机器翻译质量评测算法-BLEU

IT_xiao小巫 (/u/dc3bd5a17215) + 关注
2018.03.25 13:48* 字数 2229 阅读 5969 评论 4 喜欢 3
(/u/dc3bd5a17215)

> 本文介绍机器翻译领域针对质量自动评测的方法-BLEU，让你理解为什么BLEU能够作为翻译质量评估的一种指标，它的原理是什么，怎么使用的，它能解决什么问题，它不能解决什么问题。

什么是BLEU?

> BLEU (Bilingual Evaluation Understudy) is an algorithm for evaluating the quality of text which has been **machine-translated** from one **natural language** to another. Quality is considered to be the correspondence between a machine's output and that of a human: "the closer a machine translation is to a professional human translation, the better it is" – this is the central idea behind BLEU. BLEU was one of the first metrics to achieve a high correlation with human judgements of quality, and remains one of the most popular automated and inexpensive metric. -- 维基百科

解释一下，首先bleu是一种**文本评估算法**，它是用来评估**机器翻译**跟**专业人工翻译**之间的对应关系，核心思想就是**机器翻译越接近专业人工翻译，质量就越好**，经过bleu算法得出的分数可以作为机器翻译质量的其中一个指标。

为什么要用BLEU?

现实中很多时候我们需要用人工来评价翻译结果的，但这种方式非常慢，并且成本非常高，因为你需要请足够专业的翻译人员才能给出相对靠谱的翻译评估结果，一般这种人工评价都偏主观，并且非常依赖专业水平和经验。

为了解决这一问题，机器翻译领域的研究人员就发明了一些自动评价指标比如BLEU，METEOR和NIST等，在这些自动评价指标当中，**BLEU是目前最接近人类评分**的。

> METEOR和NIST评价指标，笔者还未做深入研究，有机会会针对这几个指标做个对比。

BLEU的原理是什么?

为什么BLEU能作为机器翻译的一个评估指标，还是得看看它的原理是什么。

接下来我们逐个这几个概念：

- N-gram
- 惩罚因子
- Bleu算法

N-gram

> N-gram是一种统计语言模型，该模型可以将一句话表示n个连续的单词序列，利用上下文中相邻词间的搭配信息，计算出句子的概率，从而判断一句话是否通顺。

BLEU也是采用了N-gram的匹配规则，通过它能够算出比较译文和参考译文之间n组词的相似的一个占比。

例子：

> 原文： 猫坐在垫子上
> 机器翻译：The cat sat on the mat.
> 人工翻译：The cat is on the mat.

