计算机科学
COMPUTER SCIENCE

一种灵活高效的增量式Web平行语料抽取方法

刘小峰, 郑禹铖, 李东阳

引用本文

刘小峰, 郑禹铖, 李东阳. 一种灵活高效的增量式Web平行语料抽取方法[J]. 计算机科学, 2024, 51(11): 248-254.

LIU Xiaofeng, ZHENG Yucheng, LI Dongyang. Incrementally and Flexibly Extracting Parallel Corpus from Web [J]. Computer Science, 2024, 51(11): 248-254.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.231000096

一种灵活高效的增量式 Web 平行语料抽取方法

刘小峰　郑禹铖　李东阳
华中科技大学软件学院　武汉 430074

摘　要　从 Web 中抽取平行语料对于机器翻译和其他多语语言处理任务来说非常重要，由此提出了一种从 Web 中灵活高效地增量抽取平行语料的方法，通过持续地对 Common Crawl 的 Web 抓取存档进行下载、扫描和分析统计，增量更新域名下的语言文本长度统计数据。对于任意给定的感兴趣目标语言对，抽取方法基于域名下的语言文本长度统计数据确定抓取网站入口，并根据目标语言进行定向抓取，忽略多语域名和目标语言外的链接。此外还提出了一种在多语域名内基于语义相似性进行全局对齐的新的句子对齐方法。实验表明，增量抽取能够持续不断地获得新的平行语料，根据指定的语言对进行抽取，可以灵活地获得感兴趣的目标语言对平行语料；新的对齐方法在对齐效率上明显优于全局方法，且能完成局部方法无法完成的对齐；在 6 个语言方向中，抽取到的平行语料在 4 个中低资源语言方向的质量优于现有 Web 开源平行语料，在 2 个高资源语言方向的质量接近现有最好的 Web 开源平行语料。

关键词：平行语料抽取；句子对齐；语料库构建；机器翻译；Web 挖掘
中图分类号　TP391

Incrementally and Flexibly Extracting Parallel Corpus from Web

LIU Xiaofeng, ZHENG Yucheng and LI Dongyang
School of Software Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Abstract　Extracting parallel corpus from the web is important for machine translation and other multilingual processing tasks. This paper proposes an incremental web parallel corpus extraction method, which incrementally updates language text length statistics for domains by continuously downloading, scanning and analyzing Common Crawl's web crawling archive. For any given interested language pairs, web sites to be crawled are determined based on language text length statistics for domains and crawled according to the target language pairs, and non-target domains and links are discarded. It also proposes a new intermediate sentence alignment method, which globally aligns sentences based on semantic similarity within multilingual domains. Experiments show that: 1) our extraction method can continuously obtain new parallel corpus and flexibly obtain the target language pair of interest via extracting the specified language pairs; 2) the proposed intermediate method is significantly better than the global method in terms of alignment efficiency, and can complete the alignment that cannot be completed by local methods; 3) out of 6 language directions, the extracted parallel corpora are superior to existing web open source parallel corpus in 4 medium-low resource languages and close to the best available web open source parallel corpus in 2 high-resource languages.

Keywords　Parallel corpus extraction, Sentence alignment, Corpus construction, Machine translation, Web mining

1　引言

平行语料对于机器翻译和其他双语或多语自然语言处理任务来说至关重要。目前，基于 Transformer 架构的神经机器翻译已成为机器翻译的事实标准，要持续地提高基于这种架构的翻译系统的翻译质量，最重要的途径之一就是使用更大的翻译模型。而要充分发挥大翻译模型的潜力，需要相应地不断扩大训练的平行语料的大小。由于专业人员人工翻译产生平行语料既成本高昂，又难以满足大翻译模型的海量数据需求，从各种媒体（Web[1-2]、字幕[3]、会议记录[4-5]等）中自动抽取平行语料成为大规模获得平行语料的首选方法，但这种方法得到的平行语料的质量较专业人员翻译产生的平行语料低。在众多可以抽取平行语料的媒体中，Web 由于其内容多样、规模庞大和快速更新等特点，成为满足不断增长的翻译大模型平行语料需求的最重要来源。

