doi:10.3772/j.issn.2095-915x.2023.02.009

多引擎机器翻译译文重排序与融合研究

李铭　张克亮　唐亮　夏榕璟

战略支援部队信息工程大学　洛阳　471003

开放科学（资源服务）标识码（OSID）

摘要：[目的 / 意义] 使用不用的模型、方法、语种、数据构建的机器翻译引擎往往在不同的场景下具有不同的翻译效果。因此，很多研究者都在构建机器翻译引擎时尝试使用多引擎译文融合或多翻译方法融合的方式来利用不同翻译引擎的优点，然而过往的工作没有考虑到如何利用用户在使用多引擎机器翻译所产生的数据来获取存在于用户认知域中对这些引擎译文的评价。[方法 / 过程] 本文研究提出了基于六个翻译引擎的多引擎翻译平台。该平台在长期使用中产生了翻译结果、用户特征、人工校译等数据，本文基于以上大规模历史数据构建了翻译模型训练资源库，结合 Page Rank 算法、贝叶斯公式和 UNQE 方法提出了多引擎机器翻译译文重排序方法，并利用译文重排序的结果与翻译模型训练资源库中的翻译实例相关数据，进一步使用 Transformer 架构训练了译文融合模型。[局限] 所提方法存在冷启动问题，需要一定时间、大量用户的真实数据才能够实现预期效果。[结果 / 结论] 实验结果表明了本文提出的方法能够融合多引擎优势，提高不同领域的平均译文质量。

关键词：多引擎机器翻译；译文重排序；译文融合

中图分类号：TP391　G35

A Study of Re-ranking and Combination for Multi-engine Machine Translation

LI Ming　ZHANG Keliang　TANG Liang　XIA Rongjing

Information Engineering University (Luoyang), Luoyang 471003, China

Abstract: [Objective/Significance] Machine Translation (MT) engines trained with different models, methods, language and data have different performance for multiple specific translation scenario. Thus, a number of research tried to use multi-engine or multi-method combination approach for constructing MT system with advances of each MT engine. [Methods/Processes] This research provides a multi-engine platform with six different MT engines. During the long-term using of it, there comes a huge

作者简介　李铭（1989-），博士研究生，研究方向为语言信息处理、计算语言学、机器翻译；张克亮（1964-），博士，教授，研究方向为计算语言学、语言智能处理，E-mail：kliang99@sina.com；唐亮（1976-），博士，副教授，研究方向为知识图谱，智能检索；夏榕璟（1998-），硕士研究生，研究方向为语言信息处理。

引用格式　李铭 , 张克亮 , 唐亮 , 等 . 多引擎机器翻译译文重排序与融合研究 [J]. 情报工程 , 2023, 9(2): 96-107.

amount of data of translation instances, user profiles and human translates. A resource warehouse for translation model training is constructed using these data. we offer a method of multi-engine MT re-ranking using the resource warehouse with Page Rank Algorithm, Bayes Rule and UNQE. Furthermore, we use the result generated by the re-ranking method with human translations provided by the resource warehouse to train a translation combination model. [Limitations] This Method has cold boot problem which requires data generated within a period of time and by a number of users to reach our goals. [Results/Conclusions] The test result shows the method we provide can use advantages of multiple MT engines and improve translation eventually.
Keywords: Multi-engine machine translation; Translation re-ranking; Translation combination

引言

针对特定领域、语种的机器翻译往往面临翻译质量不高的问题，为了在译后阶段提高翻译质量，常采用人工译后编辑、自动译后编辑以及译文重排序的方法。

机器翻译引擎的实现方法往往分为三类：基于规则的机器翻译（其中又包含基于转换的机器翻译、基于实例的机器翻译、基于中间语言的机器翻译等），基于统计的机器翻译系统以及神经机器翻译系统。不同的机器翻译引擎在面对具有特定领域、体裁、语种等特点翻译任务时，其翻译质量能够一定程度上达到人类水平，但从整体上来看依然距优秀的人工翻译有很大的差距。不同的机器翻译引擎在构建时使用的不同的方法、数据、架构会使得它们在面对具有不同原文特征的翻译任务时，其译文质量相应的不同。如 Banik[1] 认为，统计机器翻译具有更好的充分性，并且能够更好的处理稀有词，而相反的神经机器翻译能够更好处理句法结构从而生成流利的译文。如果能够同时利用多个不同的翻译引擎的优点，就能够提高译文的质量。李洪政 [2] 指出，在国际计算机协会 (ACL)2019 年举办的机器翻译大赛 (Conference on Machine Translation，WMT2019) 中，有一些团队选择了使用重排序的方法来提高其翻译效果，证明了译文重排序具有一定的有效性。本文提出了一种可以有效的利用不同的机器翻译引擎给出的不同译文，针对特定情况下的翻译任务对译文进行重排序，并自动的对这些译文进行融合，最终提升译文的质量的方法，该方法属于一种译文重排序结合译文融合的方法。

