第十六届机器翻译研讨会
厦门大学评测报告

张国成，王颖敏，钟恩俊，江秋怡，江舫，
章栋，朱宏康，陈毅东 *，史晓东
（厦门大学信息学院人工智能系，福建省厦门市，361005）
* 通信作者：ydchen@xmu.edu.cn

摘要： 本文介绍了厦门大学参加第十六届全国机器翻译研讨会的汉英平行语料过滤任务评测的参评系统情况。在此次评测中，本文参评系统主要利用规则方法对噪音句对进行严格的过滤；同时也设计了五种启发式方法，从不同侧重点对噪音句对平行程度进行度量，尤其是基于单词的编辑距离和基于双语预训练模型的马氏距离在过滤优质平行数据上有良好的表现；最后，我们对表现优异的方法，按照加法和乘法两种方式进行加权融合。最终，本文提交的系统综合排名第二。
关键词：机器翻译；语料过滤；预训练模型
中图分类号：TP391　　**文献标识码：**A

XMU Evaluation Report for CCMT2020

Guocheng Zhang, Yingmin Wang, Enjun Zhong, Qiuyi Jiang, Fang Jiang,
Dong Zhang, Hongkang Zhu, Yidong Chen*, Xiaodong Shi
(Department of Artificial Intelligence, School of Informatics, Xiamen University, Xiamen 361005, Fujian, China)
* Corresponding author: ydchen@xmu.edu.cn

Abstract: This paper introduces the situation of XMU participating in the task of Chinese-English parallel corpus filtering in the 16th China Conference on Machine Translation. In this evaluation, we introduce a rule-based method to filter noisy sentences in a harsh way; meanwhile, we also design five heuristic methods to measure the degree of parallelism between noisy sentences from different focuses. Especially the token-based Levenshtein distance and Mahalanobis distance based on bilingual pre-trained model have a good performance in selecting high-quality parallel data. Finally, we perform weighted fusion for the methods that perform well in two ways: addition and multiplication. In the end, the system submitted in this paper ranked second overall.
Keywords: Machine Translation; Corpus Filtering; Pre-trained model

1. 引言

本文对厦门大学自然语言处理实验室参加第十六届机器翻译研讨会(CCMT2020)评测任务的情况进行了描述。本实验室参加了汉英平行语料过滤任务。

随着神经机器翻译的兴起和发展，语料过滤是当下研究的热点，WMT2018[1]和WMT2019[2]均举办过语料过滤任务。目前主流的神经机器翻译系统需要大量的语料进行模型训练，语料的质量很大程度影响了翻译模型的质量[3]，所以通过对语料进行过滤来维护语料质量尤为重要。目前，语料过滤的方法并没有统一的模式，不过主流的方法一般是采用基于规则的方法和启发式方法相结合的策略，其中启发式方法非常多，比如 Junczys-Dowmunt 等人提出对偶条件交叉熵(Dual Conditional Cross-Entropy)方法[4]，S´anchez-Cartagena 等人则尝试从各种打分函数中学习权重[5]，还有不少学者试图从词嵌入(Word Embedding)[6-8]角度衡量句对平行程度。CCMT2020 语料过滤任务旨在解决清理带噪音的汉英平行语料问题。本文设计了两种不同的模式，分别是单系统模式和多系统融合模式，其中单系统分为规则系统、Zipporah 系统、词对齐系统、翻译模型系统、语言模型系统和双语预训练模型系统。多系统融合则是在单系统的基础上，对各个系统的打分加权融合，再重新排序，进而从语料库中过滤得到高质量的平行句对。最后，使用 CCMT2020 提供的机器翻译工具 Marian[1]训练神经机器翻译(NMT)系统，计算开发集上的翻译结果与参考译文之间的 BLEU 值。最终，本文提交的系统综合排名第二，在 WMT2020 News、WMT2020 Biomedical 等多个数据集上排名第一。