尽管已有较多工作研究从 Web 中大规模抽取平行语料，但目前还没有工作探讨持续增量地从 Web 大规模抽取平行语料。首先，持续增量抽取基于已完成的抽取工作，随着 Web 的动态更新不断扩大平行语料的规模。其次，以前从 Web 中抽取平行语料的方法要么关注给定语言对（例如抽取英语-日语平行语料的 JParaCrawl[6]、抽取欧洲语言的 ParaCrawl[7]），要么试图抽取各种语言间的平行语料（例如 CCMatrix[1]，CCAligned[2] 和 NLLB[8]），还没有一种方法允许我们灵活指定要抽取的感兴趣目标语言对。此外，目前对

到稿日期：2023-10-16　返修日期：2024-03-13
通信作者：刘小峰（liuxf@hust.edu.cn）

网页中不同语言句子进行对齐有两种方法。一是将全网范围内的一个语言的句子和另外一个语言的句子进行对齐,这称为全局方法[1,8]。它可以提高抽取的查全率,但计算代价非常大,实际只能选择性地稀疏抽取,即抽取部分语言对间的平行语料。二是先对一个网站下的网页进行对齐(常基于网页 URL 的模式),再对对齐网页内的句子进行对齐,这称为局部方法[2,6-7]。它可以提高抽取的查准率,计算效率也高,但会影响查全率,漏掉某些本可以对齐的句子。

针对以上几个问题,本文提出一种 Web 平行语料增量抽取方法,通过持续地对 Common Crawl 的 Web 抓取存档进行下载、扫描和分析统计,增量更新不同域名下的各种语言文本长度统计数据。对于任意给定的感兴趣目标语言范围,基于各种语言文本长度统计数据确定包含给定语言对文本的抓取网站入口,更新待抓取列表,并根据目标语言范围进行定向抓取,同时过滤掉目标语言范围和定向域名外的网页和网站。为了在平行语料抽取查全率和查准率两者间取得平衡,本文采用一种不同于全局方法和局部方法的新的中间方法对句子进行对齐,即,对同一多语域名下的不同语言的句子计算多语句子嵌入,基于多语句子嵌入计算语义相似性来对齐,或者说,在多语域名内基于语义相似性进行全局对齐。本文方法所开发的系统已经开源[1)]。

总的来说,本文在大规模平行语料抽取方面主要有以下几点贡献:

1)在平行语料抽取方面,从流程和数据结构上解决了平行语料的增量抽取问题,可以在以往抽取基础上不重复前面的工作而进行语料抽取;

2)在大规模平行句子对齐方面,提出了一种新的中间对齐方法(在单个多语域名内进行全局对齐的方法),解决了全局方法昂贵对齐代价和局部方法查全率低的问题;

3)在抽取灵活性和个性化方面,抽取流程可以根据指定的兴趣目标语言对进行平行语料抽取,这对于低资源语言对来说极为重要,也极大降低了抽取的盲目性,提高了抽取的针对性。

本文第 2 章将介绍相关工作;第 3 章将给出抽取流程和关键数据结构,并详细介绍每个抽取步骤;第 4 章通过大量实验证明中间对齐方法的优点、增量抽取的必要性以及抽取的平行语料的质量等;最后总结全文并展望未来。

