通过参照测试机器翻译

Transparency

Pinjia He
计算机科学系
ETH Zurich
Switzerland
pinjia.he@inf.ethz.ch

Clara Meister
计算机科学系
ETH Zurich
Switzerland
clara.meister@inf.ethz.ch

Zhendong Su
计算机科学系
ETH Zurich
Switzerland
zhendong.su@inf.ethz.ch

摘要机器翻译软件迅速
近年来由于深度神经网络的进步。人们经常使用机器翻译软件在日常生活中进行任务，如在外国餐厅订购食物，从外国医生接受医疗诊断和治疗，并在线阅读国际政治新闻。然而，由于潜在的神经网络的复杂性和难易性，现代机器翻译软件仍远非稳健，可以产生差或不正确的翻译;这可能导致误解，隐约损失，对人身安全和健康的威胁以及政治冲突。为了解决这个问题，我们介绍了介绍透明的输入（RTIS），这是一种简单，广泛适用的方法，用于验证机器翻译软件。参考透明输入是在不同上下文中使用时应该具有类似的翻译的文本。我们的实际实施，纯度，检测此属性在翻译时破坏。为了评估RTI，我们使用纯度来测试谷歌翻译和Bing
Microsoft翻译，其中200个未标记的句子，其检测到123和142具有高精度的错误翻译（79.3％和78.3％）。翻译错误是不同的，包括翻译的例子，过翻译，单词/短语误解，不正确的修改和逻辑不明确。

索引术语测试，机器翻译，参考跨
pa

I. Introduction

机器翻译软件旨在充分自动化
将文本从源语言添加到目标语言中。近年来，由于神经电机翻译（NMT）模型的发展，机器翻译软件的表现在很大程度上在很大程度上得到了改进的意思[1] –
[3]。特别是，机器翻译软件（例如，谷歌翻译[4]和Bing
Microsoft Translator
[5]）在人类评估方面正在接近人类壮观的性能。因此，越来越多的人在日常生活中使用机器翻译，以便在外语中阅读新闻和教科书等任务，在国外进行沟通，并进行国际贸易。这是在增加机器翻译软件的使用中的反映：2016年，谷歌翻译吸引了超过5亿用户，每天翻译超过100亿个单词[6];NMT模型已嵌入在各种软件应用程序中，例如Facebook [7]和Twitter [8]。

类似于传统软件（例如，Web服务器），MA
中文翻译软件可靠性非常重要。

然而，现代翻译软件已被证明返回错误的翻译，导致误解，隐约损失，对人身安全和健康的威胁以及政治融合[9] –
[14]。这种行为可以归因于基于神经网络的系统的脆性，其在自动车软件[15]，[16]，情绪分析工具[17] –
[19]和语音识别服务[20]中，可以归于基于神经网络的系统[15]，[16]，以及语音识别服务[20]，[
21]。同样地，NMT模型可以由对手示例（例如，源文本[22]中的扰动字符）或自然噪声（例如，拼写错误[23]）。这些方法产生的输入大多是非法的，即它们包含词汇（例如，“Bo0k”）或句法错误（例如，“他家”）。但是，对机器翻译软件的输入通常是词汇和句子正确的。例如，腾讯，该公司开发了一款拥有超过10亿个月活跃用户的消息应用程序，报告称其嵌入式NMT模型可以返回错误的翻译，即使输入没有词汇和语法错误[24]。

仍然有一种自动化的测试解决方案
机器翻译软件 –
至少部分原因是问题非常具有挑战性。首先，在模型培训过程中已经采用了可用于测试的大多数现有并行基础。因此，缺乏测试高质量的伪装。其次，与传统软件相比，神经电机翻译软件的逻辑大大嵌入了底层模型的结构和参数中。因此，现有的基于代码的测试技术不能直接应用于测试NMT。第三，AI（ARTI FIIEL INTERING）软件[15]，[17] –
[19]，[25]主要靶向更简单的用例（例如，10级分类）和/或具有透明oracles [26]的测试方法[25]
，[27]。相比之下，测试翻译的正确性是一个更复杂的任务：源文本可以具有多个正确的翻译，输出空间幅度大。最后但并非最不重要的是，现有的机器翻译测试技术[28]，[29]通过语言模型替换句子中的一个单词来生成测试用例（即，综合句子）。因此，它们的性能受到现有语言模型的效率的限制。

我们介绍了RTIS（借述透明输入），这是一个小说
和一般概念，作为验证机器翻译软件的方法。RTI的核心思想是通过参考透明度的启发[30]，[31]，编程中的概念

An RTI pair	Translations	Translation meanings	
Holmes in a movie based on Bad Blood	福尔摩斯在 电影的基础上坏血 (by Bing)	Holmes' blood becomes bad based on a movie	✗
a movie based on Bad Blood	一部基于坏血的电影 (by Bing)	a movie based on Bad Blood	✓

图。参考透明输入对的示例。左列中的带下划线的短语是从句子中提取的RTI。翻译的差异以红色突出显示，其含义在右侧给出。这对RTI对及其翻译是通过我们的方法作为可疑问题的报道。第一个翻译是错误的。

语言（规范功能编程）：方法应始终返回给定参数的相同值。在本文中，我们将参考透明输入（RTI）定义为应在不同上下文中具有相似翻译的文本。例如，“基于血液的电影”图1中的血液。1是RTI。关键洞察力是生成包含相同RTI的一对文本，并检查其对该货中的翻译是相似的。为了实现这一概念，我们实现了纯度，这是一种从任意文本中提取短语的工具作为RTIS。在源语言中给定未标记的文本，纯度通过组件解析器[32]提取短语，并通过用包含句子或包含短语分组RTI来构造RTI对。如果在同样的翻译之间存在大的差异，我们将这对文本报告为可疑问题。本文的关键思想在概念上与现有方法不同[28]，[29]，替换单词（即，上下文是XED）并假设翻译应该只有较小的变化。相反，本文假设RTI的翻译应该在不同的句子/短语上相似（即，上下文变化）。

我们应用纯度来测试谷歌翻译[33]和bing Microsoft Translator [34] 200句话由He等人从CNN爬行。[28]。纯度在谷歌翻译中成功报告154错误翻译和177个错误的Microsoft转换器中的错误翻译对，具有高精度（79.3%和78.3%），分别揭示了123和142个错误的翻译.1发现的翻译错误是多样化的，过翻译，单词/短语isranslation，不正确的修改和不清楚逻辑。与最先进的[28]相比，[29]，纯度可以以更高的精度报告更错误的翻译。由于其概念差异，纯度可以揭示现有方法尚未找到的许多错误的翻译（图6中所示）。此外，纯度花费了12.74年代和73.14秒，平均分别用于谷歌翻译和培育Microsoft翻译，实现了最先进的方法的可比性效率。
RTI的源代码和找到的所有错误翻译都被释放[35]，用于独立验证。源代码也将被释放以进行重用。本文的主要贡献如下：

10个错误的翻译可能出现在多个错误的翻译中 pairs (i.e., erroneous issues).