2. 系统描述

本文主要采用规则方法和启发式方法来构造语料过滤系统，其中规则方法又可细分为长度过滤、长度比过滤、语种过滤和去重四个类别；启发式方法有 5 个，分别为基于 Zipporah 的方法、基于词对齐的方法、基于翻译模型的方法、基于语言模型的方法和基于双语预训练模型的方法。最终提交的 2 个系统则是上述方法的融合。

[1] https://github.com/marian-nmt/marian

2.1 基于规则的方法

Pinnis 提出利用句子长度比例、最大句子长度、唯一句子对等过滤方法[9]对语料进行过滤。借鉴其工作，本文制定了 4 条规则，第一条是长度过滤，源端或目标度句子长度超过 80 个单词的句对记 0 分，否则记 1 分；第二条是长度比限制，源端与目标端(或目标端与源端)句子长度比超过 1.7 的句对记 0 分，否则记 1 分；第三条是语种识别，用 python 的 langid[2] 识别源端和目标端语种，语种不对的句对记 0 分，否则记 1 分；第四条是去重，重复的句对第一次出现记 1 分，否则记 0 分。这样可以得到一个四维特征，每一维的值为 0 或 1。

2.2 基于 Zipporah 的方法

Chaudhary 等人尝试将 Zipporah[3]作为融合系统的一部分[10]，取得了不错的成绩。Zipporah 是一种快速且可扩展的系统，可以从大量嘈杂的数据池中选择任意大小的好数据，用于神经机器翻译模型的训练。其原理是将句子映射到特征空间，特征空间包含两个特征 Adequacy scores 和 Fluency scores，之后使用逻辑回归进行二分类，类别分别是好数据和坏数据。最后采用公式(1)进行分数归一化，其中$\boldsymbol{x}$为 Zipporah 的得分。

$$\boldsymbol{score(x)} = \frac{\boldsymbol{x}-\boldsymbol{x_{min}}}{\boldsymbol{x_{max}}-\boldsymbol{x_{min}}} \tag{1}$$

2.3 基于词对齐的方法

Zarina 等人认为非平行的句对的词对齐很少[11]，于是可以考虑利用词对齐进行过滤。我们首先用 fast_align[4]词对齐工具在 CCMT2020 提供的不带噪音的中英平行语料上训练，然后对带噪音的语料进行预测，可直接得到句子对的词对齐分数。由于在 fast_align 工具中，词对齐分数的计算方法是将词对齐概率进行对数求和，显然，句子越长，分数越小，意味着系统偏好短句子。我们采用公式(2)减少句子长度对分数的影响，其中$\boldsymbol{score_{align}}$指句子对的词对齐分数，$\boldsymbol{l_{source}}$和$\boldsymbol{l_{target}}$分别表示源端和目标端句子的长度。

$$score_{avg} = \frac{2*score_{align}}{l_{source}+l_{target}} \tag{2}$$

[2] https://github.com/saffsd/langid.py
[3] https://github.com/hainan-xv/zipporah
[4] https://github.com/clab/fast_align

在将句子对的词对齐分数按照公式(2)处理后，我们按照分数从高到低进行排序，经过统计发现词对齐分数大于等于-4.5 的句子对数量约为 400 万，大约 1 亿个单词。理论上我们认为这些句子对的质量较好，它们在归一化后的分数应该较高，于是我们设计了公式(3)进行分数的归一化，其中$score_{avg}$是公式(2)计算后的分数。

$$score = \frac{-4.5}{score_{avg}-4.5} \quad (3)$$

2.4 基于翻译模型的方法

Parcheta 等人尝试对非源端句子进行翻译，然后计算译文与参考之间的相似度[12]。该方法的设想是：如果句子a与句子b是平行句对，那么a与b的语义相似，则将a翻译成a'时，a'与b的语义仍然相似。在此次评测中，我们将a与b分别看作英文和中文句子。如果a与b是平行的，则a'与b之间的相似度较高，否则相似度较低。

为实现上述设想，首先应训练一个英-中翻译模型，然后利用翻译模型将英文句子翻译成对应译文，最后计算译文与参考之间的相似性。对于相似度计算，本文采用了两种指标：一种是基于单词的编辑距离(Levenshtein Distance)，另一种是基于预训练词向量的余弦相似度(Cosine Distance)，形成 2 维的相似度特征，用于度量原始句对的平行程度。

