the 2013 conference on empirical methods in natural languag* *processing*. 2013.

NMT with Alignment

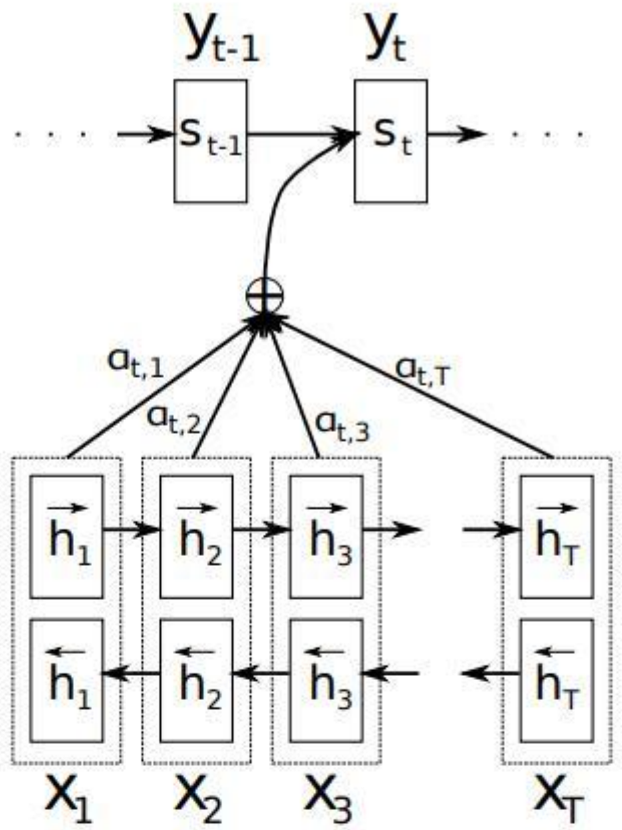


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

$$p(y_i | y_1, \ldots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i),$$

$$s_i = f(s_{i-1}, y_{i-1}, c_i).$$

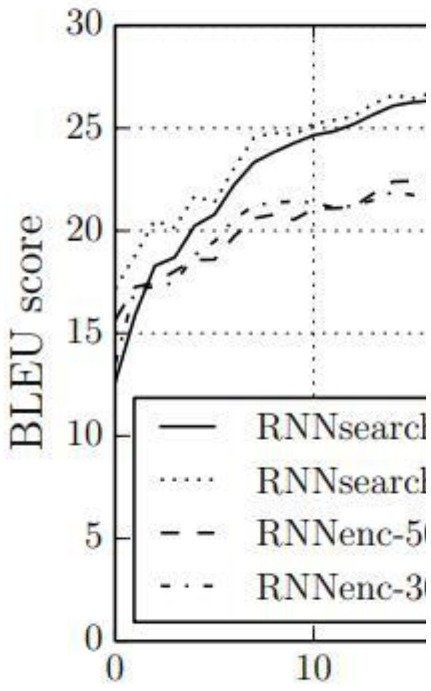
$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

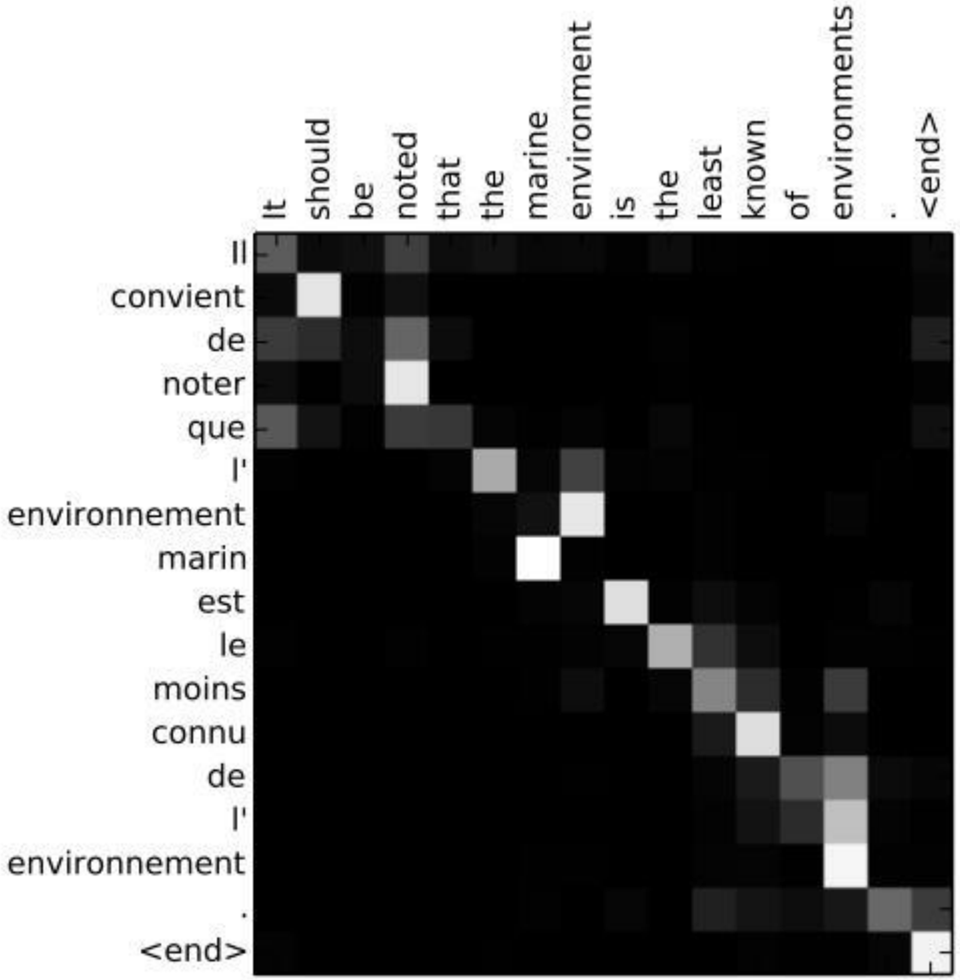
$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

$$e_{ij} = a(s_{i-1}, h_j)$$

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473* (2014).

NMT with Alignment

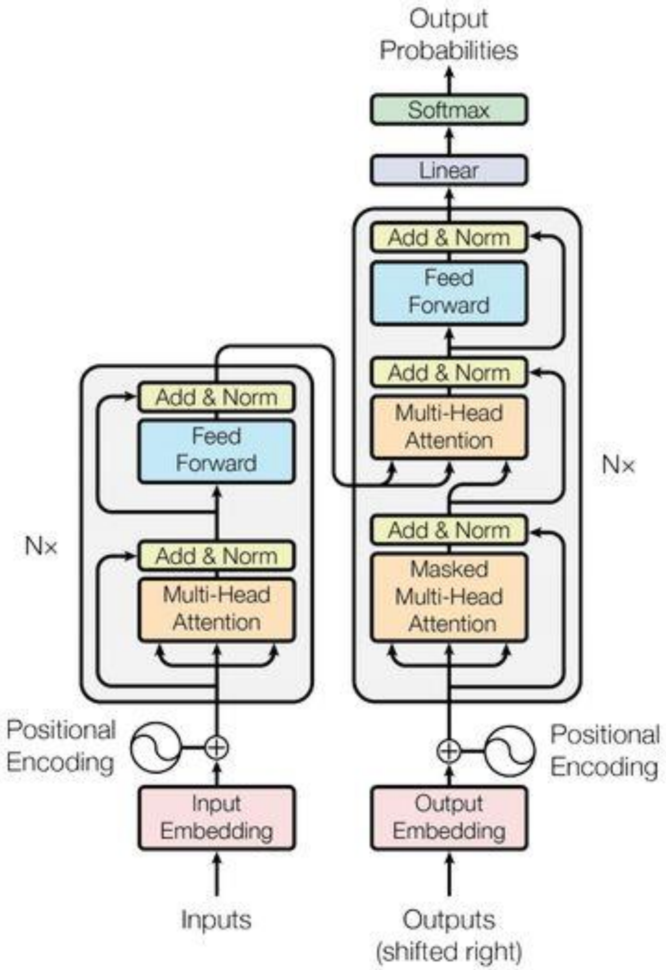




(b)