引入新颖，广泛适用的概念，
借述透明输入（RTI），用于系统机器翻译验证，

实现RTI，纯度，采用选区
解析器提取短语和单词袋（弓）模型以表示翻译，并且

验证结果表明RTI的有效性：
基于200个未标记的句子，纯度成功地发现了谷歌翻译中的123个错误翻译，并分别在Bing Microsoft翻译中的142个错误翻译分别为79.3%和78.3%的精度。

II. Preliminaries

A. Referential Transparency

在编程语言中，参考透明度
是指表达式的能力在不改变程序[30]，[31]的情况下，在程序中的相应值替换其相应的值。例如，数学函数（例如，平方根函数）是辅助透明的，而打印时间戳的函数则不是。

引用透明度已被采用作为关键功能
通过功能编程，因为它允许编译器容易地推理程序行为，这进一步促进了更高阶的函数（即，一系列功能可以粘在一起）和延迟表达式的延迟评估直到其值延迟表达式需要）[36]。术语“参考透明度”用于各种具有不同含义的各种遗嘱，例如逻辑，语言学，数学和哲学。灵感来自功能规划中的参考透明度概念，可以在RTI对中定义变质关系。

B. Metamorphic Relation

变质关系是功能的必要属性
被测软件的各势。在变质测试中[37] –
[39]，违反变质关系将是可疑的并且表明潜在的错误。我们开发了Metalymorphic关系，如下所示：RTIS（例如，名词短语）应该在不同的上下文中具有类似的翻译。正式地，对于RTI R，假设我们有两个不同的上下文C1和C2（即，不同的周围单词）。形成RTI对的C1（R）和C2（R）是含有R和两个上下文的文本。

为了测试翻译软件T，我们可以获得它们的翻译T（C1（R））和T（C2（R））。变质关系是：

$$dist_r(T(C_1(r)), T(C_1(r))) \leq d, \quad (1)$$

其中distrow表示r在t（c1（r））和t（c2（r））之间的翻译之间的距离;D是由开发人员控制的阈值。在以下部分中，我们将使用示例详细介绍我们的方法（图2）。

III. RTI AND *Purity*'S IMPLEMENTATION

本节介绍了借根透明输入

（RTIS）和我们的实施，纯洁。RTI被定义为一系列文本，该文本在文本中具有类似的翻译（例如，句子和短语）。鉴于一个句子，我们的方法打算在表现出参考透明度的句子中进行RTIS−短语 −

并利用它们来构建测试输入。要实现RTI的概念，我们实现了一个名为纯度的工具。纯度的输入是未标记的单晶句的列表，而其输出是可疑问题的列表。每个问题都包含两对文本：基本短语（即RTI）及其容器短语/句子及其翻译。请注意，纯度应检测基础或容器文本的翻译中的错误。图。图2示出了纯度使用的过程，其具有以下四个步骤：

1）识别有关透明输入。对于每一个人
句子，我们通过分析句子成分来提取一份短语列表作为RTIS。

2）在源语言中生成对。我们配对
与包含短语或原始句子的短语形成RTI对。

3）收集目标语言的对。我们养活Rti
对机器翻译软件的对进行测试并收集相应的翻译。

4）检测翻译错误。在每对，翻译
RTI对的速度相互比较。如果RTI的翻译之间存在很大差异，则纯度将该对报告为可能包含翻译错误。

算法1示出了我们的RTI实现的伪代码，其将在以下部分中详细解释。

A. Identifying RTIs

为了收集RTIS列表，我们必须提供文本

具有独特的含义，即它们的含义应该跨越上下文。为了保证RTIS的词汇和句法正确性，我们从已发布的文本中提取它们（例如，Web文章中的句子）。

特定，纯度从一组中提取名词短语

源语言中的句子作为rtis。例如，在图2中，将提取“衬套双边谈话”短语;当在不同的句子中使用时，这句话应该有类似的翻译（例如，“我参加了嵌合双边谈判。”和“她抱着无绒面双边会谈”。）为了简单和

算法1 RTI实现为纯度。要求：源头：源语言中的句子列表 d

确保：可疑的问题：可疑对列表

```
1：可疑问题←列表（ ）  用空列表初始化
2：对于在源头发送的所有来源执行3：选区树
←解析（Source发送）
5:    RTI_source_pairs ← List( )
6:    RECURSIVENPFINDER(head, List( ), RTI_source_pairs)
7:    RTI_target_pairs  TRANSLATE RTI_source_pairs)
8：在RTI目标对中的所有目标对
9:        DISTANCE target_pair > d
10：添加源对，目标对
   suspicious_issues
11: return suspicious_issues

12：功能revsivepfinder（节点，RT，所有对）13：如果
节点是叶子然后
14:
15:    if node.constituent is NP then
16:       phrase  node
17：在RTI中的所有容器短语做
18：添加容器短语，所有对短语
19:       Add phrase to rtis
20：对于Node.children（ ）的所有孩子做
21:       RECURSIVE  INDER child rtis.copy( ), all_pairs)
22:    return all_pairs

23: function DISTANCE(target_pair)
24:    rti_BOW ← BAGOFWORDS(target_pair[0])
25:    container_BOW ← BAGOFWORDS(target_pair[1])
26:    return |rti_BOW \ container_BOW|
```

为了避免语法奇怪的短语，我们只考虑本文的名词短语。

我们使用选区解析器识别名词短语，

一种易于使用的自然语言处理（NLP）工具。一个组件解析器标识了一块文本的句法结构，输出非终端节点是组成部分关系的树，终端节点是单词（图3中所示的示例）。要提取所有名词短语，我们都会遍历选区解析树并拔出所有NP（名词短语）关系。

请注意，通常，RTI可以包含另一RTI。

例如，图1中的第二个Ri对包括在内。1包含两个RTIS：“基于坏血的电影中的福尔摩斯”是含有RTI“基于坏血的电影”，当名词短语也被用作RTI时，这保持了真实，因为名词短语可以包含其他名词短语。

一旦我们从句子中获得了所有名词短语，

我们将那些含有超过10个单词的人和那些

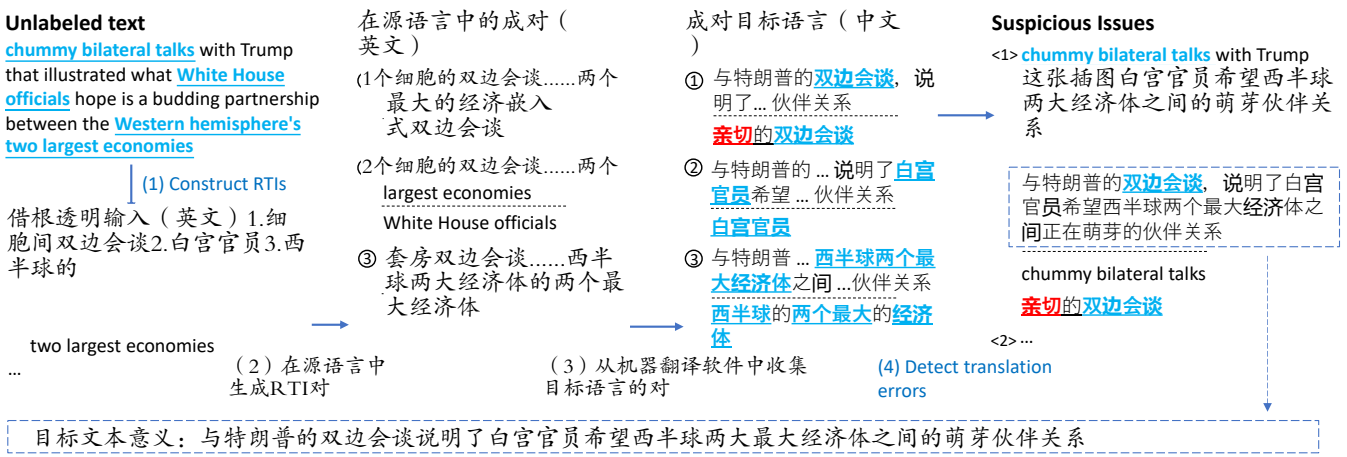


图2.我们的RTI实施概述。我们使用一个英语短语作为输入，以便清楚起见和简单。在“源语言”列的“对”列中，虚线上方的短语是原始未标记的文本。以蓝色和下划线标记的文本是RTIN或RTI对中翻译中的相同字符，而红色标记的文本是RTI的翻译中的字符，但不在其包含短语/句子中的翻译中的字符。

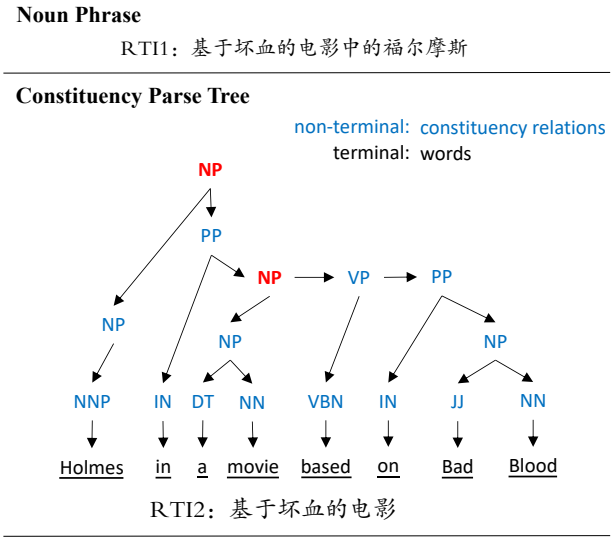


图3.选项解析树示例。粗体和红色的非终端节点是我们方法提取的RTIS。

含有少于3个单词2不是停止的字样.3这款letering帮助我们专注于更有可能携带单一含义的独特短语，大大减少误报。剩下的名词短语被认为是纯洁的rtis。

B.在源语言中生成对

一旦生成了RTIS列表，每个RTIS必须配对含有的短语，将用于参考透明度验证（III−D部分）。每次RTI都有规定的

这款体值通过网格搜索在一个数据集上经验上调。特别是，我们将在[1,20]和[2,10]中搜索最合适的值，分别为两个滤波器的步长一个。通过大多数合适的值，我们的意思是达到达到RTI之间的最高比率的最高比率和在滤器之后的名词短语之间的最高比率。

3a stop−word是主要用于结构的单词 – 而不是meaning—in a sentence, such as “is”, “this”, “an.”