2.4.1 翻译模型

根据上述简介，若想计算译文与参考之间的相似度，首先应得到译文，因此需要训练一个翻译模型。本文采用了清华大学开源的神经机器翻译工具包 THUMT[5]，该系统依赖较少，训练简便，适合快速训练神经机器翻译系统。

训练集数据来源于 CCMT2020 汉英翻译任务提供的平行语料，我们对其进行分词和小写化，并过滤掉长度超过 150 个单词的句对，形成约 1 000 万对的训练数据。开发集为 CCMT2020 语料过滤任务指定的 WMT 开发集。

主要的训练参数选择默认，并运行约 20 轮，保存开发集上 BLEU 值最高的 5 个模型，然后做模型平均，融合成一个最终模型，方向为英-中，将其记为M_0。紧接着，利用M_0对带噪音的平行句对中的英文句子进行解码，得到对应的中文译文，这里由于硬件设备有限，解码前先将英文句子进行了一些预处理，具体包括：将过长的句子

[5] https://github.com/THUNLP-MT/THUMT.git

截断，只保留前 150 个单词，以防止显存溢出；将带噪音的数据切分成多个小文件，每个文件包含 200 万平行句对，以防止内存溢出。

2.4.2 基于单词的编辑距离

该指标本质上是编辑距离，不过计算两个句子匹配程度的粒度为单词，而不是单个字符。设a'与b为两个分词后的中文句子，其中a'为英文源句a的译文，那么编辑距离$Lev_{a',b}(|a'|, |b|)$可以通过公式(4)计算：

$$Lev_{a',b}(i,j) = \begin{cases} max(i,j) & if \ \ min(i,j) = 0, \\ min \begin{cases} Lev_{a',b}(i-1,j) + 1 \\ Lev_{a',b}(i,j-1) + 1 \\ Lev_{a',b}(i-1,j-1) + 1_{(a_i' \neq b_j)} \end{cases} & otherwise. \end{cases} \tag{4}$$

其中$|a'|$和$|b|$分别指a'和b两个句子的单词数，$Lev_{a',b}(i,j)$定义为a'句前i个单词和b句前j个单词之间的距离。

在计算过程中，如果译文看作a'，参考看作b，实际上在带噪音的数据中，作为目标端的b不一定与源端相对应。如果不与源端对应，则a'和b距离较大，意味着我们更愿意相信M_0给出的判断，认为此句对平行程度较差，反之距离较小则意味着翻译模型给出的译文和实际参考之间相似度较高，则源句与目标端句子的平行程度较高，所以根据编辑距离，最终句对的平行程度得分如公式(5)所示：

$$score_{a,b} = 1 - Lev_{a',b}(|a'|,|b|) \tag{5}$$

2.4.3 余弦相似度距离

由于翻译模型M_0可以将英文源句a翻译成对应中文译文a'，因此可以借助中文词向量计算a'和参考b之间的语义相似度，用于衡量源端与目标端句子的平行程度。之所以不用中文和英文两套单独的词向量，是因为担心语种差异会造成语义空间的偏差，导致语义相似度计算不准确。

训练中文词向量用到的数据与机器翻译训练集中的中文端数据相同，在此不再赘述。训练工具采用的是 gensim[6]工具包，训练窗口取 5，去掉词频低于 5 的词，并且考虑到相似计算压力较大，因此维度取 128 维，训练 10 轮，最终保存模型记为M_1。

对于a和b句对，a'是a的中文译文，那么利用M_1即可得到该句对的平行程度得分，如公式(6)所示

[6] https://radimrehurek.com/gensim/models/word2vec.html

$$score_{a,b} = cosine(a', b|M_1) \tag{6}$$

需要注意的是，在计算余弦相似度时，我们设置了中文的停用词表，去掉了诸如“了”、“的”等常见且实际意义较弱的词，防止造成该句与其他任意句相似程度均较高的问题。

2.5 基于语言模型的方法

因为语言模型可以过滤掉不合语法的数据，所以我们考虑使用语言模型对语料进行过滤。由于在任务描述中，参赛队伍被特别要求不能使用与领域相关的度量指标。因此，本文选择基于不带噪音的语料库生成语言模型，并利用该语言模型计算待过滤数据集的困惑度(Perplexity, ppl)分数。