具体的来讲，本文提出方法首先对不同翻译引擎给出的译文进行质量评价，其次进行异同分析和融合处理。而国内外的常见多引擎翻译软件，如 QTranslate[3] 只是简单的提供了多个引擎的译文供参考，并没有对不同引擎的翻译结果进行重排序或是融合处理。本文研究首先构建了一个多引擎翻译系统，该系统为用户提供了六个不同的机器翻译引擎。其次，利用多引擎翻译系统在长期使用过程中所产生的用户数据、翻译记忆数据以及人工译文矫正数据并结合了用于搜索引擎的 PageRank 算法、贝叶斯公式和 UNQE 译文打分方法实现了多引擎译文重排序的方法。该方法可以针对不同的原文特征为多个不同的翻译引擎产生的译文进行自动打分，从而选出最优的机器译文，是一种结合了用户认知的方法。最后则利用系统产生的人

doi:10.3772/j.issn.2095-915x.2023.02.009

工译文数据结合 Transformer[4] 模型训练了一个自动译后编辑模型，进一步提升机了器译文的译文质量。最终这些组件结合起来形成多引擎翻译融合系统，实现端到端的翻译任务并提升译文质量。总的来说，本研究提出了：（1）一个多引擎翻译平台；（2）翻译模型训练资源库；（3）多引擎译文重排序方法；（4）自动译后融合模型。

1 相关工作

不论面对的是基于规则的、基于统计的、基于神经网络的机器翻译引擎，国内外都有许多相关研究都在尝试能够结合不同机器翻译引擎的优点，实现多机器翻译引擎、多机器翻译系统的融合。Matusov[5] 早在 2006 年就提出了使用对齐算法进行多引擎输出译文融合从而提升机器翻译的质量。Heafieldp[6]、Bangalore[7] 则分别使用了基于统计的方法对多引擎输出译文进行融合。Zhu[8]、Ma[9] 等则在句子级进行改写从而实现多翻译引擎的融合。除了句子级之外，也有研究者同时在句子、短语和词级别进行译文的融合，如 Freitag[10] 以及 Barrault[11] 分别提出了的开源的多机器翻译系统融合工具。

李响 [12] 使用串行结构结合规则翻译系统的输出和人工编辑后译文训练了一个基于统计的后编辑模型从而提升机器翻译的质量。宿建军 [13] 则在其构建的维汉机器翻译系统中利用最小贝叶斯方法并结合了基于短语的翻译模型、基于句法的翻译模型、基于层次短语的统计机器翻译模型，从而实现了句级、词语级以及联合式的系统融合，提升翻译的 BELU-SEP 评分。武静 [14] 在其构建的蒙汉机器翻译引擎中融合了多种方法来进行译文的重排序。

随着端到端（sequence to sequence）神经网络模型的广泛应用，有的研究者在编码器（Encoder）中融入了多种机器翻译方法。如 Zhou[15] 使用基于神经网络的、基于短语的以及基于分层短语的方法构建 encoder，并使用不同的 encoder 计算出不同的编码后向量并将这些向量利用注意力机制融合为最终的编码后向量供解码器使用（Decoder）。

为了要实现多机器翻译已经译文的重排序，需要对译文的质量进行量化打分。在机器翻译引擎训练的过程中，译文的质量往往由 BLEU(Bilingual Evaluation Understudy)、NIST(National Institute of standards and Technology)、METEOR 等可度量指标进行评价 [16-18]。这些评价指标使用字符匹配方式和参考译文进行机械的匹配，虽然具有与人工评价结果有一定的相关性，但往往并不能在句子的层面反映完全真实的翻译质量。而 QuEst 方法则通过对源语言句子复杂度、译文流利度、翻译忠实度、翻译系统置信度四个方面的评价，在句子层面对译文进行评价 [19]。Huang 等 [20] 则利用注意力机制让多个机器翻译结果与原文结合构建相似度矩阵，为多机器翻译结果中的每个单词打分，并根据打分结果向量交由 Decoder 从而将多个译文中重新组合成新的译文。

上述这些自动评价方法都需要基于现有的翻译译文作为参考，而 Li[21] 则基于 QE(句子级别翻译质量估计，Quality Estimation) 的基础上提出了 UNQE(the unified neural network for sentence-level QE tasks)，该模型使用了两个端到端

的 RNN (循环神经网络模型)，一个为双向 Encoder-Decoder RNN 模型、一个为专门计算 QE 得分的 RNN 模型，这个模型对翻译译文的评价增强了与人工评价的相关性，在译文 BLEU 值比较低的情况下，依然能够给出一个合理的译文评分。利用 UNQE 模型的重排序算法往往用于单一的机器翻译引擎在训练中 n-best 候选译文的重排序。

总来说，这些方法可以总结为：

（1）尝试将多个翻译方法融合起来，使用不同的方法处理翻译任务不同的步骤，再融合为最终的译文。

（2）使用一些特定的算法组合及规则对多个译文进行句子、短语、词级别的重排序和改写。

（3）使用联合训练的方法，将某些翻译引擎的输出作为其他方法的输入从而进一步训练出可以优化译文质量的译后编辑模型。

（4）使用不同的方法对译文质量进行评价。

上述这些研究都没有充分利用用户在使用多引擎机器翻译时所产生的数据。假设有一个能够提供多个不同翻译引擎译文并允许用户进行译后编辑的系统被用户长期使用，产生了海量的翻译实例数据，这些数据中蕴涵了大量的用户在使用该系统进行翻译时所产生的对这些机器译文的评价。如果能够有效的利用这些数据，显然能够在一定程度上计算出不同场景下不同翻译引擎的效果。

基于此，本研究结合了上述研究中的相应方法，结合大量用户在翻译实例中所做出的译文选择产生的翻译模型训练数据，对多引擎机器翻译系统生成的多译文进行排序和打分，从而进一步用于对多引擎译文进行融合处理。