对应有两个不同的文本包含相同短语。为了生成这些成对，我们将RTI与其中被发现的完整文本（如图2所示）和来自同一句子的所有包含的RTIS（即，其他包含名词短语）。例如，假设图3中的RTI1。图3是从句子中提取的RTI，可以基于组件结构找到RTI2;请注意，“基于血液的电影中的福尔摩斯”是含有RTI“基于坏血的电影”。因此，将构建3个RTI对：（1）RTI1和原始句子;（2）RTI2和原句;（3）RTI1和RTI2。

C.在目标语言中收集对

下一步是输入RTI对（在给定的源中语言）在测试的机器翻译软件上，收集其翻译（以任何所选的目标语言）。我们使用Google和Bing在我们的实施中提供的API，返回结果与谷歌翻译和Bing Microsoft Translator的Web接口相同[33]，[34]。

D. Detecting Translation Errors

最后，为了检测翻译错误，翻译从上一步的对进行RTI相似性。在翻译中检测RTI的缺点，同时避免误报是不琐碎的。例如，在图1中。在图2中，第一个对中的RTI是“嵌入式双边谈判”。鉴于整个原始句子的翻译，难以确定哪些字符参考RTI。可以在保留固有含义的同时重新排序单词，因此无法保证RTI和集装箱翻译之间的精确匹配。

NLP技术，如词对齐[40]，[41]，哪个可以在实现的此组件中将源文本中的单词/短语映射到其目标目标中的单词/短语。但是，现有工具的性能差，运行时可能非常缓慢。相反，我们采用Bagof−Lords（弓）模型，仅考虑的表示

在一段文本中的每个单词的外观（例如，参见图4）。请注意，此表示是多重状态。虽然弓形模型很简单，但它已证明在许多NLP任务中建模文本非常有效。为了纯度，使用目标文本的N−Gram表示提供了类似的性能。

BoW = {"we": *1*, "watched": *1*, "two": *2*, "movies": *1*,
"and": *1*, "basketball": *1*, "games": *1*}

图4. "我们观看了两部电影和两部电影的文字袋式表示 basketball games."

T（R），4和其集装箱T（CCON（R））。获得两种翻译（Bowr和Bowcon）的弓表示后，距离Dist（T（R），T（CCON（R））由Dist（Bowr，Bowcon）计算，如下：

$$dist(BoW_r, BoW_{con}) = |BoW_r \setminus BoW_{con}| \qquad (2)$$

用这些度量来衡量T（R）中有多少字出现，但不是在T（CCON（R））中。例如，"我们观看两部电影和两个篮球比赛"之间的距离（t（ccon（r））和"两个有趣的书"（t（r））是2.如果距离大于阈值d，这是一个选择的超参数，翻译对及其源文本将通过我们的方法作为可疑问题报告，表明至少一个翻译可能包含错误。例如，在图2中的可疑问题中。如图2所示，距离为2，因为汉字亲切不出现在容器T的平移中（CCON（R））。5

我们注意到理论上，此实现无法检测到
T（ccon（r））中的过翻误错误，因为t的附加词（ccon（r））不会改变在equ中计算的距离。然而，由于源文本CCON（R）通常在另一RI对中的RTI通常是RTI，因此不会发生该问题，在这种情况下，在后者RTI对中可以检测到过转换错误。

IV. EVALUATION

在本节中，我们评估了纯度的性能
将其应用于谷歌翻译和Bing Microsoft
Translator。本节规范，旨在回答以下研究问题：

RQ1：如何准确是这种方法，令人生畏

RQ2：我们的方法可以获得多少错误翻译

RQ3：我们的方法可以翻译错误是什么样的翻译错误

- RQ4: How efficient is the approach?

4在我们的实现中，上下文可以是空字符串。因此，
$C \quad (r) = r.$
5对于中文文本，纯度将每个字符视为一个单词。

TABLE I
STATISTICS OF INPUT SENTENCES FOR EVALUATION. EACH CORPUS CONTAINS 100 SENTENCES.

	#Words/	Average	Words	
语料库句子#单词/句子总截然不同				
Politics	4~32	19.2	1,918	933
Business	4~33	19.5	1,949	944

A.实验设置和数据集

a）实验环境：运行所有实验
在Linux工作站与6 Core Intel Core i7−8700 3.2GHz处理器，16GB DDR4 2666MHz内存，以及GeForce GTX 1070
GPU。Linux工作站正在使用Linux内核4.25.0运行64位Ubuntu 18.04.02。对于解析，我们使用zhu等人的换档减少解析器。[32]，其在斯坦福的Corenlp库中实施[42]。由于作者的知识背景，我们的实验考虑了英文→中文环境。

b）比较：我们将纯度与两个状态进行比较
最新方法：静置[28]和疯复翁（ED）[29]。我们获得了来自作者的源代码。由于工业确认，译文的作者无法释放其源代码。因此，我们仔细实施了他们的方法，并在论文中描述并咨询了工作的主作者以获得关键实施细节。
TransRepair使用0.9的阈值对于Word
Embeddings的余弦距离来生成字对对。在我们的实验中，我们使用0.8作为门槛，因为我们无法重现使用0.9的纸张报告的字对的数量。在本文中，我们评估了译者 −
因为它在谷歌翻译的四个指标中获得了最高精度，以及比变压器的频率−LCS更好的整体性能（[29]的表2）。此外，我们还使用论文中介绍的策略重新调整静坐和疯狂的参数。本评估中的所有方法都在Python中实施并发布[35]。

c）数据集：纯度测试机器翻译软件
与LexiveAdd和句法 −
正确的真实句子。我们使用从他等人发布的CNN文章中收集的数据集。[28]。此数据集的详细信息在表I中示出。此数据集包含两个语料库：政治和业务。"政治"数据集中的句子包含4 32字（平均值为19.2），它们包含1,918个单词和933个非重复字。我们使用来自两个类别的Corpora评估纯度对不同术语的句子的表现。

B.在寻找错误问题的精确度

我们的方法会自动报告可疑的问题
包含同一rid的不一致翻译。因此，该方法的有效性在于两个方面：（1）如何准确是报告的问题;（2）有多少错误翻译可以纯洁？在本节中，我们评估报告的成对的精度，即报告的许多报告

问题包含真正的翻译错误。具体而言，我们应用纯度来测试谷歌翻译和Bing
Microsoft翻译使用表I的数据集。要验证结果，两位作者手动分别检查所有可疑问题，然后统称（1）问题是否包含翻译错误（S）；（2）如果是，它包含哪种翻译错误。

1）评估度量：纯度的输出是一个列表
可疑问题，每个都包含（1）RTI，R，源语言及其翻译，T（R）；（2）源语言中的一条文本，其中包含RTI，CCON（R）及其翻译，T（CCON（R））。我们将精度定义为具有T（R）或T中具有翻译错误（CCON（R））的转换错误的百分比。明确地，对于可疑问题P，如果TP（R）或TP（CCON（R））具有转换错误（即，当可疑问题是错误的问题时，我们将错误（p）设置为true。否则，我们将错误（p）设置为false。鉴于可疑问题列表，精度计算：

$$\text{Precision} = \frac{\sum_{p \in P} \mathbb{1}\{error(p)\}}{|P|}, \qquad (3)$$