具体地，我们在不带噪音的双语语料上使用 SRILM[7]工具，为中英文语料分别训练一个 5-gram 的语言模型，并使用这个语言模型分别计算待过滤双语语料中中英句子的困惑度分数。对于得到的中英句子困惑度分数，本文使用了两个打分策略：分别是句子级困惑度分数和单词级困惑度分数，最终每个平行句对将得到四个特征分数。

为了便于后续处理，我们将困惑度分数进行归一化处理。在归一化操作中，我们基于经验设计了一系列分段函数，如果无特别说明，下面式子中的ppl均指语言模型计算得到的困惑度分数。

对中文待过滤语料句子级困惑度分数，设计的归一化的分段函数如(7)所示：

$$score(x) = \begin{cases} 1 - 0.001 * ppl & if \ \ ppl \leq 100 \\ 1 - \left((ppl - 100) * \frac{0.4}{900} + 0.1\right) & if \ \ 100 < ppl \leq 1000 \\ 1 - \left((ppl - 1000) * \frac{0.3}{9000} + 0.5\right) & if \ \ 1000 < ppl \leq 10000 \\ 0.1 & if \ \ ppl > 10000 \end{cases} \tag{7}$$

对英文待过滤语料句子级困惑度分数，设计的归一化分段函数如(8)所示：

$$score(x) = \begin{cases} 1 - 0.003 * ppl & if \ \ ppl \leq 100 \\ 1 - \left((ppl - 100) * \frac{0.4}{900} + 0.3\right) & if \ \ 100 < ppl \leq 1000 \\ 1 - \left((ppl - 1000) * \frac{0.15}{9000} + 0.7\right) & if \ \ 1000 < ppl \leq 10000 \\ 0.1 & if \ \ ppl > 10000 \end{cases} \tag{8}$$

[7] https://github.com/BitSpeech/SRILM

另外我们考虑了单词级的困惑度分数，分别计算了中英数据集上每句的词平均困惑度分数与整体数据集上的词平均困惑度分数，并设计了两个分段函数对两者的差值进行归一化处理。由于数据中存在句子很短，但困惑度值非常大的现象，因此这里计算整体数据集的词平均困惑度分数的时候，忽略了困惑度超过 1 万的句子。

对中文待过滤语料单词级困惑度分数，设计的归一化分段函数如(9)所示：

$$score(x)=\begin{cases}1 & if\ \ ppl\le 0\\ 1-ppl*0.003 & if\ \ 0<ppl\le 100\\ 1-\left((ppl-100)*\frac{0.6}{900}+0.3\right) & if\ \ 100<ppl\le 1000\\ 0.01 & if\ \ ppl>1000\end{cases} \tag{9}$$

对英文待过滤语料单词级困惑度分数，设计的归一化分段函数如(10)所示：

$$score(x)=\begin{cases}1 & if\ \ ppl\le 0\\ 1-ppl*0.002 & if\ \ 0<ppl\le 100\\ 1-\left((ppl-100)*\frac{0.7}{900}+0.2\right) & if\ \ 100<ppl\le 1000\\ 0.01 & if\ \ ppl>1000\end{cases} \tag{10}$$

2.6 基于双语预训练模型的方法

考虑到预训练模型包含大量的语义知识，因此我们利用 Reimers 等人提出的 Sentence-BERT(SBERT)模型[13]在官方给定的中英单语语料上进行微调，分别获得中英文的句向量。但是通过该方式获得的句向量可能存在语言之间的向量空间未对齐的问题，即不同语言中意义相同的句子被映射到向量空间中的不同位置。因此评估两个不同语言的句子之间的平行度时，我们采用马氏距离平方之比作为度量指标。