2 方法详述

2.1 多引擎翻译平台架构

研究首先构建了一个多引擎翻译系统平台，平台包括：

（1）可扩展的多个翻译引擎，包含内部代号为 Y，X，H，K，T，F 六个翻译引擎；

（2）基于 BS 结构的 WEB 程序系统，其具体功能包含：用户可以使用浏览器上传需要翻译的原文文本、文本文件；系统会自动的将用户上传数据进行切割并逐句翻译，并为每一句翻译提供六个翻译引擎所翻译的结果；系统会为六个翻译引擎所翻译的结果进行质量打分和排序；用户可以自行选择使用哪一个译文；用户在选择译文后可以在该译文的基础上进行编辑，并最终确定翻译结果；系统能够自动记录用户的机器译文选择和人工译文以及相关数据。

（3）翻译模型训练资源库，系统在使用时，会将每一个翻译实例所产生的相关数据存储至翻译模型资源库中。这些数据包含：用户画像、原文数据、机器译文、人工译文。

多引擎翻译平台整体系统架构图如图 1 所示，当用户完成一个翻译实例时，系统会自动将该翻译实例所相关的数据存储在翻译模型训练资源库中维护。

该平台中的多引擎译文重排序模块架构如图 2 所示，当多引擎机器译文重排序模块会利用翻译模型训练资源库中的数据，自动产生引擎的相关打分，并基于此对多个引擎的输出译文进行排序。