哪里是纯度和纯度返回的可疑问题是可疑问题的数量。

2）结果：结果显示在表II中。我们
观察到，如果目标是尽可能多的问题（即，D = 0），纯度达到78% 79.8%的精度，同时报告67 99错误问题。例如，当使用“业务”数据集进行策划Microsoft
Translator时，纯度报告100个可疑问题，而78则包含翻译错误，导致78%的精度。如果我们希望纯度更准确，我们可以使用更大的距离阈值。例如，当我们将距离阈值设置为5时，纯度在所有实验设置上实现100%的精度。注意，精度不会随阈值单调而增加。对于“Bing–政治”，精度将阈值从2到3更改为3.虽然误报的数量降低，但也可能减少真正的阳性数量。

在我们的比较中，我们的纯度纯洁地检测更多错误
与所有现有方法相比，精度更高的问题。要与SIT进行比较，我们专注于其系统报告的前1个结果（即最有可能包含错误的翻译）。特别是，SIT的前1个输出包含（1）原始句子及其翻译和（2）前1个生成的句子及其翻译。为了直接比较，我们认为坐的前1个输出作为可疑问题。Transrepair报告了一个可疑句子对的清单，我们将每个报告的副本视为可疑问题。正式。3用于计算比较方法的精度。结果显示在表II的最右侧。

当距离阈值处于最低（即，d = 0）时，
与坐骑和疯狂相比，纯度纯度更高的精确度。例如，当测试谷歌翻译在“政治”数据集上，纯度文件87错误的问题79.8%的精度，而仅坐在其中
34错误问题65.3%的精度。当D =
2时，纯度检测类似数量的错误问题，以坐下，但具有显着的精度。例如，当在“政治”数据集上测试Bing Microsoft
Translator时，纯度为39精度为92.8%的错误问题，而Ser Fi NDS
36错误的错误问题70.5%精度。虽然精度比较不是苹果，但我们相信结果表明了纯度的优越性。作为现实世界的源句几乎是无限的，在实践中，我们可以为此语言设置设置D =
2，以获得高精度的精确问题的不良问题。

我们认为纯度达到更高的精确度，因为
以下原因。首先，现有方法依赖于训练预先接受的型号（即，BERT [43]用于静坐和手套[44]和疯狂[45]的Spacy
[45]）来生成句子对。虽然伯特应该在这项任务上做得好，但它可能会产生奇怪的语义的句子，导致误报。不同地，纯度直接从真实句子中提取短语以构建RTI对，因此没有这种误报。此外，坐在目标句子表示和比较中依赖于依赖解析器[46]。依赖解析器可以返回不正确的依赖性解析树，导致误报。

Source text	Target text
a lot of innovation coming from other parts of the world 很多来自世界其他地方的创新 (by Bing)	
innovation coming from other parts of the world 来自世界其他地区的创新 (by Bing)	
由于较低的制造成本和强大的工会，南方作为外国制造商的新汽车制造枢纽。	由于较低的制造成本和较弱的工会，南方已成为外国制造商新汽车制造的枢纽。(by Google)
foreign makers thanks	外国厂商谢谢 (by Google)
他加入了菩提树·贝尔·林堡，埃琳娜卡根和索尼娅·索迪亚尔。	鲁思·巴德尔·金斯堡法官、埃琳娜·卡根法官和索尼娅·索托马约尔法官也加入了他的行列。(by Bing)
Justices Ruth Bader Ginsburg 法官露丝·巴德尔·金斯堡 (by Bing)	

图5.假阳性实例。

3）假阳性：纯度纯度来自
三个来源。在图1中。如图5所示，当D =
0时，我们呈现假正示例。首先，短语可以具有多个正确的翻译。如第一次示例所示，“零件”在上下文中有两个正确的翻译（即，地方和地址）“单词的其他部分”。但是，当D =
0时，将报告。此类别占纯度的大部分误报。为了减轻这种误报，我们可以调整距离阈值D或维持替代翻译词典。其次，我们用来识别名词短语的选项应力解析器可以返回一个非名词短语。在第二个例子中，“外国制造商感谢”被认为是名词短语，这导致了短语含义的变化。在我们的实验中，6个误报是由不正确引起的

6注意到精确的结果与他报告的结果不同。[28]
因为谷歌翻译和Bing Microsoft Translator不断更新他们的模型。

TABLE II
Purity'S PRECISION (# OF ERRONEOUS ISSUES/# OF SUSPICIOUS ISSUES) USING DIFFERENT THRESHOLD VALUES.

	Purity						SIT	TransRepair
	0	1	2	3	4	5		
Google-Politics	79.8% (87/109)	81.9% (59/72)	94.5% (35/37)	100% (18/18)	100% (11/11)	100% (7/7)	65.3% (34/52)	64.2% (45/70)
Google-Business	78.8% (67/85)	79.3% (46/58)	100% (21/21)	100% (5/5)	N.A.	N.A.	64.7% (33/51)	61.1% (22/36)
Bing-Politics	78.5% (99/126)	82.9% (68/82)	92.8% (39/42)	90.9% (20/22)	100% (7/7)	100% (3/3)	70.5% (36/51)	70.5% (24/34)
Bing-Business	78.0% (78/100)	80.0% (48/60)	90.9% (20/22)	90.0% (9/10)	83.3% (5/6)	100% (3/3)	62.7% (32/51)	55.0% (22/40)

来自选项解析器的输出。第三，正常名称通常是音译，因此可以具有不同的正确结果。在第三个例子中，名称“ruth”具有两个正确的音译，导致误报。在我们的实验中，1假阳性是由正确名称的音译引起的。

4）通过纯度提取的RTIS：我们手动检查了所有

335 RTIS通过纯度找到。在“政治”数据集中发现了173 RTIS，并在“商业”数据集中找到162个RTIS。在这些RTIS中，319个RTIS（95.2%）在不同上下文中使用时应该具有类似的翻译。剩下的16条流RTI是由选区解析器的错误引起的。在200个原始句子中有139个包含RTI。所有RT都形成了620个RTI对。

当距离阈值为2时，这意味着

RTI的翻译可能最多有两个不同的汉字，122个RTI对被报告为可疑问题，其余的498个RTI对没有违反我们的假设。如表II所示，115个可疑问题是真正的积极因素，而7则为误报。可以基于表II中的结果来计算在其他距离阈值下报告的RTI对的数量。

C. Erroneous Translation

我们表明纯洁可以报告错误的问题

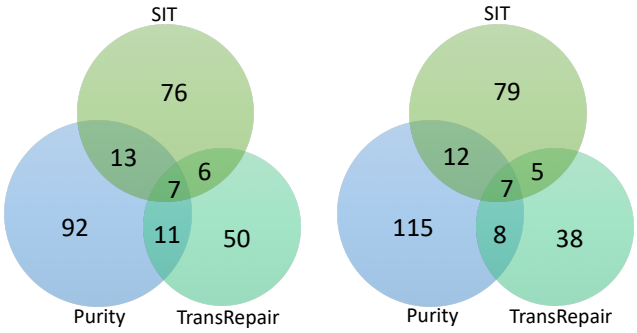
具有高精度，其中每个错误问题包含至少一个错误的翻译。因此，为了进一步评估纯度的有效性，在本节中，我们研究了如何可能会如何错误的翻译纯度。特定，如果在多个错误问题中出现错误的翻译，则将计算一次。表III呈现与表II相同的实验设置下的错误翻译数。我们可以观察到，当D = 0时，纯度找到54　74错误的翻译。如果我们打算通过设置更大的距离阈值具有更高的精度，我们将合理地获得更少的错误翻译。例如，如果我们希望实现100%精度，我们可以在谷歌翻译（D = 3）中获得32个错误的翻译。

我们进一步研究了纯度发现的错误翻译，

坐和疯狂。图。图6通过Venn图说明了结果。我们可以观察到，通过所有三种方法可以检测到谷歌翻译的7个错误翻译和来自Bing Microsoft Translator的7个错误翻译。这些是一些原始的翻译

TABLE III
THE NUMBER OF TRANSLATIONS THAT CONTAIN ERRORS USING DIFFERENT THRESHOLD VALUES.