2 相关工作

从 Web 中抽取平行语料有 3 种途径:第一种途径是从 Web 下载的多语字幕文件中抽取平行语料[3],得到的平行语料具有明显的口语对话特点;第二种途径是从 Wikipedia 中抽取平行语料[9-10],得到的语料文本质量较高,但抽取的语料数量相对有限;第三种途径是从 Common Crawl 的抓取存档中获得多语种子,并对多语种子进一步抓取来抽取平行语料[1-2,6-7],这种方法得到的平行语料规模大且内容多样,但包含的噪音较多。以上工作都未讨论如何增量抽取平行语料,我们将采用第三种途径从 Web 中增量式抽取平行语料,使得抽取的语料随着 Common Crawl 抓取存档的增加而不断增加,从而不断提高翻译质量。同时,过去的方法要么对尽可能多的语言对抽取平行语料[1-2],付出的时间和空间代价很大,并且可能抽取一些不需要语言对的平行语料;要么事先指定抽取语言对范围[6-7],抽取缺乏灵活性和针对性。我们将在动态更新的语言文本长度分布信息基础之上对任意指定的语言对进行增量平行语料抽取。

对从 Web 中抽取到的多语文本进行对齐,从总的对齐策略上看可以分为两类方法。第一种对齐策略使用全局方法[1,8],即对全网抽取到的不同语言的文本句子进行多语嵌入,计算不同语言句子间的两两语义相似性来进行对齐。这种方法计算和存储代价非常大,需要分布式 GPU 集群和向量索引[11]才能在部分语言对上实现,但由于在全网范围内进行不同语言句子对齐,因此对齐查全率较好。第二种对齐策略使用局部方法[2,6-7],即先根据网页的链接模式进行网页对齐,再在对齐后的不同语言网页内的文本句子间进行对齐。这种方法会漏掉同一网页内的不同语言文本间的对齐,这种情况在很多外语教学和双语新闻网站比较常见,也会忽略链接模式不匹配的网页,所以对齐查全率较差,但网页内句子对齐计算代价很低,对齐准确率也较高。我们将给出一种对多语域名下不同语言句子进行对齐的中间方法,在对齐查全率和准确率间取得平衡。

通过计算向量间相似性对句子多语嵌入向量进行对齐是大规模平行语料挖掘的常用方法之一,而向量间相似性计算可以使用余弦相似性或边缘分数[1,8,12]。余弦相似性计算简单高效,但全网范围内不同语言句对间余弦分数一致性较差;边缘分数相似性需要使用向量索引,并基于两个向量的 K 个最近邻居计算边缘分数,但一致性较好。通常,全局方法使用边缘分数进行句子对齐,局部方法使用链接模式进行网页对齐。尽管本文提出的中间方法和全局方法一样都基于句子嵌入向量进行对齐,但本文使用余弦分数进行相似性计算。

除了利用多语句子嵌入计算多语句子语义相似性来对齐句子外[13],其他句子对齐方法还包括:基于文本元数据进行对齐[14]、基于文本距离对齐[15]和通过翻译系统进行对齐[15]等。Web 网页或网页文本无可靠的对齐元数据,翻译对齐可伸缩性无法满足海量 Web 文本对齐需求,基于文本距离的对齐准确性较低,因此对于多语和大规模 Web 平行语料抽取场景,基于多语句子嵌入进行对齐更合适。

3 抽取方法

所提方法的整个抽取流程如图 1 所示。该流程通过持续地提供新的 Common Crawl 存档 ID 来启动新一轮增量分析和统计,在增量分析和统计过程中,更新域名下的语言文本分布信息。此外,基于增量分析结果,抽取流程允许指定感兴趣的目标语言对以抽取特定语言对的平行语料。

抽取流程可以多轮增量执行,每轮抽取流程可以分成 4 步:

1)初始化阶段以 Common Crawl 存档 ID 作为输入,得到

1) https://github.com/hlp-ai/mt-data

待分析的 WET 存档路径。

2)统计阶段对 WET 存档进行下载、分析和统计,增量更新域名下各种语言的文本分布。

3)抓取阶段对于感兴趣的目标语言对,基于域名下各种语言的文本分布信息得到包含目标语言对文本的抓取入口,并更新待抓取列表。对包含目标语言对文本的多语域名进行抓取,得到不同目标语言下的文本。