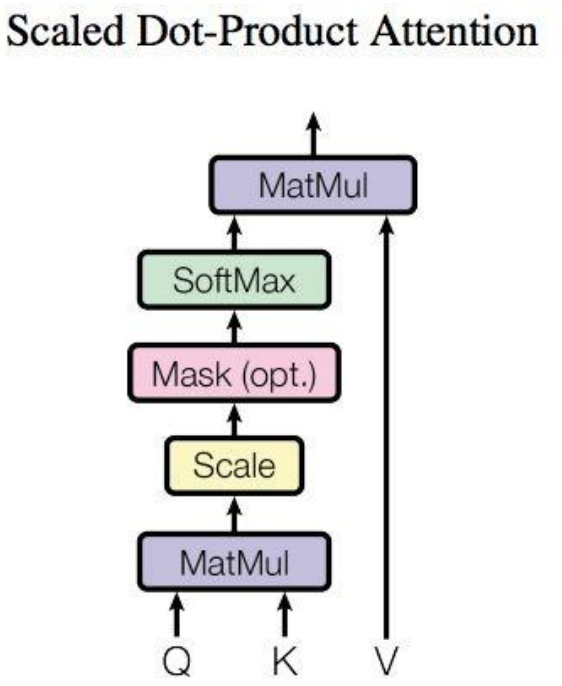
2: The BLEU scores
generated translations
test set with respect
lengths of the sen-
The results are on
test set which in-
sentences having un-
words to the models.

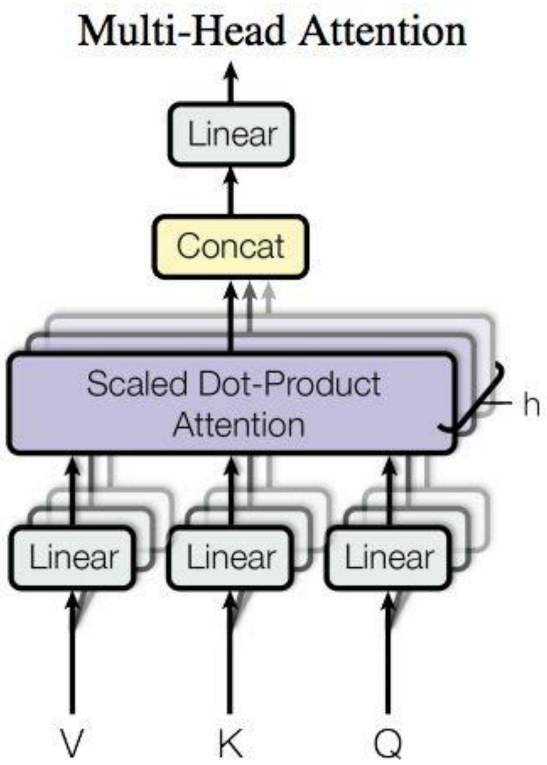
Transformer



Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

Attention


Scaled Dot-Product Attention
MatMul
SoftMax
Mask (opt.)
Scale
MatMul
Q
K
V


Multi-Head Attention
Linear
Concat
Scaled Dot-Product Attention
h
Linear
Linear
Linear
V
K
Q

Oracle Method

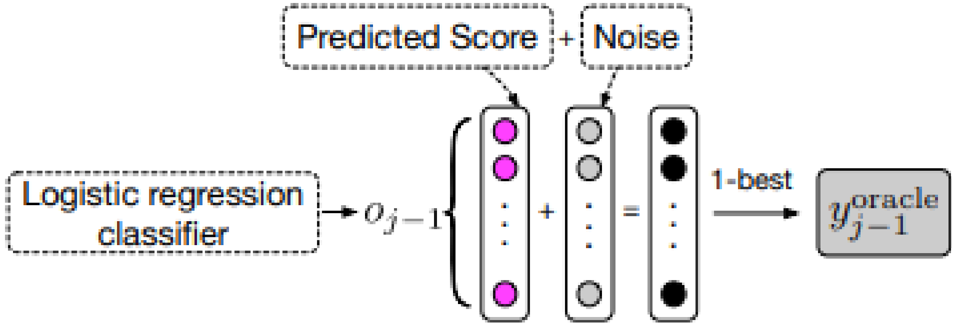
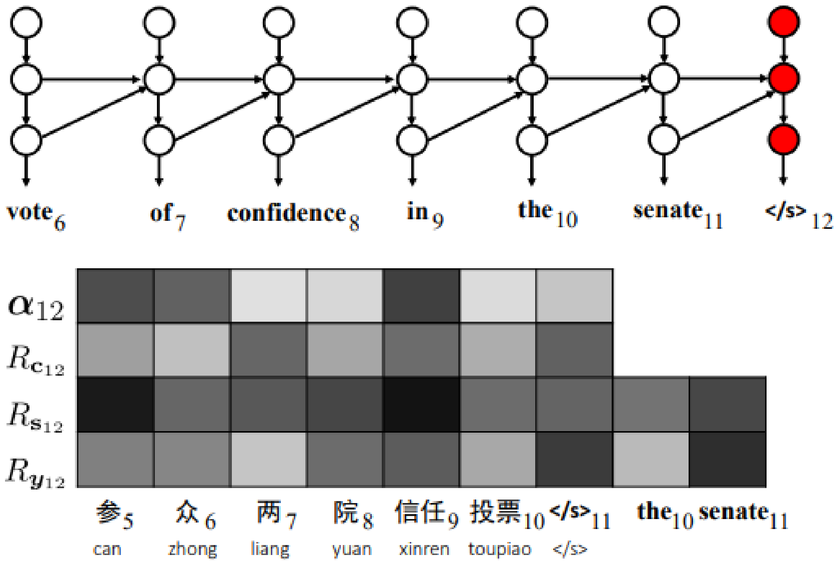


Figure 3: Word-level oracle with Gumbel noise.

Systems	Architecture	MT03	MT04	MT05	MT06	Average
Existing end-to-end NMT systems						
Tu et al. (2016)	Coverage	33.69	38.05	35.01	34.83	35.40
Shen et al. (2016)	MRT	37.41	39.87	37.45	36.80	37.88
Zhang et al. (2017)	Distortion	37.93	40.40	36.81	35.77	37.73
Our end-to-end NMT systems						
this work	RNNsearch	37.93	40.53	36.65	35.80	37.73
	+ SS-NMT	38.82	41.68	37.28	37.98	38.94
	+ MIXER	38.70	40.81	37.59	38.38	38.87
	+ OR-NMT	**40.40**‡†*	**42.63**‡†*	**38.87**‡†*	**38.44**‡	**40.09**
	Transformer	46.89	47.88	47.40	46.66	47.21
	+ word oracle	47.42	48.34	47.89	47.34	47.75
	+ sentence oracle	**48.31***	**49.40***	**48.72***	**48.45***	**48.72**

Zhang, Wen, et al. "Bridging the gap between training and inference for neural machine translation." *arXiv preprint arXiv:1906.02448* (2019).

Visualizing and Understanding NMT



Input: A neural network G for a sentence pair and a set of hidden states to be visualized $\mathcal{V}$.
Output: Vector-level relevance set $\mathcal{R}$.

for $u \in G$ *in a forward topological order* **do**
  for $v \in \mathrm{OUT}(u)$ **do**
    calculating weight ratios $w_{u \to v}$;
  end
end
for $\mathbf{v} \in \mathcal{V}$ **do**
  for $v \in \mathbf{v}$ **do**
    $r_{v \leftarrow v} = v$; // initializing neuron-level relevance
  end
  for $u \in G$ *in a backward topological order* **do**
    $r_{u \leftarrow v} = \sum_{z \in \mathrm{OUT}(u)} w_{u \to z} r_{z \leftarrow v}$; // calculating neuron-level relevance
  end
  for $\mathbf{u} \in \mathcal{C}(\mathbf{v})$ **do**
    $R_{\mathbf{u} \leftarrow \mathbf{v}} = \sum_{u \in \mathbf{u}} \sum_{v \in \mathbf{v}} r_{u \leftarrow v}$; // calculating vector-level relevance
    $\mathcal{R} = \mathcal{R} \cup \{R_{\mathbf{u} \leftarrow \mathbf{v}}\}$; // Update vector-level relevance set
  end
end

Algorithm 1: Layer-wise relevance propagation for neural machine translation.