doi:10.3772/j.issn.2095-915x.2023.02.009

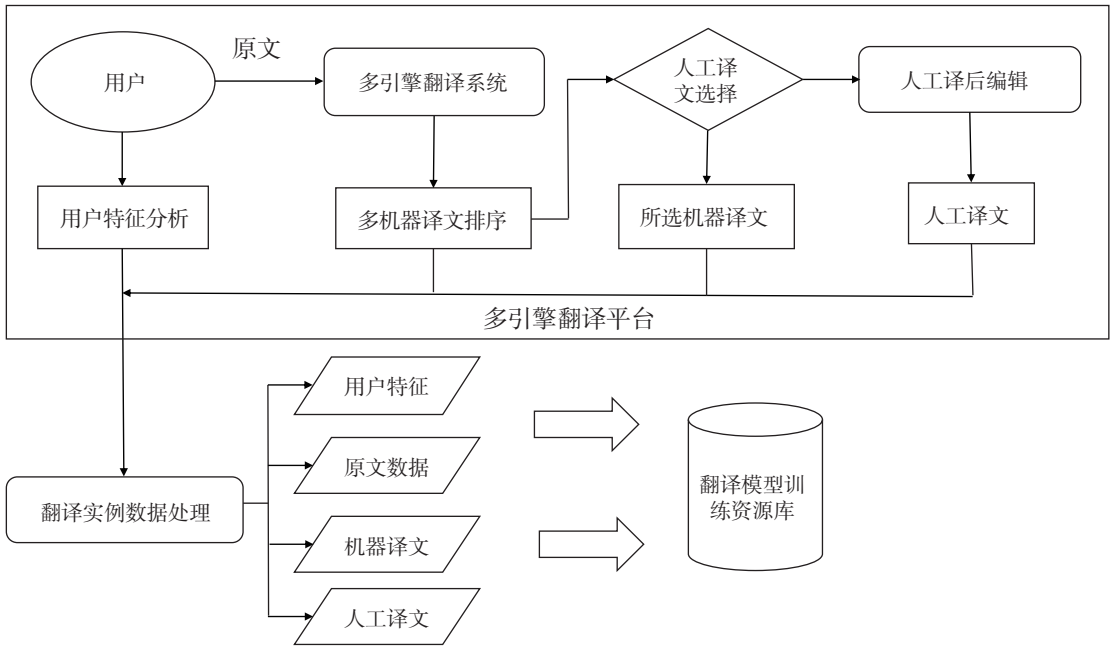

图 1　多引擎翻译平台架构图

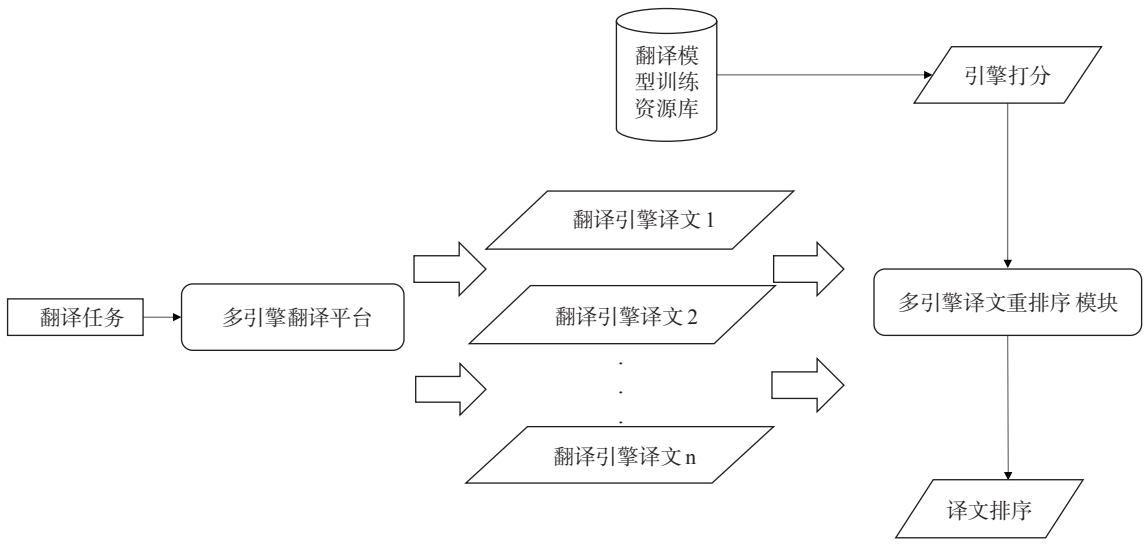

图 2　多引擎翻译译文重排序架构

2.2　翻译模型训练资源库

通过大量用户在使用多引擎翻译平台时产生的数据，我们会维护一个翻译模型训练资源库。该资源库在长期的使用中提供了大量的实际人工翻译实例，针对不同领域、体裁、述者类型不同的原文，不同的翻译引擎会有不同的表现，通过对翻译实例中不同类型的用户在特定时间段内所选的不同机器译文，可以使用贝叶斯公式得到机器译文被选择的先验概率。这个先验概率可以作为机器译文排序和打分的主要依据。具体的排序和打分算法将在多引擎译文重排序模型设计部分具体介绍。

翻译模型训练资源库中以翻译实例为核心，储存每一次某个用户在具体进行翻译时的原文特征数据、机器译文选择以及用户最终提供的人工译文。除此之外，用户在进行翻译的时候，平台会动态的为用户提供排序好的多引擎译文。库中还存储用户自身的特征数据，方便应用程序按照需要构建用户画像，从而进一步的可以从用户认知的角度对多个翻译引擎做出打分。翻译模型训练资源库的架构如图 3 所示。

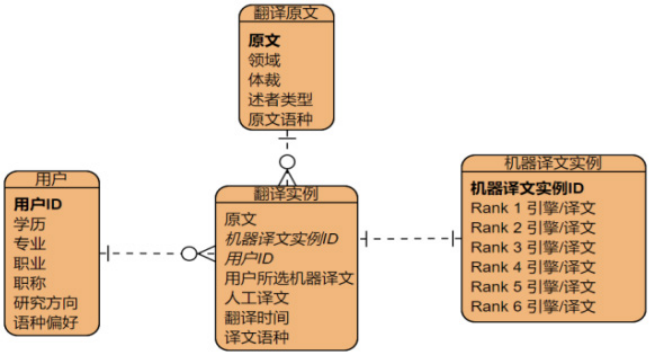

图 3　翻译模型训练资源库

这里对资源库的使用方法进行举例，假设需要从该资源库中找出所有 2019 年“经济”领域所有由“记者”进行的新闻报导的翻译实例数据中，不同研究方向、不同专业的用户所选的排序首位的引擎的数量，可使用以下 SQL 示例代码：

select count(i.Rank 1 引擎) as 数量， i.Rank 1 引擎 as from 翻译实例

where 翻译实例 e. 用户 ID = 用户 u. 用户 ID

and e. 原文 = 原文 s. 原文

and e. 机器译文实例 ID= 机器译文实例 i. 翻译模型训练资源库 ID

and e. 原文 = 原文 s. 原文 and s. 领域 =”经济”

and s. 体裁 =”新闻”

and s. 述者类型 =”记者” and e. 翻译时间

is between “2020-01-01” and “2018-12-31”

group by u. 专业， u. 研究方向 order by 数量；

在具体的使用中，可以针对性为用户设计若干个层级，根据用户特征的不同为他们所选的首选机器译文和人工译文打上权重，如我们可以将用户分为四类，其权重值设为：1、0.75、0.5、0.25，或是动态的根据用户画像计算其权重分数。具体的计算方法在节 2.3 中详细介绍。

2.3　多引擎译文重排序模型

本文提出了一个多引擎翻译排序实现方法，该方法的整体架构如图 4 所示。

该方法包括以下步骤：

第一步，使用翻译模型训练资源库中的机器译文和所选译文，基于 QE(句子级别翻译质量估计，Quality Estimation) 的方法训练一个 UNQE(the unified neural network for sentence-level QE tasks) 模型，该模型使用了两个端到端的 RNN (循环神经网络模型)，一个为双向 Encod-

doi:10.3772/j.issn.2095-915x.2023.02.009

er-Decoder RNN 模型、一个为专门计算 QE 得分的 RNN 模型，这个模型对翻译译文的评价增强了与人工评价的相关性，在译文 BLEU 值比较低的情况下，给出一个译文评分：S_{UNQE}。

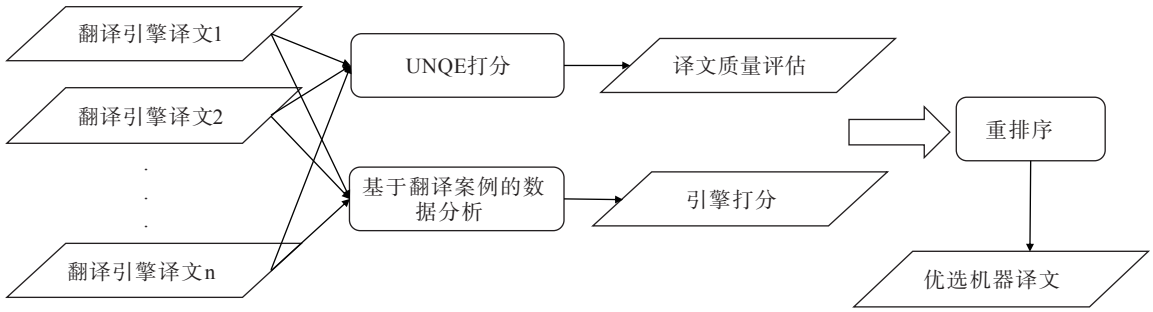

图 4　多引擎译文重排序模型

第二步，使用翻译模型训练资源库中的翻译案例的原文数据统计每一个不同前提条件出现的概率 $p(y)$，设 $p(y)$ 代表前提条件某一个具体的领域、体裁和述者类型出现的概率，具体计算公式为：