	Purity						SIT	Trans Repair
	0	1	2	3	4	5		
Google-Politics	69	53	38	24	15	9	50	44
Google-Business	54	39	20	8	0	0	52	30
Bing-Politics	74	56	42	22	8	4	55	33
Bing-Business	68	46	20	9	6	5	48	25



Google Translate　　Bing Microsoft Translator

图6.纯度，坐着和疯复塞报告的错误翻译。

来源句子。207错误的翻译是纯度独有的，而155个错误的翻译是独一无二的，并且88个错误的翻译是疯狂的翻译。在检查所有错误的翻译后，我们在报告短语翻译错误时纯度是有效的。同时，静静的独特错误主要来自一个名词或形容词差异的类似句子。疯狂的独特错误主要来自一个数字差异的类似句子（例如，“五”→“六”）。根据这些结果，我们认为我们的方法补充了沦列状态的方法。

D.报告的翻译错误的类型

纯度能够检测多样化的翻译误差

种类。在我们的评估中，纯度已经成功地检测到5种翻译错误：翻译，过度翻译，单词/短语误解，不正确的修正，

TABLE IV
NUMBER OF TRANSLATIONS THAT HAVE SPECIFIC ERRORS IN EACH CATEGORY.

	Under translation	Over translation	Word/phrase mistranslation	Incorrect modification	Unclear logic
Google-Politics	17	9	43	5	12
Google-Business	12	6	29	8	11
Bing-Politics	8	2	51	4	23
Bing-Business	11	5	38	6	32

逻辑不清楚。表IV显示了具有特定错误的翻译数。我们可以观察到这个词/短语误解和不明确的逻辑是最常见的翻译错误。

提供揭露的多样性

错误，本节突出显示所有5种错误的示例。检测到的翻译误差的各种演示了RTI（通过纯度提供）EF
FICY和广泛的适用性。我们将这些错误与SIT
[28]对齐，因为它是发现并报告了这5种翻译错误的第一个工作。

1）在翻译下面：如果源文本的某些部分是

未翻译在目标文本中，它是一个翻译错误。例如，在图1中。在图7中，谷歌翻译没有翻译“幅度”。在翻译中经常导致不同语义含义的目标句子和缺乏关键信息。图。图2还揭示了翻译误差。在此示例中，源文本强调双边会话是套子，而目标文本中缺少此密钥信息。

来源我们工作的各种问题以及我们开始工作的数据的几乎焦虑
Target 我们正在研究的各种问题以及几乎令人焦虑的数据 (by Google)
Target meaning 我们工作的各种问题以及我们开始工作的几乎焦虑的数据

图7.检测到的翻译错误示例。

2）过翻译：如果目标文本的某些部分不是

从源文本的字或源文本的某些部分的翻译不必要多次翻译，这是一个过翻译错误。在图1中。在图8中，“是一个荣誉”，通过谷歌翻译在目标文本中两次翻译，同时只出现一次在源文本中，因此它是一个过翻译错误。过翻译带来不必要的信息，因此很容易导致误解。

3）Word /短语isranslation：如果有些单词或短语

在源文本中在目标文本中错误地翻译，它是一个单词/短语错误错误。在图1中。在图9中，“创建外壳”被转换为目标文本中的“建筑物房屋”。此错误是由Polysemy的歧义引起的。“住房”一词意味着“人们居住的一般地方”或“由地面楼和上层组成的混凝土建筑”。在这个例子中，翻译被错误地认为“住房”是指的

涵盖全国首都纪念服务的来源，然后前往德克萨斯州另一家服务以及殡仪列车是一个荣誉
Target 荣幸地报道了该国首都的追悼会，然后前往得克萨斯州进行另一项服务以及葬礼列车，这是一种荣幸 (by Google)
Target meaning 荣幸地涵盖了国家首都的纪念服务，然后前往德克萨斯州进行另一项服务，葬礼列车是一个荣誉

图8.检测到过翻误差的示例。

后来的意思，导致翻译错误。除了多义的歧义之外，Word /短语isranslation也可以由周围的语义引起。在图1的第二示例中，如图9所示，“工厂”被翻译为目标文本中的“公司”。我们认为，在NMT模型的培训数据中，“通用汽车”通常有翻译“通用汽车公司”，这导致了这种情况下的单词/短语误解错误。

不是创建住房，就业或信用广告的源广告商
Target 未制作住房，就业或信用广告的广告客户 (by Google)
目标意思是没有建立房屋，就业或信用广告的广告客户
来源通用电机厂
Target 通用汽车公司 (by Bing)
目标意思是通用汽车公司

图9.检测到的单词/短语误差错误的示例。

4）不正确的修改：如果某些修改器修改

错误的元素，它是一个不正确的修改错误。在图10中，“更适合大量业务问题”应该修改“更具规格的技能”。但是，Bing Microsoft
Translator推断它们是两个单独的条款，导致错误的修改错误。

源更具体的技能组，更适合大量业务问题
Target 更具体的技能集，更适合于许多业务问题 (by Bing)
目标意味着更具体的技能，更适合大量的业务问题

图10.检测到不正确的修改错误的示例。

5）逻辑不清楚：如果所有单词都被正确翻译

但目标文本的逻辑是错误的，这是一个不明确的逻辑错误。在图11中，Bing
Microsoft翻译器正确翻译了“批准”和“两个单独的场合”。但是，Bing Microsoft
Translator返回“批准两个单独的场合”而不是“两个单独的场合批准”，因为翻译不了解它们之间的逻辑关系。图。图1还展示了逻辑错误不明确的逻辑错误。在现代机器翻译软件返回的翻译中广泛存在不明确的逻辑错误，这在一定程度上

TABLE V
RUNNING TIME OF *Purity* (SEC)

		Google Politics	Google Business	Bing Politics	Bing Business
Initialization	Purity	0.0048	0.0042	0.0058	0.0046
RTI construction		0.83	0.85	0.86	0.89
Translation		11.51	12.22	72.79	71.66
Detection		0.0276	0.0263	0.0425	0.0301
Total		12.38	13.10	73.70	72.59
SIT		391.83	365.22	679.65	631.26
TransRepair		15.17	12.71	56.39	54.24

翻译是否真实理解某些语义含义的标志。

两个独立场合的来源批准	
Target	批准两个不同的场合 (by Bing)
意思是批准两个单独的场合	

图11.检测到不清晰的逻辑误差示例。

E. Running Time

在本节中，我们研究了EF效率（即运行时间）
纯度。指定，我们采用纯度来测试谷歌翻译和Bing Microsoft Translator与“政治”和“业务”数据集。对于每个实验设置，我们运行纯度10次并使用平均时间作为“结果”。表V显示了纯度的总运行时间以及初始化的详细运行时间，RTI对构建，翻译集合和参考透明违规检测。

我们可以观察到纯度花费不到15秒
在测试Bing Microsoft
Transler测试谷歌翻译和大约1分钟。在通过翻译器的API中，超过90％的时间使用了超过90％的时间。在我们的实现中，我们为每条源文本调用一次转换器API一次，因此包括网络通信时间。如果开发人员打算用纯度测试自己的机器翻译软件，则此步骤的运行时间将更少。

表V还介绍了坐下的运行时间和tran
使用相同的实验设置srepair。SIT花费超过6分钟才能测试谷歌翻译和大约11分钟才能测试Bing Microsoft
Translator。这主要是因为SIT为“政治”数据集和“业务”数据集的41,897个单词翻译44,414个单词。同时，纯度和型号需要较少的翻译（7,565和6,479，纯洁和4,271和4,087个用于疯狂的人）。基于