马氏距离表示数据的协方差距离，是一种计算两个未知样本集相似度的有效方法。使用马氏距离，就等同于通过数据转换的方法，消除样本中不同特征维度间的相关性和量纲差异，使得欧式距离在新的分布上能有效度量样本到分布间的距离。假设向量 x 服从于均值$\boldsymbol{\mu}$，协方差为$\boldsymbol{\Sigma}$的 X 分布，则其到中心的马氏距离计算公式如(11)所示：

$$d^2(x)=(x-\mu)^T\Sigma^{-1}(x-\mu)=\left\|\Sigma^{-\frac{1}{2}}(x-\mu)\right\|_2^2 \tag{11}$$

在本文系统中，首先我们将每个句向量进行标准化，使得其服从均值为 0 的分布。对于每个已经重新中心化中英文句子向量对$<l_1,l_2>$，我们考虑变化空间中的三种情况：

$$e_1 = \Sigma^{-\frac{1}{2}}(l_1, \vec{0}) \tag{12}$$

$$e_2 = \Sigma^{-\frac{1}{2}}(\vec{0}, l_2) \tag{13}$$

$$\mathrm{e} = \Sigma^{-\frac{1}{2}}(l_1, l_2) \tag{14}$$

其中e_1，e_2，e分别表示拼接向量$(l_1, \vec{0})$，$(l_2, \vec{0})$，(l_1, l_2)在马氏空间中的向量,通过以上三种情况，我们可以利用下面的马氏距离平方之比来度量两个不同语言的句子之间的平行度：

$$m = \frac{\|e\|_2^2}{\|e_1\|_2^2 + \|e_2\|_2^2} \tag{15}$$

如果两个句子具有相同的含义，则该句子对在马氏空间中的向量e的可能性不应小于孤立地单个句子e_1，e_2在马氏空间中向量的概率，m值越小，两个句子之间的平行度越高。

最后，我们将m值进行标准化，将其转化到[0,1]之间，同时利用式(16)来衡量两个句子之间的平行度，即m'越小，两个句子之间的平行度越高。

$$m' = 1 - m \tag{16}$$

3. 实验与结果

3.1 数据处理

语料过滤任务数据来自 CCMT2020 指定的语料过滤任务开发集(来自 WMT2018 和 WMT2019 的中-英新闻测试集，分别包含 3981 句及 2 000 句原文和对应参考译文)、CCMT2020 不带噪音的汉英平行语料以及 CCMT2020 带噪音的平行语料(3 432 万中-英句对)。

其中对汉语语料使用 Jieba[8]分词工具进行分词，对英语语料使用 Moses[9]脚本分词和小写处理。由于数据量过大，防止在解码时出现显存溢出(Out Of Memory, OOM)报错，因此将小写后的噪音数据进行截断处理，每一个句子最多保留前 256 个单词。同时为了缓解未登录词(Out Of Vocabulary, OOV)问题，即提高模型对稀有词和未登录词的处理能力，本研究使用基于子词(subword)切分的方法，对中文语料和英语语料使用 BPE[10]进行切分。此外，防止一次性加载并解码 3 400 万句对，从而造成的内存紧张

[8] https://github.com/fxsjy/jieba
[9] http://statmt.org/moses/
[10] https://github.com/rsennrich/subword-nmt

和解码时间过长等问题，我们对带噪音的数据进行了切分，每份包含 200 万条数据。

最后，对带噪音的平行语料进行过滤，首先，去掉长度大于 150 个单词的句子，再去掉语种错误的句子。

表 1：各系统对应的 BLEU 成绩，“*”表示该系统仅有一个打分

Table1: BLEU scores for each system, “*” means that the system has only one score

系统	BLEU	
	add	multi
随机系统 0	16.59	*
随机系统 1	16.93	*
Zipporah	15.45	*
词对齐	17.04	*
翻译模型	**17.26**	17.19
语言模型	16.92	16.38
双语预训练模型	17.18	*
规则	16.75	*
领域分类器	14.82	*