$$p(y)=count(domain，textType，authorType)/count(allTranslation) \quad (1)$$

第三步，计算每一个译文在该前提条件下的出现概率，其出现概率需要根据用户特征进行打分，本发明提出一个多引擎翻译打分排序方 法 (Multi-Engine Machine Translation Rank，MEMTRank)；根据用户的特征为每一个用户进行权重计算，每一个用户的权重 $user_{score}$ 的具体计算公式为：

$$S_{user}=S_{digree}+S_{Ocuupation}+S_{major}+S_{distance(language)}+S_{distance(domain)} \quad (2)$$

其中每一项打分都为整数 1 到 5 之间，各项详述分别为：

（1）S_{digree}：学历打分，不同的用户根据学历的不同获得不同的打分。

（2）$S_{Ocuupation}$：职业和职称打分，根据用户的职业与职称进行加权打分。

（3）S_{major}：专业打分，专业为翻译，语言类专业用户打分相对要高 .

（4）$S_{distance(language)}$: 语种距离打分，将用户所掌握的语种与待翻译的原文及目标语种进行匹配，若匹配则打分高 .

（5）$S_{distance(domain)}$：用户研究方向和译文领域的语义距离，计算方法为计算研究方向词和译文领域词的欧式距离：

$$S_{distance(domain)}=abs(\overrightarrow{direction}-\overrightarrow{domain}) \quad (3)$$

第四步，在计算得到 S_{user} 后，针对每一个前提条件，计算每一个翻译引擎的译文被用户选为首选引擎的概率 $p(x)$，为了避免某个具体用户进行的翻译实例过多，导致整个打分模型的结果出现严重的偏差，其计算方法借鉴了谷歌用于搜索引擎的 PageRank[20] 算法，将每一个用户计算所得分数加权分配给其历史翻译案例中所选引擎，具体算法如下：

（1）对翻译模型训练资源库中的数据进行统计，得到用户 u_1 到 u_n 所有用户所选的机器译文引擎数量 $count(u_n)$，之后将根据每一个引擎

被该用户所选的数量计算比率 $count(e_iu_i)$，并计算比率：

$$w(e_iu_i)=count(e_iu_i)/count(u_n) \tag{4}$$

（2）将每一个用户 u_i 的用户打分 S_{u_i} 按比率分配给其历史数据中所选的每一个翻译引擎 e_i，得到引擎打分 S_{e_i}：

$$S_{e_i}=\sum_{j=1}^{j=n}S_{u_j}*w(e_iu_j) \tag{5}$$

（3）最终计算具体翻译引擎 e_i 在前提条件 $p(y)$ 的情况下出现的概率 $p(x)$:

$$p(x)=\frac{S_{e_i}}{\sum_{j=1}^{j=n}S_{e_j}} \tag{6}$$

第五步，计算针对所有翻译实例中翻译引擎 e_i，被所有用户首选为机器译文的情况下，特定前提条件出现的概率 $p(y|x)$，其计算方法为:

$$p(y|x)=\frac{count_{e_i}(domain,textType,authorType)}{\sum_{j=1}^{j=n}count(e_iu_j)} \tag{7}$$

第六步，使用贝叶斯公式计算在特定领域、体裁、述者类型的情况下，每一个翻译引擎的 MEMTRank(Multi-Engine Machine Translation Rank，多引擎机器翻译排序) 的相应分数：

$$S_{MEMTRank}=p(x|y)=p(y|x)*\frac{p(x)}{p(y)} \tag{8}$$

最终，计算各引擎译文最后得分，并根据得分进行排序：

$$S_{e_iRank}=S_{MEMTRank}*S_{UNQE} \tag{9}$$

$$Rank=sort_{DESC}\left(\bigcup_{i=1}^{i=n}S_{e_iRank}\right) \tag{10}$$

该方法使用了大量的用户使用中产生的历史数据，能够为用户提供多个机器译文的重排序结果，方便用户选择更好的译文。该多引擎翻译排序实现方法在基于经验主义的多翻译引擎中融入了用户的理性判断，用户使用所产生的翻译模型训练资源库中蕴含了大量的基于用户认知的数据，对这些数据的使用就是在整体上使用了用户在长期使用过长中对不同翻译引擎的评价，从而能够对方法进行持续的优化，增强其对多引擎译文打分和重排序的能力。同时算法还借鉴了 Page Rank 算法用于平滑不同用户使用情况，避免出现因个别具有大量数据的用户所造成的数据异常情况。整个多引擎重排序模型的训练方法如图 5 所示。