这些结果，我们得出结论，纯度达到了最先进的方法的相当效率。

F.用纯度报告的错误进行微调

测试的最终目标是改进软件RO
胸围。因此，在本节中，我们研究报告的错误是否可以充当针对FI
NE−TOENING设置，以提高NMT模型的鲁棒性，并在测试期间快速发现错误。微调是NMT中的常见做法，因为目标数据的域（即运行时使用的数据）通常与训练数据的域不同，[47]，[48]。为了模拟这种情况，我们培养一个全球关注的变压器网络[3]−a为NMT模型的标准架构 − 关于WMT'18
ZH−EN（中英文）语料库[49]，其中包含 20米句子对。我们将翻译的标准方向（即en−Zh）逆转，与我们的其他实验相比。我们使用Fairseq Framework [50]创建模型。

要测试我们的NMT模型，我们爬行了10篇最新文章
在“娱乐”类别的CNN网站下，随机提取80个英语句子。数据集收集过程与“政治”和“业务”数据集[28]对齐，在主实验中使用。我们使用训练有素的模型作为被测系统运行纯度与“娱乐”数据集;纯度成功地证明了42个错误的翻译。我们手动将其标记为正确的翻译，并在这42个翻译对中对NMT模型进行标记为8个时期 −
直到WMT'18验证集的丢失停止减少。此之后，我们调整后，40个句子中的40个被正确翻译。未纠正的两个翻译中的一个可以归因于解析错误;虽然另一个（源文本：“一个用于最佳导演”）有一个“暧昧的参考”问题，其基本上使得在没有上下文的情况下难以翻译。同时，WMT'18验证集上的BLEU分数在标准偏差范围内保持良好[51]。这证明了纯度报告的错误确实可以在没有从划痕中重新培训模型的情况下进行XEED −
资源和时间密集的过程。

V. DISCUSSION

A. RTI用于强大的机器翻译

在本节中，我们讨论了参考跨境的效用
景观人建立强大的机器翻译软件。与传统软件相比，机器翻译软件的误差过程可以说是更困难的，因为NMT模型的逻辑在于复杂的模型结构和其参数而不是人类可读代码。即使可以识别导致错误的计算，也可以识别出现错误，通常不清楚如何更改模型以纠正错误而不会引入新错误。虽然模型校正是一个难以打开的问题，但不是我们论文的主要焦点，但我们发现通过纯度发现的翻译错误可以用于FI X并改善机器翻译软件。

用于在线翻译系统，最快的方法
isrranslation是难以编码的翻译对。就这样

开发人员可以快速且容易地解决纯度的翻译错误，以避免可能导致负面影响的错误[9] –
[14]。更强大的解决方案是将误读纳入训练数据集。在这种情况下，开发人员可以将翻译错误的源句添加到神经网络的训练集和培训或恢复网络的正确转换。在从头开始重新培训大型神经网络时可能需要数天，在几百个错误上的内部调整只需要几分钟，即使是大型的SOTA型号。我们注意到这种方法并不绝对保证误解将被误导，但我们的实验（IV-F部分）显示它在解决错误方面非常有效。开发人员也可以将报告的问题有用，这对于进一步分析/调试有用，因为它类似于通过输入最小化/本地化调试传统软件。此外，由于RTI报道的结果成对，它们可以用作数据集以用于对翻译错误的未来实证研究。

B. Change of Language

在我们的实施，纯度，我们使用英语作为源
语言和中国人作为目标语言。要匹配我们的确切实现，需要有一个选区解析器或数据以在所选的源语言中培训此类解析器，因为这是我们如何提供RTIS。斯坦福Parser7.

目前支持六种语言。或者，人们可以训练解析器，例如zhu等。[32]。其他纯度模块保持不变。因此，原则上，它非常容易重新定位RTI到其他语言。请注意，虽然我们预计RTI属性持有大多数语言，但在打破假设的语言结构中可能会有混淆因素。

VI. Related Work

A. Robustness of AI Software

最近，Artifial Intelligence（AI）软件已经存在
许多领域采用;这主要是由于深神经网络的建模能力。然而，这些系统可以产生错误的输出，例如，导致致命事故[52] –
[54]。为了探讨AI软件的稳健性，一系列研究专注于攻击使用深神经网络的不同系统，例如自主车[25]，[55]和语音识别服务[20]，[56]。这些工作旨在通过喂养具有难以察觉的扰动（即，对抗示例）的输入来欺骗AI软件。同时，研究人员还设计了改善AI软件的鲁棒性的方法，例如鲁棒训练机制[57] –
[59]，对抗的例子检测方法[60]，[61]和测试/调试技术[15]，[16]，[62] –
[67]。我们的论文还研究了广泛采用的AI软件的稳健性，但侧重于这些论文尚未探索的机器翻译系统。此外，这些方法中的大多数是白箱，利用梯度/激活值，而我们的方法是黑盒，根本不需要模型内部细节。

B. Robustness of NLP Systems

灵感来自于计算机视觉的鲁棒性研究，
NLP（自然语言处理）研究人员已经开始探索各种NLP系统的攻击和防御技术。典型的实例包括情绪分析[17] –
[19]，[68]，[69]，文本意外[18]和有毒内容检测[19]。但是，这些都是基本的分类任务，而机器翻译软件在模型输出和网络结构方面更复杂。

其他复杂的NLP系统的稳健性也有
近年来研究过。贾和梁[27]提出了稳健性评估计划，为斯坦福问题接听数据集（阵容），广泛用于阅读理解系统的评估。他们发现即使是最先进的系统，在人类F1分数附近实现，当插入对抗句时，就会正确地回答关于段落的问题。mudrakarta等。[26]还生成问题的问题，用于在文本的图像，表和段落上接听任务。这些方法通常会扰乱系统输入并假设输出（例如，人物或特定年份）应保持不变。但是，机器翻译的输出（即，一条文本）更复杂。特别是，一个源句可以有多个正确的目标句子。因此，测试机器翻译软件是本文的目标，更脆弱。

C. Robustness of Machine Translation

最近，研究人员已经开始探索鲁棒性
NMT模型。Belinkov和Bisk
[23]发现源码句子中的合成（例如，字符掉期）和自然（例如，拼写错误）噪声可以打破基于角色的NMT模型。相比之下，我们的方法旨在将LEXIVESAND和句法校正源文本引起，这些源文本导致机器翻译软件的错误输出，在实践中更常见的错误。为了提高NMT模型的稳健性，已经研究了各种鲁棒训练机制[70]，[71]。特别地，在训练期间将噪声添加到输入和/或内部网络嵌入。与这些方法不同，我们专注于测试机器翻译。

Z很get AL. [24] proposed specialized approaches to的
Tect分别使过度翻译错误分别。与他们不同，我们的方法旨在解决翻译中的一般错误。他。[28]和Sun等人。[29]提出的普通翻译错误的变质测试方法：它们通过句子结构比较两个类似的句子（即，一个单词不同）的翻译[28]和四个

子字符串的现有度量分别[29]。此外，Sun等人[29]设计了一种自动翻译错误修复机制。与这些方法相比，RTI可以更具错误的翻译，具有更高的精度和相当的EF效率。报告的翻译误差是多样化的，并与现有论文发现的误区有所不同[28]，[29]。因此，我们认为RTI可以恭维最先进的方法。gupta等。[72]制定了翻译

[7]https://nlp.stanford.edu/software/lex-parser.html#Download

基于病理不变性的测试方法：不同含义的句子不应具有相同的翻译。我们没有与本文进行比较，因为它基于正交方法，我们将其视为并发工作。

D. Metamorphic Testing

变质测试的关键概念是检测违规行为

输入输出对的变质关系。变质测试已被广泛用于测试传统的软件，例如编译器[73]，[74]，科学文库[75]，[76]和面向服务的应用[77]，[78]。由于其对测试“不可验证”系统的有效性，研究人员还为各种AI软件设计了变质测试技术。典型的例子包括自主车[16]，[79]，统计分类器[80]，[81]和搜索引擎[82]。在本文中，我们介绍了一种用于机器翻译软件的新型变质测试方法。