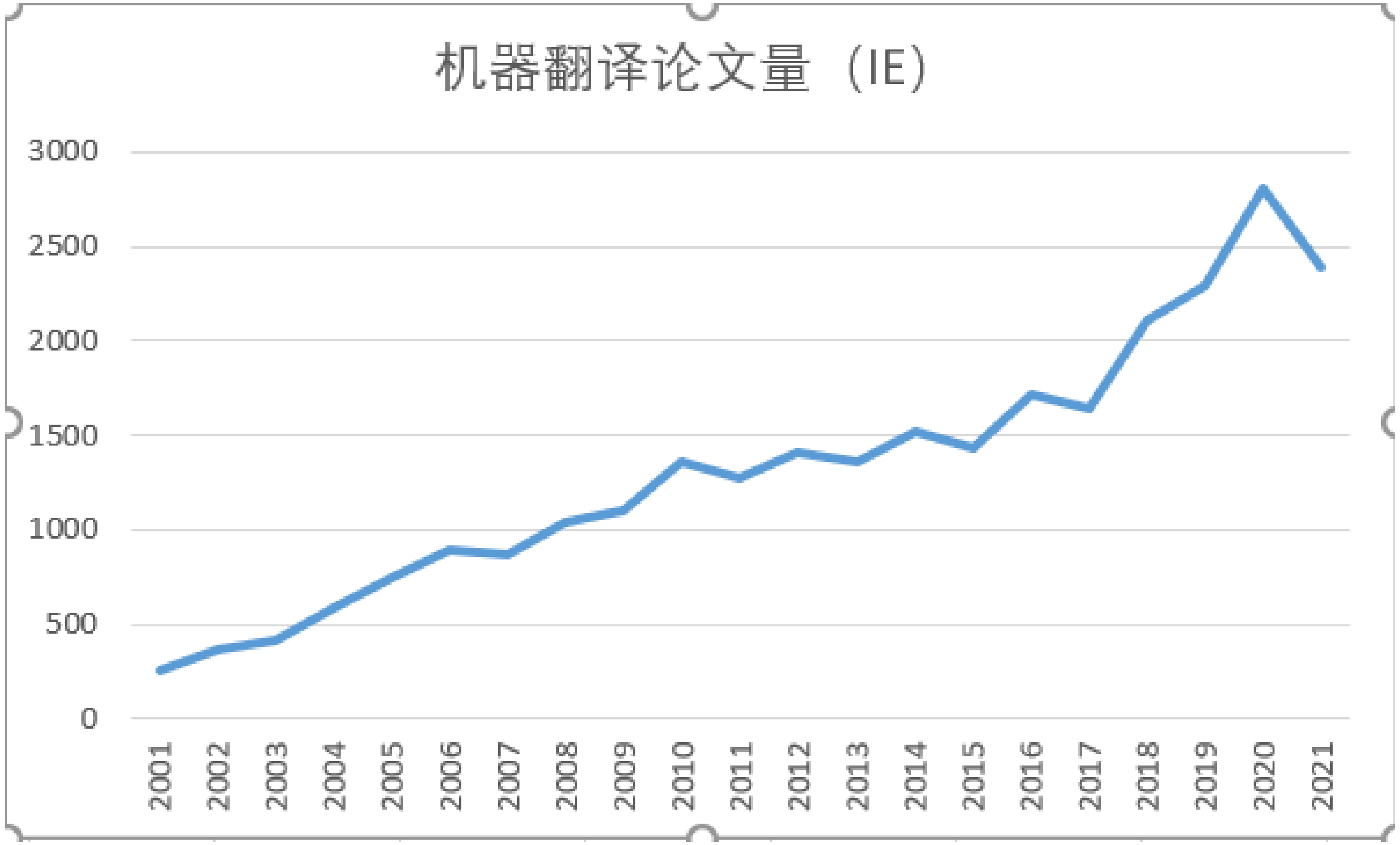
Ding, Yanzhuo, et al. "Visualizing and understanding neural machine translation." *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2017.

04

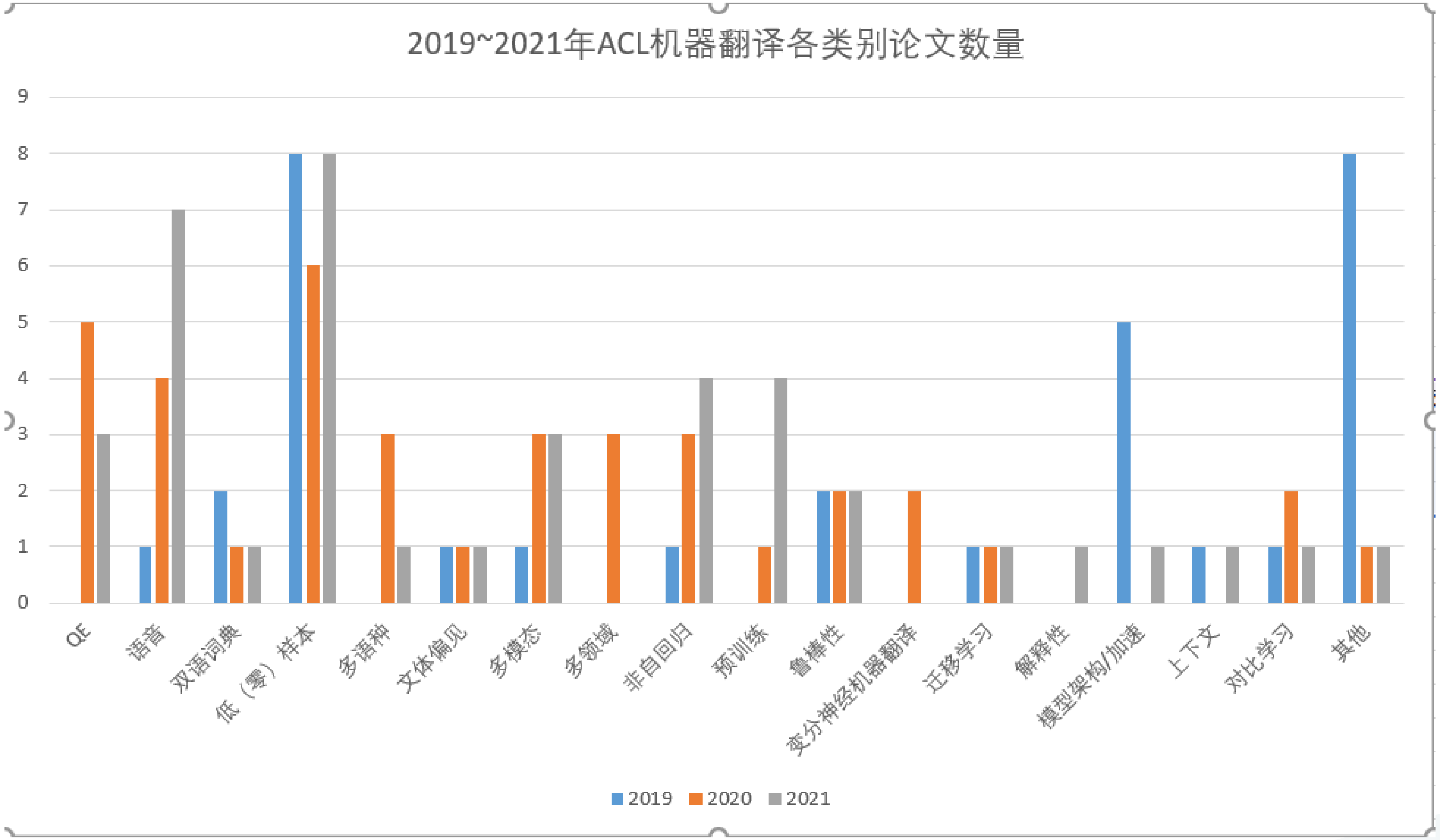
PART FOUR

前沿进展

机器翻译的发展趋势


机器翻译论文量（IE）
3000
2500
2000
1500
1000
500
0
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021