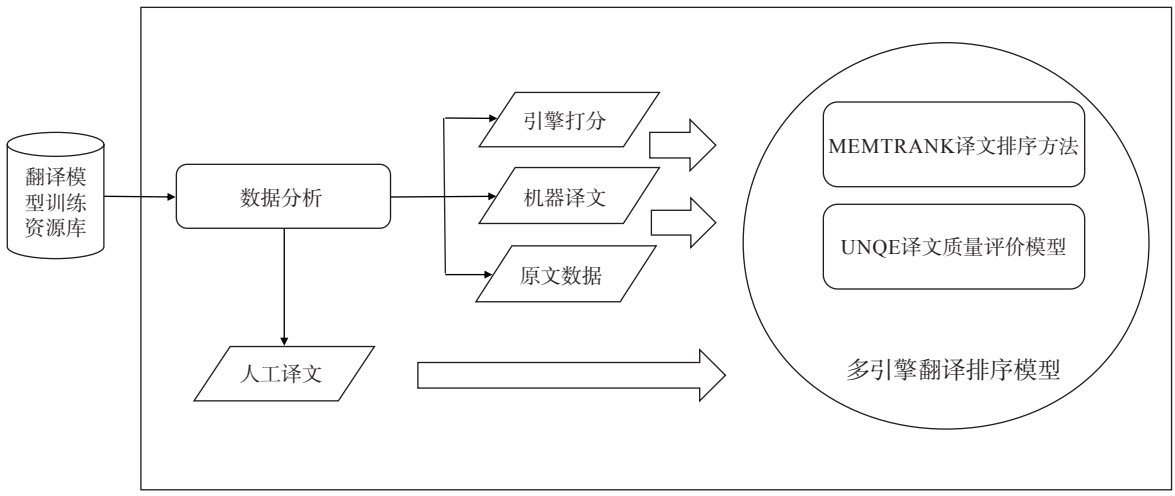

图 5　多引擎译文重排序训练模块

doi:10.3772/j.issn.2095-915x.2023.02.009

2.4 译文融合模型

在具有六个翻译引擎的多引擎翻译平台对机器译文进行重排序之后，平台将这些排序后的机器译文以及翻译任务中的原文特征数据作为输入，使用自动译后融合模型得到第七个机器译文。自动译后融合模型同样使用翻译模型训练资源库中的用户翻译实例产生的相关数据进行训练，其架构如图 6 所示。

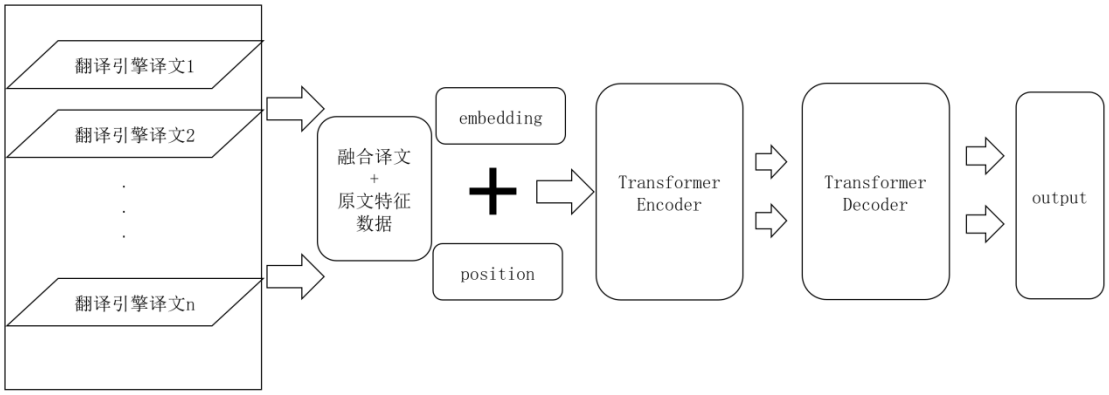

图 6 译文融合模块架构

可以看到，模型的输入是一个经过排序并添加了原文特征的文字序列，并对每一个字符都进行 word embedding 后与 position embedding 相加，而模型将多个翻译引擎译文融合为一个机器译文，这是一种典型的序列到序列（Seq2Seq）的任务，因此我们使用谷歌提出的基于自注意力机制的 Transformer 模型。在处理输入时，为了解决输入序列过长的问题，我们提供了一个输入预处理器，将译文相似的部分与不同的部分标记出来，提前融合为一个句子后再加入原文特征数据于句子结尾。预处理部分会根据译文重排序结果将排序靠前的译文选择置于前方，最终再使用人工译文训练出的译文融合模型得到最终结果，其效果示例如表 1 所示。

表 1 译文融合预处理

原文	Furthermore, we use the result generated by the re-ranking method with human translations provided by the resource warehouse to train a translation combination model.
RANK1引擎译文	此外，我们使用重新排序方法生成的结果和资源仓库提供的人工译文来训练翻译组合模型。
RANK2引擎译文	此外，我们使用资源仓库提供的人工翻译和重排序方法生成的结果来训练翻译融合模型。
RANK3引擎译文	此外，我们使用重新排序方法生成的结果和资源仓库提供的人工翻译来训练翻译组合模型。
RANK4引擎译文	此外，我们使用重新排序方法生成的结果以及资源仓库提供的人工译文来训练翻译组合模型。
RANK5引擎译文	此外，我们使用由资源仓库提供的人工翻译和重新排序方法生成的结果来训练翻译组合模型。
RANK6引擎译文	此外，我们利用由资源仓库提供的人工翻译的重新排序方法生成的结果来训练翻译组合模型。
融合译文	[START]此外，我们使用[重新排序方法生成的结果和资源仓库提供的人工[译文/翻译]/由资源仓库提供的人工翻译[和/的]重新排序方法生成的结果]来训练翻译[组合/融合]模型[END]
输出结果	此外，我们使用重新排序方法生成的结果和资源仓库提供的人工译文来训练翻译组合模型。
人工译文	此外，我们使用重新排序方法所生成的结果以及资源仓库提供的人工译文来训练翻译融合模型。