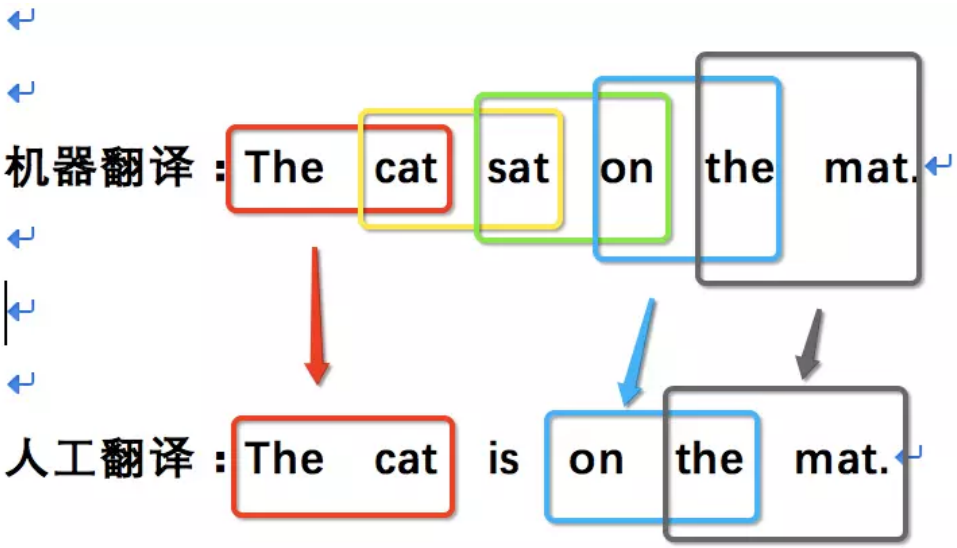
我们分别看下1-4 gram的匹配情况：

1-gram

1-gram

可以看到机器翻译6个词，有5个词命中参考以为，那么它的匹配度为 5/6。

2-gram



2-gram

2元词组的匹配度则是 3/5。

3-gram

3-gram

3元词组的匹配度是1/4。

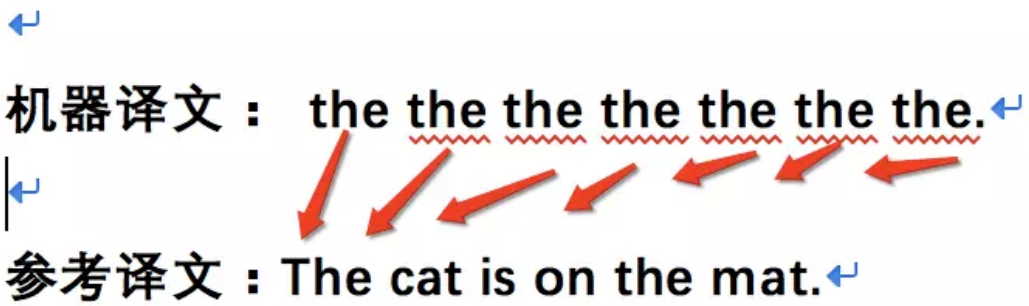
4-gram

4元词组的匹配情况就没有了。

经过上面的举例你应该很清楚n-gram是怎么计算了吧。一般情况1-gram可以代表原文有多少词被单独翻译出来，可以反映译文的**充分性**，2-gram以上可以反映译文的**流畅性**，它的值越高说明可读性越好。这两个指标是能够跟人工评价对标的。

但是它存在一些特殊情况，通过n-gram是没办法反映译文的正确性的，例如：

> 原文：猫坐在垫子上
> 机器译文： the the the the the the the.
> 参考译文：The cat is on the mat.



1-gram错误情况

如果计算1-gram的话，你会发现所有the都匹配上了，匹配度是 7/7 ，这个肯定不能反映充分性的，怎么办?

BLEU修正了这个算法，提出取机器翻译译文N-gram的出现次数和参考译文中N-gram最大出现次数中的最小值的算法，具体如下：

$$Count_{clip} = \min(Count, Max\_Ref\_Count)$$

这里写图片描述

所以上面修正后的结果应该是count = 7，Max_ref_Count = 2，取它们之间的最小值为2，那么修正后的1-gram的匹配度应该为 2/7 。

是时候拿出论文中的计算各阶N-gram的精度计算公式：

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}.$$

Pn

一眼看过去是不是觉得很高大上，看不懂了有木有，解释一下吧：

$$\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)$$

这里写图片描述

> 表示取n-gram在翻译译文和参考译文中出现的最小次数，比如上面的1-gram出现的最小次数是2.

$$\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')$$

这里写图片描述

> 表示取n-gram在翻译译文中出现次数，比如上面的1-gram出现的次数是7.

ok，到这里你基本清楚bleu中n-gram精度到底是怎么计算的了。

上面的计算已经足够好了吗？ No，还得继续改进，举个例子：

> 机器译文：The cat
> 参考译文：The cat is on the mat.

如果出现这种短句子，你会发现计算n-gram的精度会得很高分，很显然这次的得分为1，但实际上它的得分应该是比较低的。针对翻译译文长度比参考译文要短的情况，就需要一个惩罚的机制去控制。

惩罚因子

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

BP

这里的c是机器译文的词数，r是参考译文的词数，

这样的话我们重新算精度就应该是：

BP = e^(1- 6 / 2) = 7.38905609893065

> e是一个常无理数,是一个无限不循环小数,所以用e来表示2.718281828

Bleu算法

经过上面的各种改进，BLEU最终的计算公式如下：

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right).$$

BLEU

BP我们已经知道了，那么

$$\exp\left(\sum_{n=1}^{N} w_n \log p_n\right).$$

这里写图片描述

又是什么鬼？先不用想这么多，其实就是一些数学运算，它的作用就是让各阶n-gram取**权重服从均匀分布**，就是说不管是1-gram、2-gram、3-gram还是4-gram它们的作用都是同等重要的。由于随着n-gram的增大，总体的精度得分是呈指数下降的，所以一般**N-gram最多取到4-gram**。

怎么使用BLEU?