VII. Conclusion

我们介绍了一般概念 – 参考译文

父输入（RTI）–
用于测试机器翻译软件。与现有的方法相比，这涉及自然句子中的单词（即，上下文是隐藏的）并假设翻译应该只有较小的变化，本文假设RTIS在不同上下文中存在类似的翻译。结果，RTI可以报告不同种类的不同转换错误（例如，短语翻译中的错误），从而补充了现有方法。
RTI的独特性好的是其简单性和广泛的适用性。我们使用它来测试谷歌翻译和Bing Microsoft
Translator，并分别发现了123和142个错误的翻译，以及最先进的运行时间，清楚地展示了纯度 –
测试机器翻译软件的RTI−Revented的能力。对于未来的工作，我们将继续进行一般方法并将其扩展到其他RTI实现，例如使用动词短语作为rtis或将整个句子视为rtis，将它们与语义上的串联配对。我们还将推出广泛的翻译错误诊断和机器翻译系统自动修复。

Acknowledgments

我们感谢匿名ICSE评论者的价值

关于本文早期汇票的反馈。此外，从斯坦福大学NLP组的语言解析器统一的工具实现是巨大的[42]。

References

[1] B. Zhang，D. Xiong和J. Su，“通过一个加速神经变压器平均关注网络，“在Proc。2018年第56届计算语言学协会年会（ACL）。

[2] J. Gehring, M. Auli, D. Grangier, and Y. N. Dauphin, “A convolutional 神经机翻译的编码器模型，“in
proc。2017年第55届计算语言学协会年会（ACL）。

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, 我是Kaiser，Lukasz Abd
Polosukhin，“关注你所需要的，”在Proc。第三次关于神经信息处理系统（Neurips）的第33次会议，2017。

[4] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun，Y.Cao，Q. Gao，K.
Macherey等，“Google的神经机翻译系统：弥合人类和机器翻译之间的差距，”Arxiv预印符号arxiv：1609.08144,2016。

[5] H. Hassan，A. Aue，C. Chen，V.Chowdhary，J.Clark，C. Federmann，X. Huang，M. Junczys−Dowmunt，W. Lewis，M.
Li等人，“在自动汉语中实现人类平价”英语新闻翻译，“Arxiv预印迹：1803.05567，2018。

[6] B. Turovsky。（2016）谷歌翻译十年。[在线的]。可用的：https://blog.google/products/translate/ten-years-of-google-translate/

[7] Facebook。（2019）如何翻译帖子或注释
another language? [Online]. Available: https://www.facebook.com/help/509936952489634/

[8]推特。（2019）关于推文翻译。[在线的]。可用：HTTPS：//help.twitter.com/en/using-twitter/translate-tweets

[9] A. Okrent。（2016）9造成大的翻译错误
problems. [Online]. Available: http://mentalfloss.com/article/48795/9-little-translation-mistakes-caused-big-problems

[10] F.麦克唐纳。（2015）有史以来最大的错位。
[Online]. Available: http://www.bbc.com/culture/story/20150202-the-greatest-mistranslations-ever

[11] T. ONG。（2017）Facebook在错误翻译看到后道歉
巴勒斯坦人因发布'早上好'而被捕。[在线的]。可用：https://www.theverge.com/us−world/2017/10/10/10/10/11/10/11/16533496/facebookapology−wrong−translation−palestinian−arrested−post−good−morning.

[12] G.戴维斯。（2017）巴勒斯坦人被警察逮捕
在Facebook上发布“早上好”后
这被错误翻译为“攻击他们”。[在
line]. Available: https://www.dailymail.co.uk/news/article-5005489/Good-morning-Facebook-post-leads-arrest-Palestinian.html

[13] T. W. Olympics。（2018）向挪威交付的15,000个鸡蛋
谷歌翻译错误后奥林匹克团队。[在线的]。可用的：https://www.nbcwashington.com/news/national-international/Google-Translate-Fail-Norway-Olympic-Team-Gets-15K-Eggs-Delivered-473016573.html

[14] B. Royston。（2018）以色列Eurovision Winner Netta叫'真实的牛'由自动翻译总理失败。[在线的]。avail
able: https://metro.co.uk/2018/05/13/israel-eurovision-winner-netta-called-a-real-cow-by-prime-minister-in-auto-translate-fail-7541925/

[15] K. Pei，Y.Cao，J. Yang和S. Jana，DeepXplore：自动白箱深层学习系统的测试，“普通学习系统”。第26次关于操

[16] Y. Tian，K.Pei，S. Jana和B. Ray，“Deeptest：自动化测试deep-neural-network-driven autonomous cars,” in *ICSE*, 2018.

[17] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W.
chang，“生成自然语言逆势例子”，升级。2018年自然语言处理的实证方法（EMNLP）的大会上的2018年。

[18] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer, “Adversarial 示例生成具有语法控制的释义网络，“在Proc中”。2018年人类语言技术（NAACL−HLT）的北美章节年会第16届北美章节年会。

[19] J. Li，S. Ji，T. You，B. Li和T. Wag，Texbugger：发电一般反对现实世界应用的对抗文本，“在Proc。第26届年度网络与分布式系统安全研讨会（NDSS），2019。

[20] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. WAGNER和W.周，“隐藏的语音命令”，在PROC中。第25届Usenix安全研讨会（Usenix Security），2016年。

[21] Y. Qin, N. Carlini, I. Goodfellow, G. Cottrell, and C. Raffel, “Imper-用于自动语音识别的可靠，鲁棒和有针对性的对手示例，“在Proc中。第36国际机器学习国际会议（ICML），2019年。

[22] J. Ebrahimi，D. Lowd和D. Dous，“关于对抗的例子字符级神经电机翻译，“在Proc。第27届计算语言学国际会

[23] y.Belinkov和Y.Bisk，“综合和自然噪音都突破神经网络机器翻译，“在proc。第六届国际学习陈述会议（ICLR）

[24] W. Z恒，W. Wang，D. l IU，C. Zhang，Q. Zen G，Y. Deng，W. yang，P. He，And T. Xie，“测试不可遗传的神经机器翻译：工业案例，”Arxiv预印迹arxiv：1807.02340,2018。

[25] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2015.

[26] P. K. Mudrakarta, A. Taly, M. Sundararajan, and K. Dhamdhere, “Did
模型了解这个问题？ “在proc。2018年第56届计算语言学协

[27] R.Jia和P. Liang， “评估阅读的对抗例子
理解系统，“陷入困境。2017年自然语言处理的实证方法

[28] P. HE，C. MEISTER和Z. SU， “机器的不变性测试
translation,” in *ICSE*, 2020.
[29] Z. Sun, J. M. Zhang, M. Harman, M. Papadakis, and L. Zhang,
“自动测试和改进机器翻译”，ICSE，2020。

[30] H. Søndergaard and P. Sestoft, “Referential transparency, definiteness
and unfoldability,” *Acta Informatica*, 1990.
[31] P.-Y. Saumont. (2017) What is referential transparency? [Online].
Available: https://www.theverge.com/2016/2/29/11134344/google-self-
driving-car-crash-report
[32]朱，Y.张，W.陈，张和J.朱， “快速准确
转移 -
减少成分解析，“在Proc中。在计算语言学协会的第51次年会中（
第1卷：长篇论文）。计算语言学协会，2013年8月，第434−443。
[33]谷歌翻译。[在线的]。可用：https://translate.google.com [34] Bing
Microsoft Translator。[在线的]。可用：https://www.bing.com/
translator
[35] Machine translation testing. [Online]. Available: https://github.com/
RobustNLP/TestTranslation
[36]凯文。（2018）为什么功能编程？好处
referential transparency. [Online]. Available: https://sookocheff.com/
post/fp/why-functional-programming/
[37] T. Y.陈，S. C. Cheung，以及S. M. Yiu， “变质测试：a
香港科技科技大学计算机科学系技术报告，香港科技技术举报技术
报告“HKSTCS98−01技术报告”。REP。，1998年。

[38] S.Segura，G. Fraser，A. B. Sanchez和A. Ruiz−Cort'， “调查
关于变质测试，“软件工程（TSE）的IEEE交易，Vol。201

[39] T. Y. Chen，F.−C.Kuo，H. Liu，P.−1.Poon，D. Towery，T.TSTSTS和Z.Q.
周， “变质测试：挑战和机遇审查”，ACM计算调查（CS