机器翻译的各类别论文发展趋势



(神经)机器翻译的主要挑战

01 / 平行语料的匮乏

神经机器翻译往往依赖于大规模的平行语料进行训练。现实世界中，除了部分富资源语言（如英语，汉语，德语，俄语，印地语等），更多的语言本身受众较小，缺乏海量的双语平行语料进行监督学习。

02 / 参数量大，训练、运行慢

过参数化的（超大规模）模型能够有效提升神经机器翻译的性能，但是庞大的存储开销和高昂的计算复杂度使得这类模型无法直接部署到边缘设备(如手机，翻译笔，离线翻译机等)上。

03 / 翻译结果的鲁棒性

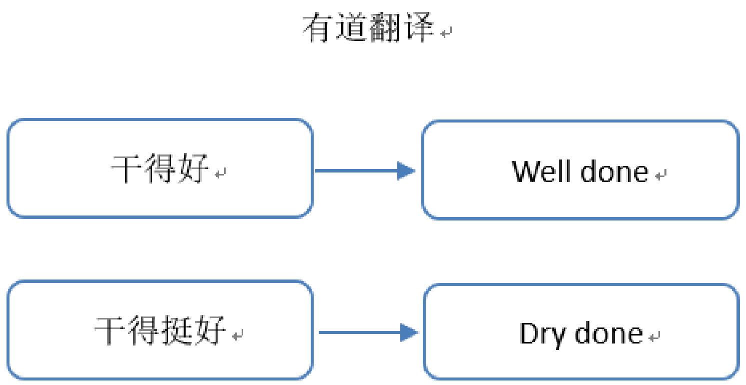
NMT 模型尽管取得了巨大的成功，但它对输入中存在的微小干扰仍然很敏感，这就会导致它出现各种不同的错误，如翻译不足、翻译过度或翻译错误。

04 / 先验知识与NMT的结合

神经机器翻译的成功依赖于大量参数和数据的堆叠，而忽视了先验语法知识，如何将先验知识与NMT结合是研究人员迫切想解决的一个问题。

05 / NMT的可解释性

大多数的神经机器翻译都是基于attention机制的encoder-decoder模型，然而这种模型在内部传递的是浮点数，类似于“黑箱”，难以理解和调试。



低资源机器翻译

问题的产生：缺乏大量的平行语料数据

01 / 生成“伪数据”