说实话，数学用人话来解释是非常困难的，我们还是来通过例子来学习，还是之前的：

> 机器翻译：The cat sat on the mat.
> 人工翻译：The cat is on the mat.

第一步：计算各阶n-gram的精度

P1 = 5 / 6 = 0.833333333333333
P2 = 3 / 5 = 0.6
P3 = 1 / 4 = 0.25
P4 = 0 / 3 = 0

第二步：加权求和

取权重：Wn = 1 / 4 = 0.25

加权求和：

$$\sum_{i=1}^{N} w_n \log P_n = 0.25 * \log P_1 + 0.25 * \log P_2 + 0.25 * \log P_3 + 0.25 * \log P_4 = -0.5198603854199589$$

加权求和

第三步：求BP

机器翻译长度 = 参考译文长度，所以：

```
BP = 1
```

最后求BLEU

$$BLEU = 1 * exp(-0.5198603854199589) = 0.5946035575013605$$

BLEU计算

写程序的时候，不用费那么大的劲去实现上面的算法，现成的工具就可以用:

```
from nltk.translate.bleu_score import sentence_bleu
reference = [['The', 'cat', 'is', 'on', 'the', 'mat']]
candidate = ['The', 'cat', 'sat', 'on', 'the', 'mat']
score = sentence_bleu(reference, candidate)
print(score)
# 输出结果：0.5946035575013605
```

BLEU的优缺点?

优点：方便、快速，结果比较接近人类评分。

缺点：

1. 不考虑语言表达（语法）上的准确性；
2. 测评精度会受常用词的干扰；
3. 短译句的测评精度有时会较高；
4. 没有考虑同义词或相似表达的情况，可能会导致合理翻译被否定；

BLEU本身就不追求百分之百的准确性，也不可能做到百分之百，它的目标只是给出一个快且不差的自动评估解决方案。

最后

BLEU原理其实并不是很复杂，更多是基于n-gram基础上的优化，写这篇文章的目的也是想梳理清楚BLEU能够解决的问题，还有不能解决的问题，这对自己后续思考如何通过其他手段去更好地提高翻译评估的能力有一定的启发作用。翻译质量评估本身就是MT领域的热门课题，如果我们能够找到一个比BLEU更好的，这将会产生很大的价值。

最后，文中很多内容从其他参考文章都可以找到，参考文章对BLEU如何计算，原理也有很不错的讲解，大家也可以参考学习下。

参考文章

- 机器翻译评测——BLEU算法详解 (https://link.jianshu.com?t=http%3A%2F%2Fwww.cnblogs.com%2Fby-dream%2Fp%2F7679284.html)
- 机器翻译评价指标之BLEU详细计算过程 (https://link.jianshu.com?t=https%3A%2F%2Fblog.csdn.net%2Fguolindonggld%2Farticle%2Fdetails%2F569666200)
- 机器翻译自动评估-BLEU算法详解 (https://link.jianshu.com?t=https%3A%2F%2Fblog.csdn.net%2Fqq_31584157%2Farticle%2Fdetails%2F77709454)
- 浅谈用Python计算文本BLEU分数 (https://link.jianshu.com?t=https%3A%2F%2Fcloud.tencent.com%2Fdeveloper%2Farticle%2F1042161)

欢迎关注我的公众号：

wwjblog

深度学习 (/nb/17021873)

IT_xiao小巫 (/u/dc3bd5a17215) ♂
写了 153248 字，被 630 人关注，获得了 618 个喜欢
(/u/dc3bd5a17215)