3.2 单系统实验

由于各个系统之间无依赖关系，因此可以并行进行各个系统的实验。具体来说，我们选定规则系统、Zipporah 系统、词对齐系统、翻译模型系统、语言模型系统、双语预训练模型系统这 6 个作为基础系统。然后分别依据这 6 个系统的得分从高到低进行排序，最后在 Marian 上进行训练，在开发集上测试排序后的数据对应的 BLEU 值。根据每个系统对应 BLEU 值的高低选择优势特征，之后尝试在优势特征之间组合，得到更优的排序。

对于每一个系统，分别依据该系统的打分对带噪音的数据进行排序，需要注意的是，有些系统不止有一个打分，因此如果有多个打分，那么最终得分有 2 个，一个是各个分数相加(add)，另一个是各个分数相乘(multi)。然后根据得分，使用主办方提供的脚本采样 1 亿个单词(英文端)规模的平行语料进行 Marian 系统的训练。

由于计算资源限制，我们对每个系统只训练 10 轮，取开发集上最高的 BLEU 值作为该系统的最终成绩。每个系统的成绩参考表 1，从表 1 可以看出，各系统成绩相

差较大。此外，我们尝试将数据随机打乱，同样采样 1 亿个单词的平行语料，2 次随机的结果甚至超过了大部分的启发式系统，尤其是 WMT 语料过滤任务中做基线系统的 Zipporah 在此次实验中表现并不好，最好的是基于翻译模型的译文与参考相似性指标。我们观测到翻译模型过滤后的语料排名对句长并不是非常敏感，大量长度适中的句子都有希望排到前面，而其他系统得分都倾向于短句优先。规则系统虽然能无差别对待长句和短句，但由于无法衡量平行程度，因此在独自发挥作用时效果并不突出。

此外，为探究领域对成绩的影响，我们从不带噪音的平行语料中采集了 1409 条中文新闻样本和 1434 条中文非新闻样本，从中划分出 200 条新闻和 200 条非新闻作为开发集，训练一个基于 CNN 的二分类器。该分类器作用于带噪音的数据上，将新闻的预测概率作为得分。其中二分类器性能参考表 2，可以看到该分类器性能较高，但从表中可以看到，该分类器 BLEU 值很低，所以可以认为在此任务中，领域对翻译模型的影响并不大。不过该分类器仅用做验证，我们并未将其纳入到最终的系统中。

表 2：二分类器性能

Table2: Performance of two classifier

类别	Metric		
	P	R	F1
新闻	0.828	0.990	0.902
非新闻	0.988	0.795	0.881

3.3 多系统融合实验

根据单系统实验结果显示，我们认为翻译模型系统、词对齐模型系统、语言模型系统以及双语预训练模型系统是相对潜力较大的系统，因此这些系统之间的组合会被优先进行融合测试。

多系统融合的方法相对比较简单，即将各系统的打分进行融合，然后再重新排序。融合的方法有 2 种：按权重相乘、按权重相加。大部分情况下，我们仅尝试了权重均为 1 的融合。表 3 展示了部分实验结果，可以看到融合系统总体上成绩超过单系统成绩，而且相乘的方法总体优于加法。我们认为融合系统成绩更好的主要原因是因为不同系统从不同出发点对平行程度进行度量，因此多系统融合后能对句对有更全面和更公正的评价。

3.4 提交系统

实验发现并不是集成的系统越多成绩就越好，考虑到系统复杂性，我们取“1，3，4”组合作为主系统，而副系统选择“1，2，3，4，6”组合。经过大量的测试，“1，3，4”组合的鲁棒性和 BLEU 值都较高。而至于副系统，规则方法在 WMT2018 和 WMT2019 年语料过滤任务中被证明为有效的手段，因此我们加入了规则。此外，考虑到预训练模型在语义提取上的优势，因此也将预训练模型系统加入到副系统。