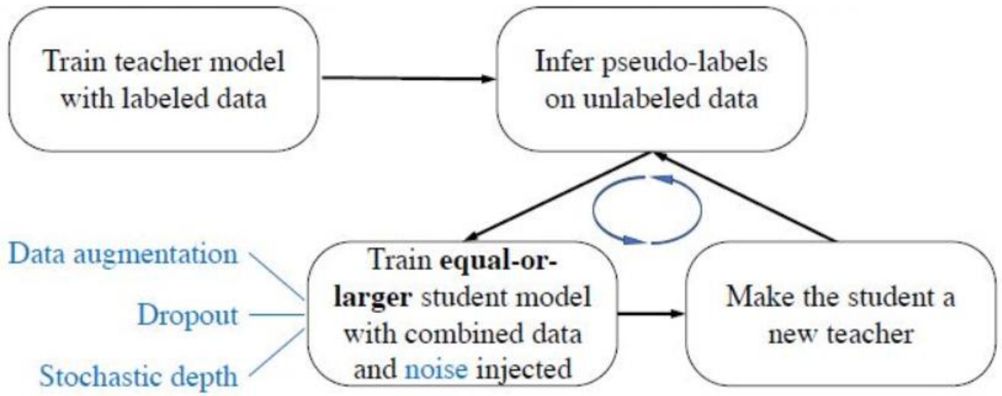
Back-translation

训练“目标语->源语“网络来生成伪数据

Self-training

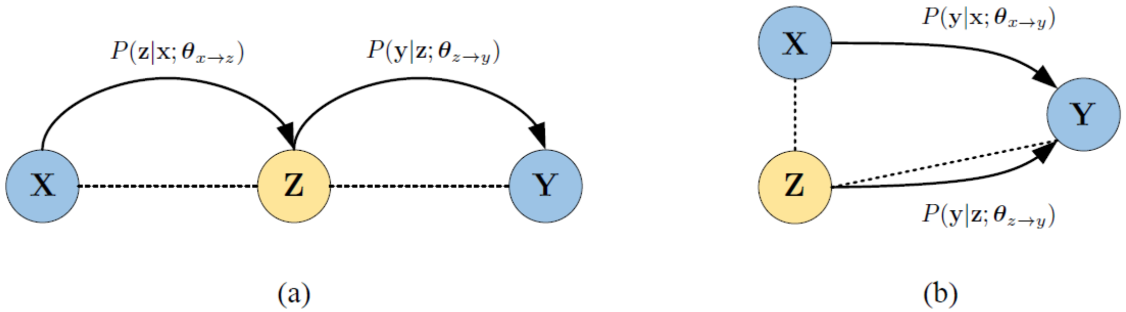
训练“源语->目标语”网络来生成伪数据

QE（翻译质量评估系统）



02 / 枢轴语言

先训练一个“源语言->枢轴语言”的网络，在以“枢轴语言->目标语音”为知识知道网络进一步训练，成为“源语言->目标语言”网络



A Teacher-Student Framework for Zero-Resource Neural Machine Translation

模型加速：轻量模型

轻量模型

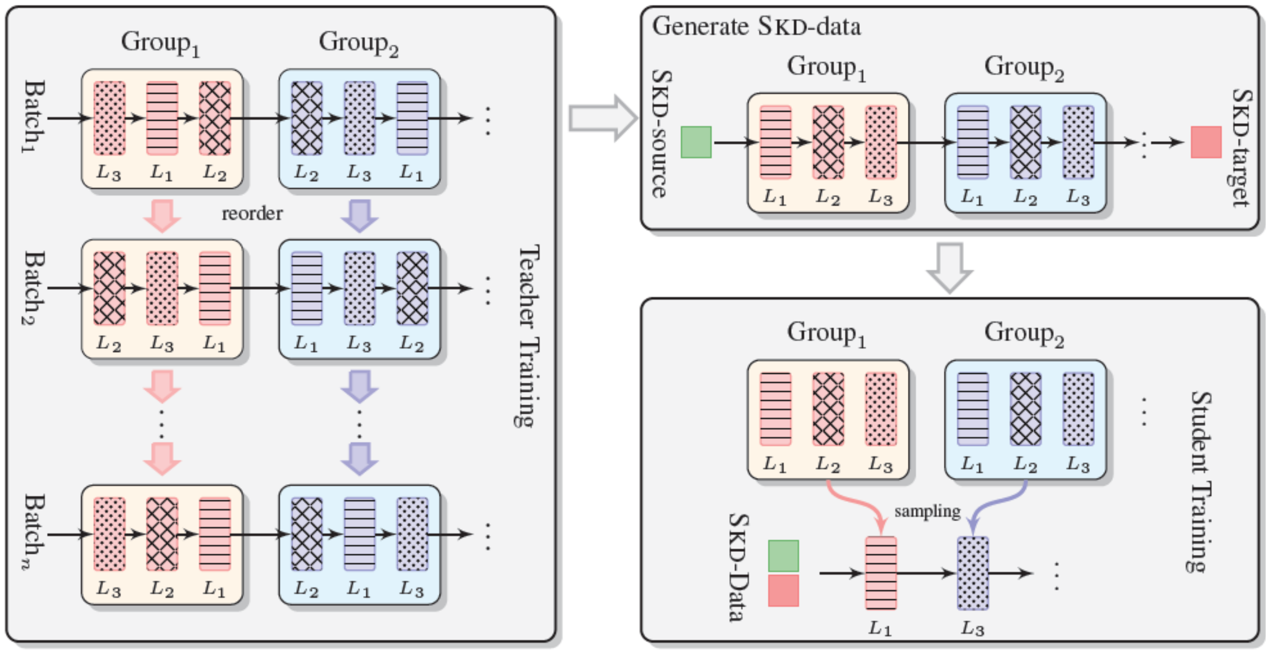
用比较小的参数量的模型来代替大模型

最新实例：GPKD模型（群体知识蒸馏）

用一层表示一组网络

用一组网络中的一层作为初始参数

使用生成的SKD数据进一步训练



Learning Light-Weight Translation Models from Deep Transformer

模型加速：非自回归编码（NAT）

NAT与AT(自回归编码)的差异

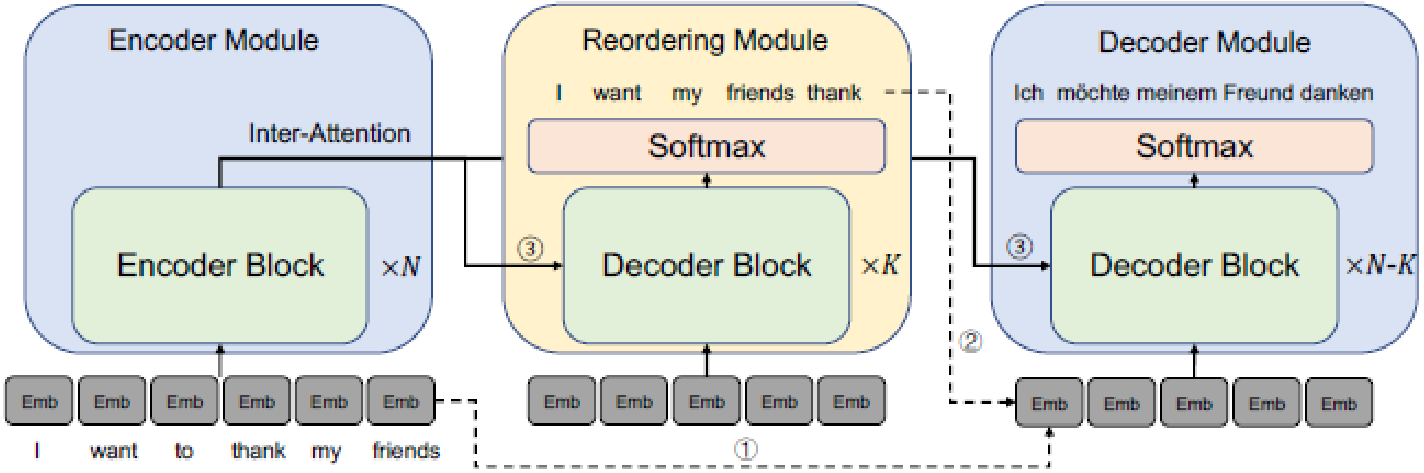
NAT并行翻译句子的所有词，速度更快，但翻译质量较差

AT每次使用已生成的序列作为已知信息预测未来的一个单词，效果好，但速度慢

最新实例：ReorderNAT

加入重排序模块

先进行重排序（把源语言的顺序调整为目标语言顺序）的“伪翻译”



Guiding Non-Autoregressive Neural Machine Translation Decoding with Reordering Information