最终的训练使用了开源框架 fairseq[22] 中的 Transformer Encoder 和 Transformer Decoder 以及相应的 Seq2Seq 模型，并设置 batch size 为 750 句、学习率和 dropout 分别为 0.001 和 0.3、前馈神经网络的维度为 1 024，注意力头的个数为 4，使用 Adam 优化器以及逆平方根学习率衰减公式。

3 系统测试与实验结果分析

3.1 系统测试

本研究所提方法的一个重要前提是需要拥有一定数量以上的高质量用户数据，所设计的平台首先需要在经过一定时间的使用，收集大量用以构建翻译模型训练资源库的数据后，才能够进行因为重排序和融合模型的训练。在多引擎翻译平台上线后，翻译实例数据由系统用户在约 10 个月的使用中产生，并组织过数次针对特定新闻领域的翻译实战演练。截至系统测试之前，本研究所架设系统收集了超过 1 000 个用户以及 40 万个翻译实例数据用于重排序和融合模型的训练，其中新闻报道类型翻译实例数量最多，共收集约 30 万翻译实例，而其中军事领域共约 20 万个翻译实例。

本研究选用五个来自不动不同领域各 1 000 句经过人工校对英汉翻译实例对系统进行测试，使用模拟用户数据分别训练多引擎机器译文重排序模型和自动译文融合模型。对比多引擎翻译平台的六个翻译引擎、使用多引擎机器译文重排序方法所提供的最优机器译文以及译后融合模型所提供的译文。除此之外，本研究还尝试使用了在 transformer 架构 encoder 端构建相似度编码矩阵（简称：Similarity Model）的机器翻译模型、以及多方法结合的机器翻译模型（简称：NSC，Neural System Combination），在上述翻译实例中进行测试。最终的 BLEU 值测试结果如表 2 所示。

表 2 翻译测试 BLEU 值

翻译引擎/任务	军事领域新闻报道	经济领域新闻报道	经济领域百科词条	Ted Talk精选	书籍领域	平均值
Y	37.8	46.7	**49.6**	36.2	38.8	41.8
X	39.6	43.2	45.9	37.3	33.9	40.0
H	**47.6**	40.2	40.1	31.9	33.8	38.7
K	35.6	33.2	36.5	43.7	34.1	36.6
T	36.5	37.1	38.6	45.2	41.1	39.7
F	38.9	33.6	35.4	32.3	44.0	36.8
UNQE	41.2	38.6	42.1	38.6	36.9	39.7
Similarity Model	41.8	43.1	42.6	45.1	**46.7**	43.9
NSC	42.2	43.9	45.6	**46.8**	43.9	44.6
UNQE+MEMTRank	45.3	47.1	45.2	43.9	44.5	45.3
Reranking+译后融合模型	**48.7**	**47.8**	47.6	45.9	46.3	**45.9**

3.2 实验结果分析

从测试结果可以明显的看出在神经网络模型中融入多方法、多引擎能够有效提高机器翻译的平均质量，并在跨领域的翻译场景中减少单一引擎面对特定领域翻译质量不足的问题。

doi:10.3772/j.issn.2095-915x.2023.02.009

使用本文所提 MEMTRank 打分的重排序模型，从而计算出的的优选机器译文能够在不同的场景下都稳定的表现出相对较高的译文质量。相比于任意单个机器翻译引擎以及基于 UNQE 的译文重排序方法，其在多领域的平均译文质量有明显的提高。其次，使用大量用户提供的人工译文训练的译后自动融合模型，翻译结果的 BLEU 值也有了进一步的提高。

与其他相关多引擎系统融合的方法相比，本方法在新闻类型的翻译任务中，以及军事、经济领域的翻译任务中取得了更高的BELU值。而在翻译实例数据资源有限的其他领域则未能显著优于其他相关方法。因此，充足的翻译实例数据，是本方法能够有效提高翻译质量的重要前提。

总的来说，其他翻译系统融合的方法寻求结构和方法上的改变，本文所提方法可以能够在不改变其他翻译引擎内部结构的情况下将其融合到本文所提方法中。现有的商用翻译平台和引擎往往不需要注册，难以收集翻译用户画像，也无法方便的获取用户对翻译结果的反馈。本文所提的翻译系统平台则能够方便的获取上述信息，从而构建翻译训练资源库。在拥有充足的用户，并收集到足够数量翻译数据实例的前提下，本文所提方法能够高效的利用已有的不同机器翻译引擎，提高多领域、多类型的翻译任务译文质量。