表 3：部分系统融合成绩

其中“1”表示翻译系统、“2”表示双语预训练系统、“3”表示词对齐系统、“4”表示语言模型系统、“5”表示 Zipporah 系统、“6”表示规则系统。

Table3: Partial system integration results

“1” means translation system, “2” means bilingual pre-trained system, “3” means word alignment system, “4” means language model system, “5” means Zipporah system, “6” means rule system

系统组合	BLEU	
	add	**multi**
1，3	17.53	17.60
1，2，3	17.37	17.56
1，3，4	17.66	**17.92**
1，2，3，4	17.18	17.65
1，2，3，4，5	17.38	17.13
1，2，3，4，6	17.30	17.40

4. 总结与展望

本文主要介绍了厦门大学参加 CCMT2020 的汉英平行语料过滤任务构建的系统。我们设计并实现了规则方法和 5 种启发式方法，并进行多种方法的融合来实现噪音语料的过滤。最后实验表明，相比于单系统，我们改进的按权重相乘的多系统融合方法在测试集上取得了效果最好的 BLEU 值。

不过，我们的系统还存在不足之处，比如权重设置单一 ，组合方式有限等。所以在将来的工作中，至少有 2 点改进：首先是可以挖掘更可靠的特征来区分高质量和低质量的数据；其次，将针对特征组合方式做进一步的优化调整，比如引入机器学习模型自动学习最优权重组合，从而提升过滤质量。

参考文献

[1] Koehn P, Khayrallah H, Heafield K, et al. Findings of the wmt 2018 shared task on parallel corpus filtering[C]//Proceedings of the Third Conference on Machine Translation: Shared Task Papers. 2018: 726-739.

[2] Koehn P, Guzmán F, Chaudhary V, et al. Findings of the wmt 2019 shared task on parallel corpus filtering for low-resource conditions[C]//Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2). 2019: 54-72.

[3] Belinkov Y, Bisk Y. Synthetic and natural noise both break neural machine translation[J]. arXiv preprint arXiv:1711.02173, 2017.

[4] Junczys-Dowmunt M. Dual conditional cross-entropy filtering of noisy parallel corpora[J]. arXiv preprint arXiv:1809.00197, 2018.

[5] Sánchez-Cartagena V M, Bañón M, Rojas S O, et al. Prompsit's submission to wmt 2018 parallel corpus filtering shared task[C]//Proceedings of the Third Conference on Machine Translation: Shared Task Papers. 2018: 955-962.

[6] Paetzold G. UTFPR at WMT 2018: Minimalistic supervised corpora filtering for machine translation[C]//Proceedings of the Third Conference on Machine Translation: Shared Task Papers. 2018: 923-927.

[7] Lo C, Simard M, Stewart D, et al. Accurate semantic textual similarity for cleaning noisy parallel corpora using semantic machine translation evaluation metric: The NRC supervised submissions to the parallel corpus filtering task[C]//Proceedings of the Third Conference on Machine Translation: Shared Task Papers. 2018: 908-916.

[8] Pham M Q, Crego J M, Senellart J. SYSTRAN participation to the WMT2018 shared task on parallel corpus filtering[C]//Proceedings of the Third Conference on Machine Translation: Shared Task Papers. 2018: 934-938.

[9] Pinnis M. Tilde's parallel corpus filtering methods for WMT 2018[C]//Proceedings of the Third Conference on Machine Translation: Shared Task Papers. 2018: 939-945.

[10] Chaudhary V, Tang Y, Guzmán F, et al. Low-Resource Corpus Filtering using Multilingual Sentence Embeddings[J]. arXiv preprint arXiv:1906.08885, 2019.

[11] Zariņa I, Ņikiforovs P, Skadiņš R. Word alignment based parallel corpora evaluation and cleaning using machine learning techniques[C]//Proceedings of the 18th Annual Conference of the European Association for Machine Translation. 2015: 185-192.

[12] Parcheta Z, Sanchis-Trilles G, Casacuberta F. Filtering of noisy parallel corpora based on hypothesis generation[C]//Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2). 2019: 282-288.

[13] Reimers N, Gurevych I. Sentence-bert: Sentence embeddings using siamese bert-networks[J]. arXiv preprint arXiv:1908.10084, 2019.