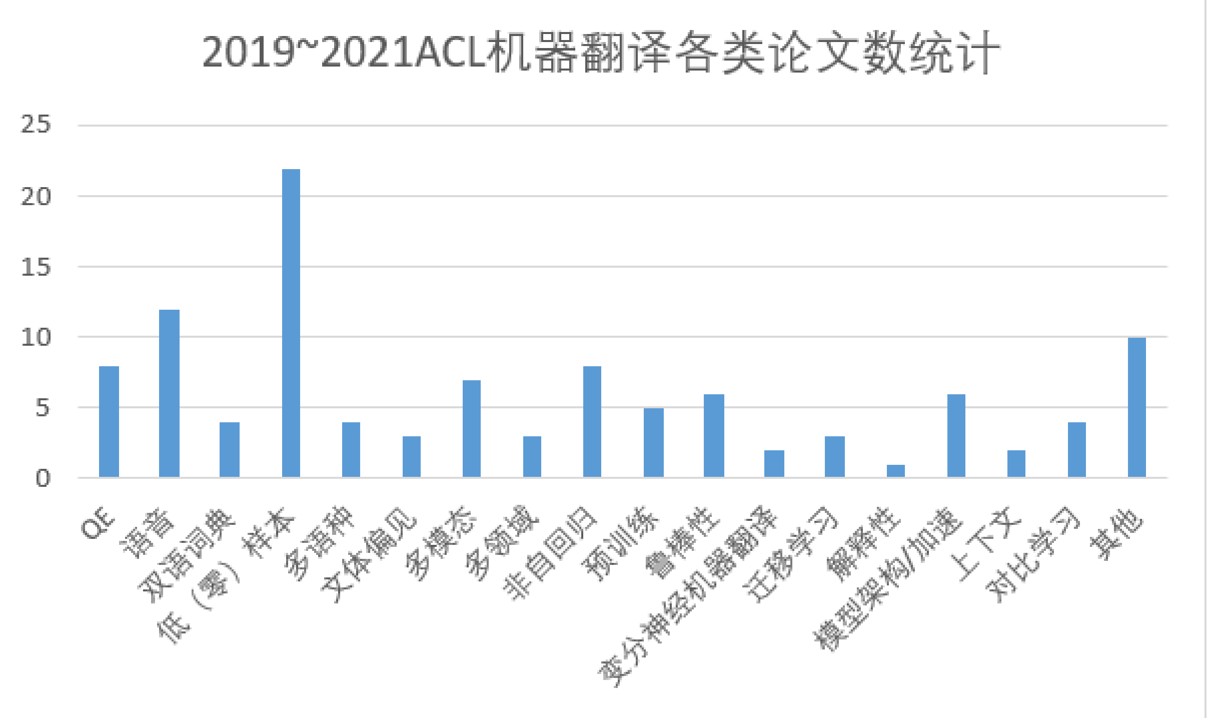
机器翻译的新热门应用

01 / 语音翻译/多模态

翻译的内容不再局限于文本，还有语言、图片、视频等，常见的方法主要有级联系统、端到端系统。

02 / 多语言机器翻译

主要为一对多、多对多的翻译
个人感觉最主要的是共享参数、语义信息



语音翻译

01

级联系统

识别-翻译-生成

速度慢，错误积累

02

端到端系统

直接由语言到目标语言的文本或语音

训练数据少，需要更多特征

03

Wait-k

每个词等待k个词后就进行翻译，无需等待整句话，加快解码

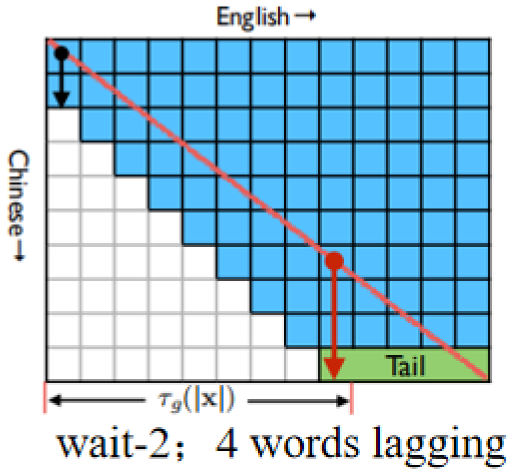
Future-guided Training（新进展）

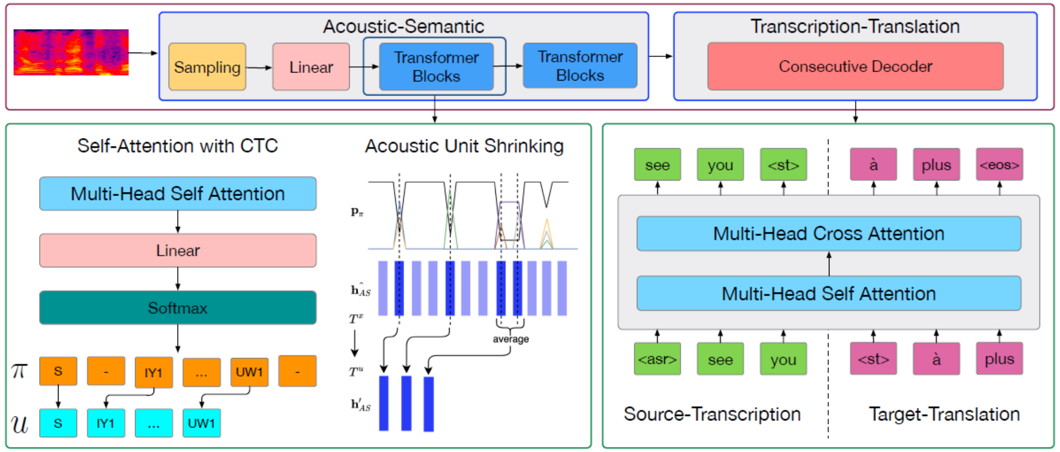
减少重复编码的时间

04

COSTT

把级联系统与端到端系统进行了结合





Future-Guided Incremental Transformer for Simultaneous Translation

Consecutive Decoding for Speech-to-text Translation

One-to-many多语言翻译

传统方法

一个encoder，多个decoder

解码过程不能充分利用翻译模型的信息

解码效率较低

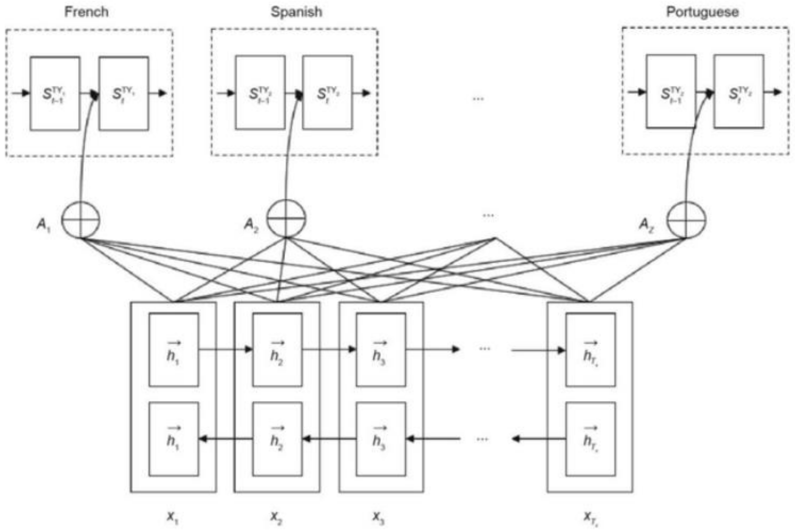


图5. one-to-many多任务学习的通用框架

SimNMT

同步交叉交互解码器

可依赖未来信息

可依赖其他目标语言的上下文信息

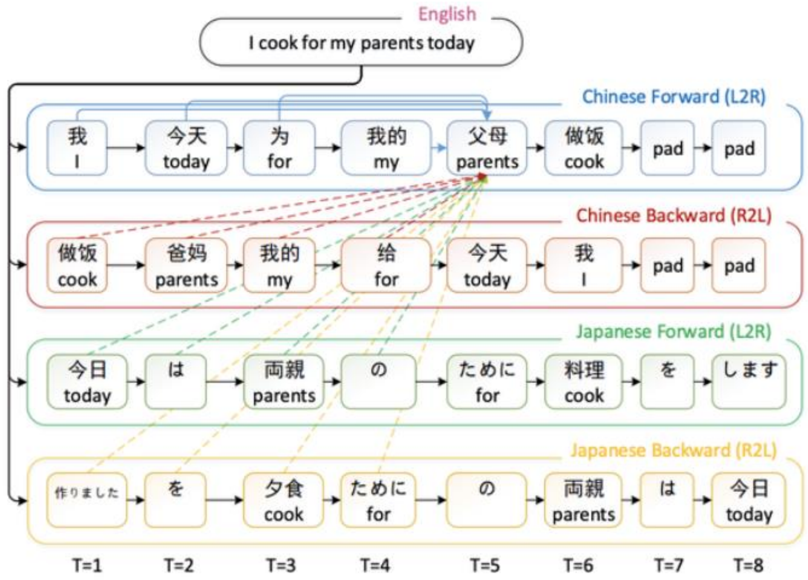


图6. 利用不同解码信息的多语言交互式解码框架[5]

Synchronous Interactive Decoding for Multilingual Neural Machine Translation

05

PART FOUR

Demo实现

About Dataset

Tatoeba

infomation from Wikipedia

Tatoeba(tatoeba.org)是一个自由的在线数据库，收集面向外语学习者的例句。Tatoeba一词来源于日语，意思是“举个例子”。与在线辞典网站不同，Tatoeba关注的是句子及其语法结构和翻译。

数据量

62922对中英语句对

数据分布

简单的日常用语，jieba分词后得到中文词24968个 英文词15318个 (包含标点符号)

数据样例

```
我十分理解你的立场。
I understand your position perfectly.
以这种方式钱是重要的。
Money is important in this way.
我知道你在学校学法语。
I know that you are learning French at school.
我认为汤姆一点教学经验都没有。
I don't think Tom has any teaching experience.
买你想要的。
Pay what you want.
我们没有冰激凌。
We don't have any ice.
汤姆现在跟换了个人一样。
Tom is like a different person now.
我依旧相信爱情。
I still believe in love.
我开始记起来了。
I am beginning to remember it.
推销员经常是语速快的人。
Salesmen are usually fast talkers.
```

Data can be downloaded from https://tatoeba.org/zh-cn/downloads

Seq2Seq model（Cell is GRU, with attention）

预处理

为了降低训练难度，方便模型收敛，将数据集中 中文样本分词之后超过十个词的语句对舍弃，余下30065个语句对。

同时将低频词（我设定为出现次数小于3的词）舍去，用<unk>替换，构建词典，分别得到

中文词数 2968　　英文词数 2564

Seq2Seq model

解码

Single Word

```
high_freq_words=[i[0] for i in cn_count.items() if i[1]>100]
for i in range(5):
    translate(random.choice(high_freq_words)+'。')
```

```
Last executed at 2022-03-13 21:09:52 in 35ms
```

```
回家 。 ------go home .
狗 。 ------the dog dog dog dog dog dog dog dog
给 。 ------give me a UNK .
今天 。 ------today today today today today today today today today
别 。 ------don't underestimate .
```

Seq2Seq model

解码

Custom Very Basic Sentences

```
translate('我回家了。')
translate('我有一条狗。')
translate('把它给我。')
translate('我今天很累。')
translate('别伤害它。')
```

```
Last executed at 2022-03-13 21:14:56 in 36ms
```

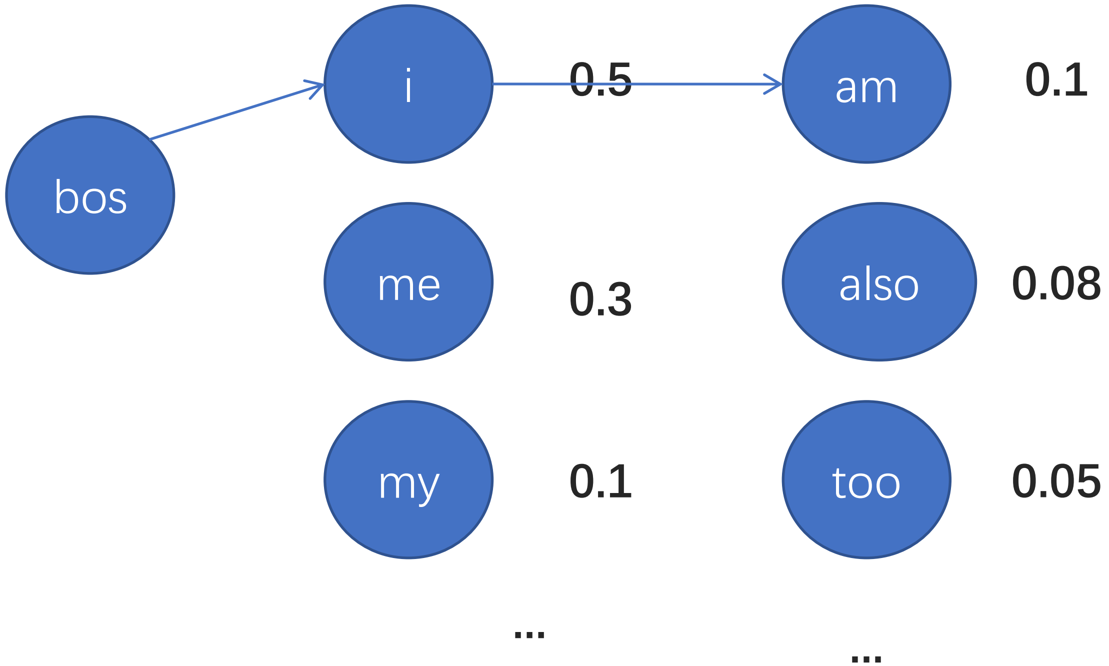
```
我 回家 了 。------i felt home .
我 有 一条 狗 。------i have a dog .
把 它 给 我 。------give me me .
我 今天 很累 。------i had a tired today today .
别 伤害 它 。------don't want to touch it it .
```