4)对齐阶段对同一域名下的不同目标语言文本进行对齐和过滤,得到平行语料。下面详细介绍每步处理和所涉及的数据结构。

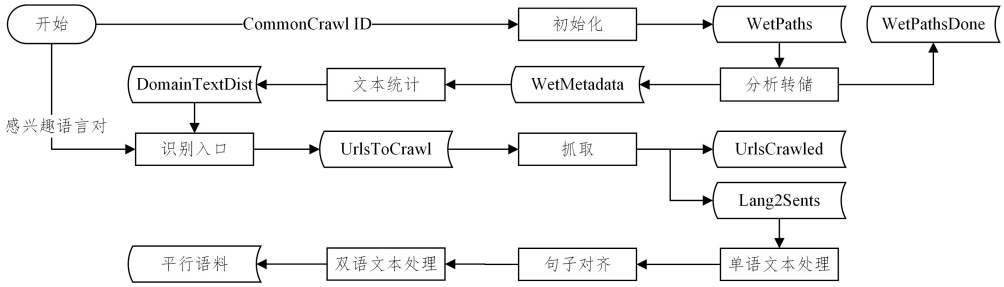

图 1 抽取流程

Fig. 1 Extraction pipeline

3.1 初始化

给定 Common Crawl 存档 ID(如 CC-MAIN-2021-04)标识,对应的 wet.paths.gz 文件包含 WET 存档列表,将其下载到本地,解压得到该存档中包含的所有 WET 存档路径,并保存到对应的 WetPaths 文件中。

3.2 统计

3.2.1 分析转储

将 WetPaths 中每一个未处理(不在已扫描分析 WET 文件 WetPathsDone 中)的 WET 存档下载到本地,解压得到 WET 记录文件。对 WET 记录文件进行解析,顺序扫描每个网页链接记录,利用 pycld2 统计该网页中各种文本的长度,将统计信息顺序添加到对应的 WetMetadata 文件中。WET 存档扫描结束后,将 WET 存档路径顺序添加到已扫描分析的 WET 文件 WetPathsDone 中,删除下载和解压的 WET 存档。

3.2.2 文本统计

对于每一个未统计的 WetMetadata 文件,顺序扫描文件中每条网页的语言文本统计信息,更新内存中临时 DomainTextDist,并将 WetMetadata 文件路径加入已处理 WetMetadata 文件列表中,删除已统计的 WetMetadata 文件。当临时 DomainTextDist 大小达到规定阈值时,按照域名、主机和语言对临时 DomainTextDist 进行排序,然后和外存中同样有序的全局 DomainTextDist 进行顺序扫描合并,得到新的全局 DomainTextDist。

3.3 抓取

3.3.1 识别入口

给定感兴趣的目标语言列表,要抽取这些语言间的平行句对,首先通过顺序扫描全局 DomainTextDist 确定包含这些语言的多语域名,从而获得抓取的入口站点。对于全局 DomainTextDist 中的每条域名记录,若域名下包含全部目标语言且各种语言文本长度之比在阈值范围内(实验中取长度比范围为[1/10,10]),则该域名为多语域名。将多语域名下的网站链接加入该多语域名下的待抓取链接列表 UrlsToCrawl 中。

3.3.2 抓取

由于采用中间对齐,只在多语域名内进行文本对齐,因此可以独立地抓取每个多语域名,域名抓取器间不需要同步和通信,这极大地提高了抓取的可伸缩性,在集群环境下几乎可以实现抓取速度的线性加速。具体来说,每个多语域名启动一个抓取线程进行抓取,多语域名抓取线程从它的 UrlsToCrawl 中获得下一个待抓取的链接进行抓取。对于成功抓取到的网页,利用 BS4 进行 HTML 文档解析,从网页中提取文本和出链。网页文本去掉空行,并进行段落划分,对每个段落利用 FastText 进行语言识别,去掉目标语言范围外的段落,将语言过滤后的段落保存到域名下语言到句子的段落结构 Lang2Sents 中。对于抽取到的出链,先对链接进行绝对化和规范化,再过滤掉目标域名外、目标语言列表外和文本类型链接外的链接。对于出链的语言判断,我们采用一个类似于 ParaCrawl 的启发式规则,但增加了更多的语言支持,并将硬编码语言映射规则改进为配置文件。最后,过滤后的链接如果不在已完成链接列表 UrlsCrawled 中,则将其加入到它的 UrlsToCrawl 中,成功抓取和处理的链接则加入到已完成链接列表 UrlsCrawled 中。