[40] Y.刘和M. Sun， “对比无监督的词对齐
非本地特征，“在proc。第29届AAAI智力大会（AAAI），

[41] A. Fraser和D.Marcu， “测量词对齐质量
统计机器翻译，“计算语言学，2007。
[42] S. N.组。Stanford Corenlp − 自然语言软件。[在线的]。
Available: https://stanfordnlp.github.io/CoreNLP/
[43] J. Devlin，M. −w。张，K. Lee，K. Toutanova， “伯特：预训练
关于语言理解的深度双向变压器，“Arxiv预印迹arxiv：181

[44]手套。[在线的]。可用：https://nlp.stanford.edu/projects/glove/ [45]
spacy。[在线的]。可用：https://spacy.io/ [46] D. Chen和C.
Manning， “一种快速准确的依赖性解析器使用
神经网络，“在Proc。2014年自然语言处理的实证方法大会

[47] R. Sennrich，B. Haddow和A. Birch，改善神经机
与单语数据的翻译模型，“2016年计算语言学协会第54届
年度会议的会议记录”中，“2016年。

[48] C.楚，R. dabre和S. Kurohashi， “经验性比较
神经机翻译简单域适应方法，“2017年计算语言学协会第5
5届年会的汇报中”。

[49] O. r. Bojar, C. Federmann, M. Fishel, Y. Graham, B. Haddow,
M. HUCK，P. Koehn和C. Monz， “2018的调查结果
机会翻译会议（WMT18），“在诉讼程序中
第三次关于机器翻译会议，第2卷：共享任务文件。比利时
，布鲁塞尔：计算协会
Linguistics, October 2018, pp. 272–307. [Online]. Available: http:
//www.aclweb.org/anthology/W18-6401
[50] Fairseq：序列建模的一个快速，可扩展的工具包。[在线的]。
Available: https://github.com//pytorch/fairseq
[51] M. POST， “呼吁清楚地报告BLEU得分”，2018年。[52]
C. Ziegler。（2016）谷歌自动驾驶汽车造成撞车
first time. [Online]. Available: https://www.theverge.com/2016/2/29/
11134344/google-self-driving-car-crash-report

[53] F. Lambert。（2016）了解自动驾驶仪的致命特斯拉事故
and the nhtsa probe. [Online]. Available: https://electrek.co/2016/07/01/
understanding-fatal-tesla-accident-autopilot-nhtsa-probe/
[54]莱文。（2018）Tesla致命崩溃：'Autopilot'模式
在司机杀死之前加上了汽车，报告了。[在线的]。
Available: https://www.theguardian.com/technology/2018/jun/07/tesla-
fatal-crash-silicon-valley-autopilot-mode-report
[55] A. Athalye，N. Carlini和D. Wagner， “混淆渐变给了一个
虚假安全感：对抗对抗例子的防御防御，“Proc。第35届国
际机械学习会议（ICML），2018。

[56] T. D U, S. J i, J. l i, Q. GU, T. Wang, and R. be雅虎， “SI人attack:
为端到端声学系统产生对抗性音频，“ARXIV预印迹ARXI
V：1901.07846,2019。
[57] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards
深度学习模型耐受对抗攻击，“普罗华舞。第六届国际学

[58] J. Lin，C.GaN和S. Han， “防御量化：当EF效率时
符合鲁棒性，“普罗斯队”。第七届国际学习陈述会议（I

[59] C. Mao，Z. Zhong，J. Yang，C.Vondrick和B. Ray， “公制学习
对于对抗性鲁棒性，“普通话。第35次神经信息处理系统

[60] G. TAO，S. MA，Y. LIU和X. Zhang，攻击符合可解释性：
属性转向检测对抗性样本，“在proc中。第34次关于神经信

[61] J. Wang，G. Dong，J. Sun，X. Wang和P. Zhang， “对抗样本
通过模型突变测试检测深度神经网络，“2019年ICSE。

[62] l. MA, F. J UE非−X U, F. Zhang, J. sun, M. X UE, B. l i, C. Chen, T. SU,
L. Li，Y.
Liu等，“DeepGauge：深度学习系统的多粒度测试标准”，2018年ASE。
[63] S. MA，Y. Liu，W.−C。Lee，X. Zhang和A. Grama， “模式：自动化
通过状态差分分析和输入选择调试神经网络模型，“proc。
2018年软件工程基金会欧洲软件工程会议与研讨会26号AC
M联席会议中的第26届ACM联席会议。

[64] J. Kim，R. Feldt和S. Yoo “指导深度学习系统测试
using surprise adequacy,” in *ICSE*, 2019.
[65] J. M. Zhang，M. Harman，L. Ma和Y. Liu， “机器学习测试：
Survey, landscapes and horizons,” *arXiv preprint arXiv:1906.10742*,
2019.
[66] X. D U, X. X IE, Y. l i, l. MA, Y. l IU, and J. Zhao, “deep stellar: model
基于基于对状态深度学习系统的定量分析，“Proc。2019年
软件工程基金会欧洲软件工程会议和研讨会上的第27届AC
M联席会议（Esec / FSE）。

[67] X. X IE, l. MA, F. J UE非−X U, M. X UE, H. Chen, Y. l IU, J. Zhao,
B. Li，J. Yin和S.
See，“DeepHunter：深度神经网络的覆盖引导模糊试验框架，”
普罗华舞中“第28届ACM
Sigsoft国际软件检测和分析国际研讨会（ISSTA），2019。
[68] D. Pruthi, B. Dhingra, and Z. C. Lipton, “Combating adversarial
拼写错误具有强大的词识别，“在proc。2019年第57届计算
语言学协会年会（ACL）。
[69] M. T. Ribeiro, S. Singh, and C. Guestrin, “Semantically equivalent
调试NLP模型的对抗规则，“在Proc中。2018年第56届计算

[70] Y. Cheng, Z. TU, F. me ng, J. Z还, Andy. l IU, “toward是robust
神经机翻译，“在Proc。2018年第56届计算语言学协会年会

[71] Y. Cheng，L. Jiang和W. Macherey，“强大的神经机译
双重逆势投入，“在Proc。2019年第57届计算语言学协会年

[72]古普塔，P.HE，C. MEISTER和Z. SU， “机器翻译测试
通过病理不变性，“在ACM联合欧洲软件工程会议和专题
讨论会上的软件工程（ESEC / FSE），2020年。

[73] V.Le，M. afshari和z. su， “通过等价mod编译器验证
ULO投入，“在ACM
Sigplan编程语言设计和实施（PLDI）会议中，2014年。
[74] C. Lidbury, A. Lascu, N. Chong, and A. F. Donaldson, “Many-core
编译器模糊，“在ACM
Sigplan编程语言设计和实施（PLDI）会议中，2015年。

[75] J. Zhang, J. Chen, D. ha O, Y. X ion G, B. X IE, l. Zhang, and H. M诶，“基于搜索的多项式变质关系推断”，2014年ASE。

[76] U. Kanewala, J. M. Bieman, and A. Ben-Hur, “Predicting metamorphic 测试科学软件的关系：使用图形内核的机器学习方法，“软件测试，验证和可靠性（STVR），Vol。26，不。2016年3月3日。

[77] W.K. Chan，S. C. Cheung和K. R. Leung，“朝向变质”以服务为导向的软件应用程序的测试方法，“in

[78] －，“服务在线测试的变质测试方法面向导向的软件应用，“Web服务研究（IJWSR），Vol。4，不。2007年2日。

[79] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid, “Deeproad: 基于GaN的变质自主驾驶系统测试，“在2018年ASE。

[80] X.Xie，J.Ho，C. Murphy，G. Kaiser，B. Xu和T. Y. Chen，申请变质试验对监督分类，“在Proc。第9届质量软件国际会议

[81] X.Xie，J.W.Ho，C.Murphy，G. Kaiser，B. Xu和T. Y. Chen，“测试通过变质测试，“系统和软件（JSS）”，Vol“验证机器学

[82] Z. Q. Q.周，S. Xiang和T. Y. Chen，软件的变质测试质量评估：对搜索引擎的研究，“软件工程（TSE）的IEEE交易，Vol。2016年42日。