可以看出翻译效果非常垃圾

Beam Search

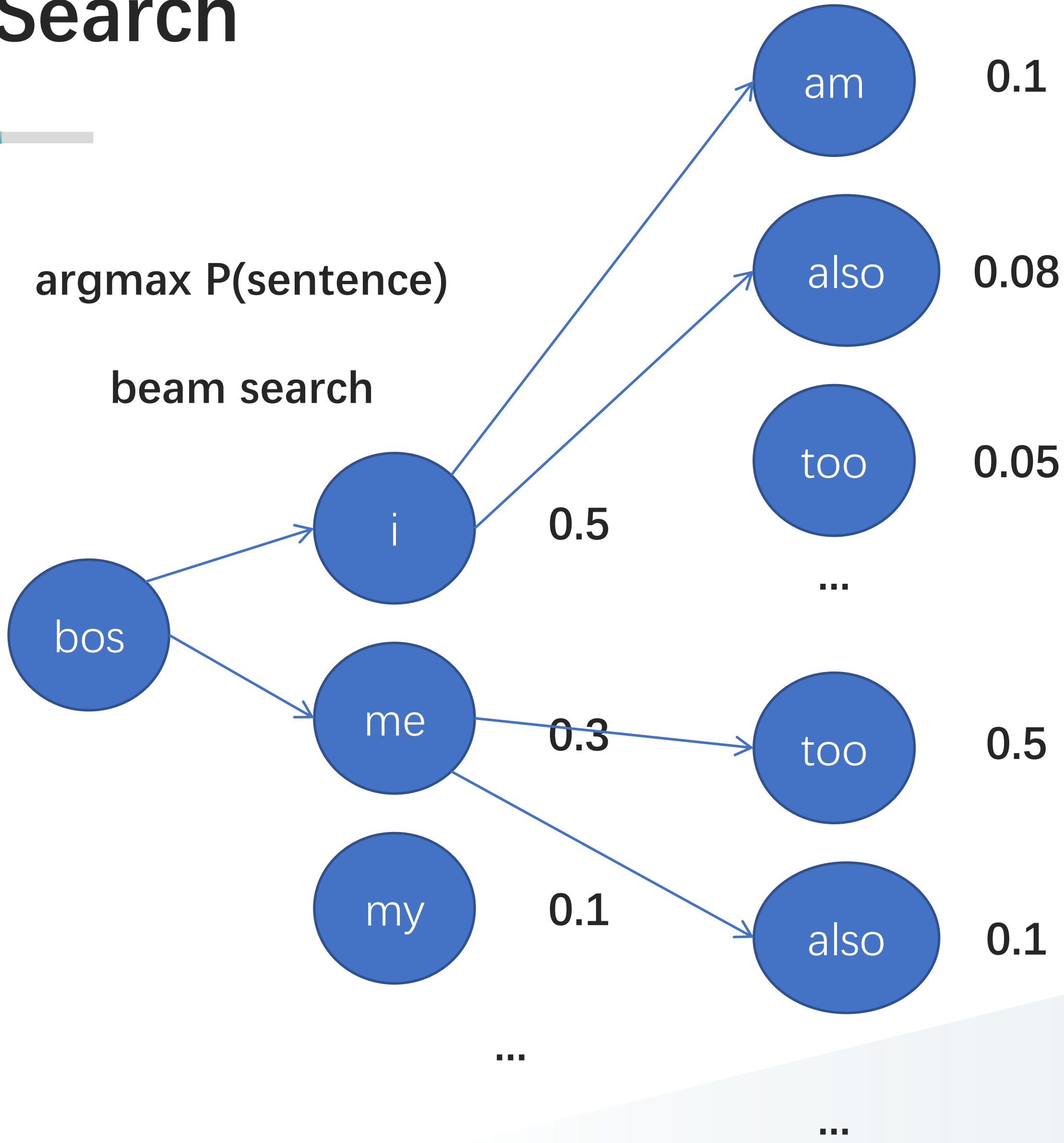
e.g. translate('我也是')

greedy decode



argmax P(sentence)

beam search



Seq2Seq model

束搜索

Beam Search

```
translate('这只猫是我的。')
```

Last executed at 2022-03-13 21:27:44 in 15ms

```
这 只 猫 是 我 的 。------this cat is UNK me me .
```

```
translate_beam('这只猫是我的。',num=3)[:10]
```

Last executed at 2022-03-13 21:27:45 in 150ms

```
[('this cat is mine . EOS', tensor(0.0195, device='cuda:0')),
 ('this cat is UNK me me . EOS', tensor(0.0169, device='cuda:0')),
 ('this is is my UNK . EOS', tensor(0.0066, device='cuda:0')),
 ('this cat is UNK me me me . EOS', tensor(0.0061, device='cuda:0')),
 ('this cat is mine me me . EOS', tensor(0.0061, device='cuda:0')),
 ("that's only hard . EOS", tensor(0.0041, device='cuda:0')),
 ("that's only something . EOS", tensor(0.0026, device='cuda:0')),
 ('this is is UNK . EOS', tensor(0.0024, device='cuda:0')),
 ("that's only hard my . EOS", tensor(0.0022, device='cuda:0')),
 ('this is just UNK . EOS', tensor(0.0021, device='cuda:0'))]
```

Transformer model

完整地使用了数据集，长句子也用了，也没有舍弃低频词，即使是只出现了一次的词。

一些参数

编解码都使用了三层，embed_size=512, nhead=8 ......

Transformer model

解码

Single Word

```
print(transformer.translate('回家。'))
print(transformer.translate('狗。'))
print(transformer.translate('给。'))
print(transformer.translate('今天。'))
print(transformer.translate('别。'))
```

Last executed at 2022-03-13 22:00:59 in 120ms

```
go home .
dogs .
give it to it .
today is today .
don't be here .
```

Custom Very Basic Sentences

```
print(transformer.translate('我回家了。'))
print(transformer.translate('我有一条狗。'))
print(transformer.translate('把它给我。'))
print(transformer.translate('今天我很累。'))
print(transformer.translate('别伤害它。'))
```

Last executed at 2022-03-13 22:03:00 in 125ms

```
i went home .
i have a dog .
give it to me .
i'm very tired today .
don't hurt it .
```

没有对比就没有伤害

Transformer model

解码

A Little Complex Sentences

```
print(transformer.translate('我比他更爱你。'))
print(transformer.translate('爸爸，我们去哪里呀。'))
print(transformer.translate('我准备明天回学校。'))
print(transformer.translate('我的乌龟昨天死了。'))
print(transformer.translate('我们应该学会享受生活。'))
```

```
Last executed at 2022-03-13 22:10:52 in 185ms
```

```
i love you more than he loves you .
dad , where we're going .
i'm ready to go back to school tomorrow .
my grandmother died yesterday .
we should learn to enjoy life .
```

Transformer model

束搜索

Beam Search

```
sentence="这真是个错误。"
print(transformer.translate(sentence))
transformer.beam_translate(sentence,2)[:10]
```

```
Last executed at 2022-03-13 22:31:35 in 1.68s
```

```
 this is a serious mistake .
[(' this is really a mistake . ', 0.04487004057884216),
 (" it's such a mistake . ", 0.03580236434936523 4),
 (' this is really an error . ', 0.024027446284890175),
 (' this is a serious mistake . ', 0.019881561398506165),
 (" it's really a mistake . ", 0.01691895909607410 4),
 (' this is a true mistake . ', 0.014445099048316479),
 (" it's really an error . ", 0.01386343780905008 3),
 (" it's such an error . ", 0.004108923487365246),
 (' this is really an mistake . ', 0.002496010391041636 5),
 (" it's really an mistake . ", 0.001990678487345576 3)]
```