3.4 对齐

3.4.1 单语文本处理

对于每个多语域名抽取到 Lang2Sents 中的不同语言的句子,在对句子进行对齐之前,对单语句子依次进行规范化、去重和单语过滤。其中,规范化包括:去掉不可显示打印和无意义的控制字符,统一各种不同类型的空格字符。句子规范化后,将文本小写并去掉文本中的非字母符号,依据处理后文本的哈希值进行去重。单语过滤依次过滤掉满足以下条件的句子:FastText 给出的目标语言置信度小于 0.5,非字母文字符号比率超过 50%,非本语言符号超过 50%,字符串重复超过 3 次。

3.4.2 句子对齐

同一域名下两种不同语言的句子经过单语文本处理后使用 LaBSE V1[16]进行句子嵌入,将不同语言文本嵌入一个共享的向量空间,得到 784 维句子多语嵌入向量。然后,对一种语言的句子向量和另外一种语言的所有句子向量分别计算

余弦相似性，余弦相似性分数大于给定阈值的句对作为候选平行句对。不同语言对选择的相似性阈值不一样，一般高资源语言对选择较大的相似性阈值。

相较于 CCMatrix 和 NLLB 采用的全局方法，本文提出的中间方法可以极大地提高抽取的效率和速度，结论如断言 1 所示。

断言 1　对 D 个多语域名进行句子对齐，中间方法比全局方法句子对齐快 D 倍。

证明：假设两种方法每个域名平均抽取到 S 条句子，单个句对相似性分数计算代价为 v，则全局方法需要对齐的句子总数为：

$$Ng=D\times S \tag{1}$$

全局方法需要对 N_g 条句子间进行两两相似性分数计算才能对齐，所以全局方法句子对齐相似性分数计算代价为：

$$Cg=Ng\times Ng=(D\times S)\times(D\times S)=D^2S^2 \tag{2}$$

而中间方法分别对 D 个域名中每个域名下的 S 条句子进行两两对齐，则单个域名下中间方法句子对齐相似性分数计算代价为：

$$C_m^d=S^2 \tag{3}$$

因此，中间方法对 D 个域名句子对齐需要的相似性分数计算总代价为：

$$Cm=D\times C_m^d=D\times S^2 \tag{4}$$

所以，中间方法句子对齐代价是全局方法的 $1/D$，或者说中间方法比全局方法快 D 倍。

在对齐过程中选择余弦相似性计算句子向量间的语义相似性。尽管采用余弦相似性具有全局不一致性[1]，但是单个句对的余弦相似性计算代价和句子集合大小无关，具有很好的伸缩性。相反，基于向量索引的边缘分数计算中单个句对的分数计算代价不仅和句子集合大小有关，还和最近邻居数有关。实验部分将定量比较这两种做法的对齐效率和效果。

3.4.3　双语文本处理

由于平行句子中每个语言的句子都经过了规范化，因此对对齐后得到的候选平行句对依次进行去重和句对过滤，不再进行规范化。

句对去重有 3 种选择，分别是基于源语言句子、基于目标语言句子和基于两种语言句子拼接的文本进行哈希去重。基于源语言或目标语言句子的哈希进行去重，只保留句子的唯一翻译，而两种语言句子拼接去重可以保留句子的多种翻译。我们缺省使用基于两种语言句子拼接去重的方式。

由于句对中单语句子在单语文本处理时已经经过单语过滤，我们的句对过滤与已有的方法[17]相比较为简单。依次按照以下过滤条件进行过滤：源语言句子和目标语言句子相同；源语言句子和目标语言句子长度（切分后的词条个数）相差大于 3 倍；源语言句子和目标语言句子非零数字串差异大于 20%。