4 结语

本文研究构建了一个多引擎翻译平台，并利用平台在长期使用产生的相关数据，实现了多引擎机器翻译译文重排序方法，以及自动译文融合模型并提高了最终的翻译译文质量。该方法使用用户在使用多引擎机器翻译平台所产生的数据，提取了蕴涵在大量用户对特定场景下不同引擎效果的评价，又使用了大量用户提供的人工译文训练了译文融合模型，从而利用了不同机器翻译引擎在不同场景下的优势，提高了机器翻译译文的质量。为了避免某个单独用户可能产生过多的数据会对整体方法的效果造成影响，多引擎译文重排序的方法借鉴了谷歌用于网页打分的 Page Rank 算法的思想。实验结果表明，多引擎系统融合的方法能够有效提高翻译质量。与其他系统融合的方法相比，本文提出方法是针对多机器翻译引擎输出的重排序和融合方法，不追求改变各个引擎的内部结构。该方法充分利用了现有机器翻译引擎，如果能够广泛应用，拥有降低不断重新训练新的机器翻译引擎所带来了资源消耗的可能性。

与过往方法不同，本文所提方法需要大量的用户实例数据，尽管本文所构建多引擎机器翻译平台选用了市面上主流的六个大型商用机器翻译引擎，译文重排序和融合模型的训练依然局限于本研究内部平台所产生的数据并且难以快速应用于其他多引擎翻译平台。针对当下大量广泛应用的大型开源神经机器翻译平台以及 API，如何高效的获取世界范围内的相关用户使用数据问题仍然需要解决。

未来，研究将关注于本文所提方法在多语种翻译任务上有效性验证，并持续观察在更长时间段，更大数据量基础上的使用中，该方法的效果。从理论上来讲，当多引擎翻译平台具有更多的用户以及相关数据会对其效果产生的

影响。另外，将多个译文进行融合的必要性也有待于进一步的对比验证，如对比仅使用多引擎机器译文重排序方法所提供的最优机器译文和人工译文训练自动译后编辑模型。

参 考 文 献

[1] BANIK D, EKBAL A, BHATTACHARYYA P, et al. Assembling translations from multi-engine machine translation outputs [J]. Applied Soft Computing, 2019, 78.

[2] 李洪政 , 冯冲 , 黄河燕 . 稀缺资源语言神经网络机器翻译研究综述 [J]. 自动化学报 , 2021, 47(6):15.

[3] ZHANG X W. Research on the QTranslate secondary platform for business exchange in real-time business english system [C]// 2017.

[4] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [J]. arXiv, 2017.

[5] MATUSOV E. Computing consensus translation from multiple machine translation systems using enhanced hypotheses alignment [C]// EACL 2006, 11st Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, April 3-7, 2006, Trento, Italy. DBLP, 2006.

[6] HEAFIELD K, LAVIE A. Combining machine translation output with open source: the Carnegie Mellon Multi-Engine Machine Translation Scheme [J]. Prague Bulletin of Mathematical Linguistics, 2010, 93(-1): 27-36.

[7] BANGALORE B, BORDEL G, RICCARDI G. Computing consensus translation from multiple machine translation systems [C]// IEEE Workshop on Automatic Speech Recognition and Understanding, 2001.

[8] ZHU J, YANG M, SHENG L, et al. Sentence-Level paraphrasing for machine translation system combination [J]. 国际计算机前沿大会会议论文集 , 2016, 000(001): 156-158.

[9] MA W Y, MCKEOWN K. System combination for machine translation through paraphrasing [C]// Conference on Empirical Methods in Natural Language Processing, 2015.

[10] FREITAG M, HUCK M, NEY H. Jane: Open source machine translation system combination [J]. 2014.

[11] BARRAULT L. MANY: Open source machine translation system combination [J]. Prague Bulletin of Mathematical Linguistics, 2010, 93(-1): 147-155.

[12] 李响 , 胡小鹏 , 袁琦 . 面向多引擎融合技术的统计后编辑方法研究 [J]. 工业技术创新 , 2015, 2(6): 591-596.

[13] 宿建军 , 张小燕 , 吐尔洪·吾司曼 , 等 . 联合式多引擎维汉机器翻译系统 [J]. 计算机工程 , 2011, 37(16):179-181.

[14] 武静 . 多方法融合蒙汉机器翻译与译文重排序研究 [D]. 内蒙古大学 , 2017.

[15] ZHOU L, HU W, ZHANG J, et al. Neural system combination for machine translation [C]// ACL 2017.

[16] PAPINENI K, ROUKOS S, WARD T, et al. BLEU: a Method for automatic evaluation of machine translation[J]. ACL proceedings of annual meeting of the association for computational linguistics, 2002.

[17] QIN Y, WEN Q, WANG J. Automatic evaluation of translation quality using expanded N-gram co-occurrence [C]// IEEE, 2009.

[18] DENKOWSKI M, LAVIE A. Meteor Universal: Language Specific Translation Evaluation for Any Target Language [C]// Proceedings of the Ninth Workshop on Statistical Machine Translation. 2014.

[19] SPECIA L, SHAH K, CAMARGO J G, et al. Quest-A translation quality estimation framework [C]// ACL. 2013.

[19] LI M, XIANG Q, CHEN Z, et al. A unified neural network for quality estimation of machine translation [J]. IEICE Transactions on Information and Systems, 2018, 101(9): 2417-2421.

[20] HUANG, X, ZHANG J, TAN Z, et al. Modeling voting for system combination in machine translation [C] // IJCAI 2020.

[21] PAGE L. The PageRank citation ranking: Bringing order to the Web [J]. Proceedings of ASIS, 1998, 98: 161-172.

[22] OTT M, EDUNOV S, BAEVSKI A, et al. Fairseq: A fast, extensible toolkit for sequence modeling [J]. Proceedings of the 2019 Conference of the North, 2019.