也实现了一下，但是没做优化跑的有点慢，而且显存占用还极高不得不换cpu跑

Transformer model

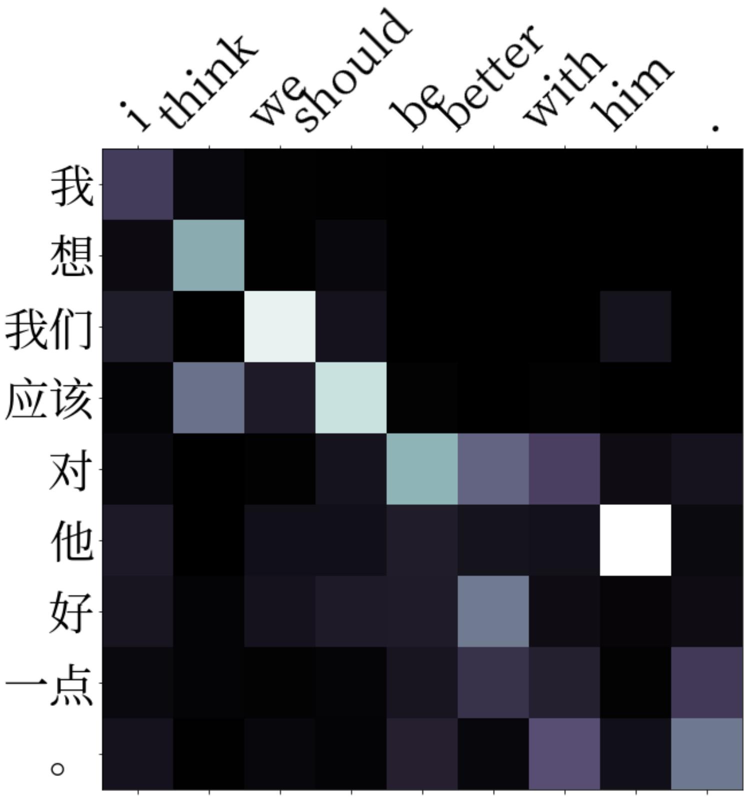
注意力

Show Attentions

```
sentence="我想我们应该对他好一点。"
print(transformer.translate(sentence,show=True))
```

Last executed at 2022-03-13 22:26:38 in 231ms

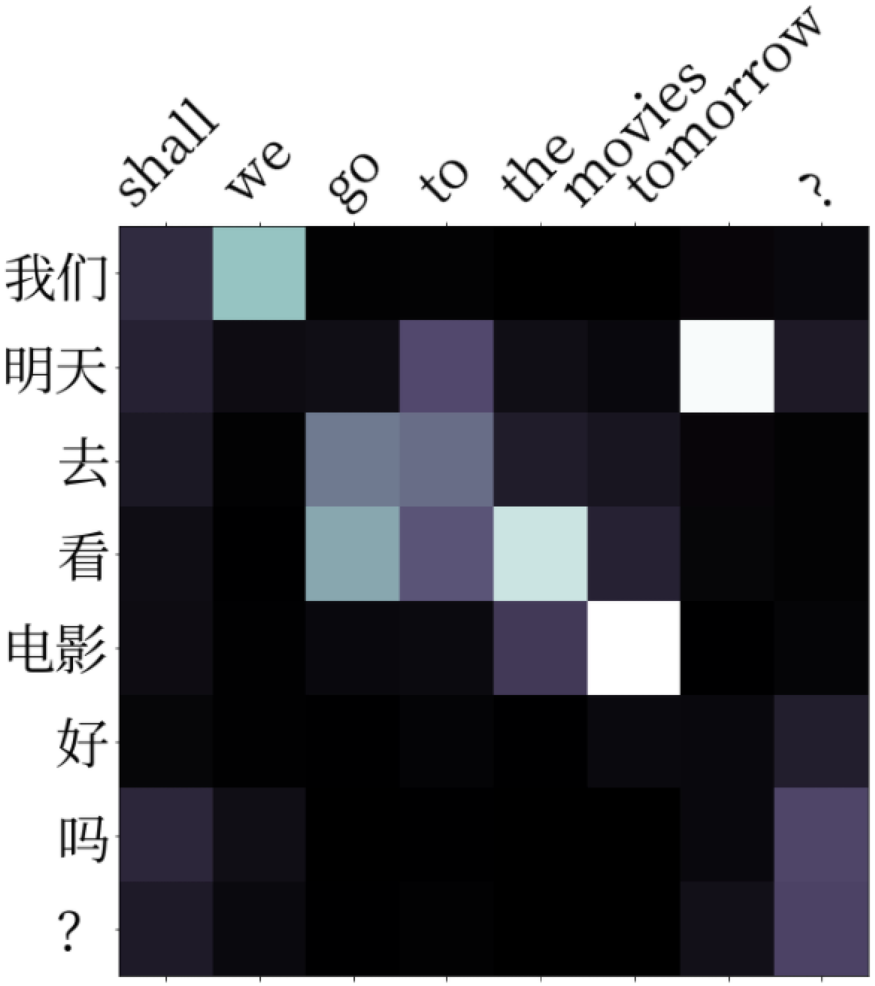
i think we should be better with him .




```
transformer.translate('我们明天去看电影好吗？',show=True)
```

Last executed at 2022-03-13 22:16:01 in 229ms

' shall we go to the movies tomorrow ? '



A Test Set

STUDYCHINESE101.COM

HOME BEGINNER INTERMEDIATE HSK TEST

intermediate ▸ 1000 Chinese Sentences In Daily Life

1000 Chinese Sentences In Daily Life

22/07/2019

数据量

462对中英语句对

数据样例

358. She mended the broken doll. - 她修补了破了的洋娃娃。 (Tā xiūbǔle pòle de yángwáwá.)
359. So I just take what I want. - 那么我只拿我所需要的东西。 (Nàme wǒ zhǐ ná wǒ suǒ xūyào
360. Spring is a pretty season， - 春天是一个好季节。 (Chūntiān shì yīgè hǎo jìjié.)
361. The figure seems all right. - 数目看起来是对的。 (Shùmù kàn qǐlái shì duì de.)
362. The stars are too far away. - 星星太遥远了。 (Xīngxīng tài yáoyuǎnle.)

Data can be downloaded from https://studychinese101.com/1000-chinese-sentences-in-daily-life.html

Bleu Score

reference: “this is a mistake”

hypothesis: “this is mistake”

P1=3/3 “this”,“is”,“mistake” 均出现

P2=1/2 “this is”出现,“is mistake” 未出现

P3=0/1 “this is mistake” 未出现

r: 参考译句长度
c: 模型输出长度

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

$$BLEU = BPexp(\sum_{n=1}^{N} w_n \log p_n)$$

nltk.translate.bleu_score

Comparation

Seq2Seq

平均解码时间：4.8ms

Bleu Score

```
corpus_bleu(labels,outputs,weights=[(1,),(0.5,0.5),(1/3,1/3,1/3),(1/4,1/4,1/4,1/4)])
```

```
[0.3830098864884658,
 0.18872479917408677,
 0.10195104729007537,
 0.05720886532975675]
```

束搜索（宽为2）Bleu Score

```
corpus_bleu(labels,beam_outputs,weights=[(1,),(0.5,0.5),(1/3,1/3,1/3),(1/4,1/4,1/4,1/4)])
```

```
[0.3997713241061398,
 0.20525431497080612,
 0.11457494414088032,
 0.06414322255004422]
```

Transformer

平均解码时间：19.3ms

Bleu Score

```
[0.5383150087082585,
 0.38130931690628656,
 0.2952351690804237,
 0.2351299128528364]
```

束搜索（宽为2）Bleu Score

```
[0.5352309990370393,
 0.3927977502424449,
 0.31172458058880687,
 0.25461999774740074]
```

THANK YOU

谢谢观看