4　实验及结果分析

考虑到 CCMatrix V1 已经对 Common Crawl 中 2017－2020 年的抓取存档进行了多语言平行句对抽取，为了避免重复工作，也为了证明增量抓取的必要性，将基于 Common Crawl 中 2021 年开始的抓取存档进行平行句对抽取。将实验范围选定为 Common Crawl 的 2021 年上半年 6 个月的抓取存档（CC-MAIN-2021-04，CC-MAIN-2021-10，CC-MAIN-2021-17 和 CC-MAIN-2021-25），并选择中文和其他有代表性的语言对进行实验和比较。

具体来说，根据 CCMarix 对每个汉语（ZH）和其他语言对抽取平行句子数的统计，我们将 ZH-X 语言对分为 3 类：大于 10×10^6 的语言对属于高资源语言对、$1\times10^6\sim10\times10^6$ 间的语言对属于中资源语言对、少于 1×10^6 的语言对属于低资源语言对。为了保证实验的代表性，从每类语言对中选择 2 种属于不同书写体系的语言对，其中，高资源语言对为 ZH-EN（汉语-英语）和 ZH-RU（汉语-俄语），中资源语言对为 ZH-KO（汉语-韩语）和 ZH-VI（汉语-越南语），低资源语言对为 ZH-TA（汉语-塔米尔语）和 ZH-SW（汉语-斯瓦西里语）。

所有实验在一台 32GB 内存、1T SSD、RTX2080TI 和 100M 互联网连接的 PC 上完成，所有代码运行在 Python3.8 解释器上，LaBSE 句子嵌入运行环境为 TensorflowGPU 2.6。

4.1　对齐效率和效果

余弦相似性和边缘分数相似性都可以用于句子嵌入的向量相似性计算，我们的中间方法采用了余弦相似性。下面通过实验定量比较余弦相似性对齐和边缘分数对齐的效率和效果，两种相似性计算方法都采用 LaBSE V1 计算句子的 784 维多语嵌入向量。

首先比较这两种方法的对齐效率。对于余弦相似性，采用成批计算向量余弦相似性的方法，批大小为 16，平均单个句对的余弦相似性计算代价为 2.01ms，这个计算代价与句子集合大小没有关系。对于向量间边缘分数的相似性计算，使用 Annoy 对句子向量进行索引，索引树数为 10，实验结果如表 1 所列。可以看出，随着不同语言句子集合大小 N 和最近邻居数 K 的增加，单个句对的边缘分数计算代价也稳定增加。在 $K=16$ 和 $N=125\,000$ 条句子的情况下，边缘分数计算代价几乎是余弦相似性分数计算的 10 倍。

表 1　边缘分数计算时间随数据集大小和 K 的变化

Table 1　Computation cost of margin score with respect to dataset size and K

(ms)

	$N=1\,000$	$N=5\,000$	$N=25\,000$	$N=125\,000$
$K=4$	11.38	12.64	13.75	15.79
$K=8$	11.85	13.09	14.18	16.25
$K=16$	12.69	14.98	16.07	19.19

接下来比较两种方法的对齐效果。余弦分数和阈值在不同语言对之间难以直接比较。余弦相似性具有全局不一致性，这对全局方法影响较大，对我们的中间方法对齐影响很小。从定性的角度看，全局方法对全网抽取的不同语言句子集合中的句子进行对齐，若采用余弦相似性，则很难选择一个对各种语言对都适用的阈值。而我们的中间方法根据语言选择阈值，并且对齐范围限制在同一多语域名内，因此一致性问题基本不存在。为了验证此猜测，在抽取到的 6 种语言中，对未过滤的句对集合随机采样 20 万个句对，采用文献[1]中推荐的 1.06 边缘分数阈值进行过滤得到边缘过滤结果集，然后

采用不同的余弦相似性阈值进行过滤得到余弦过滤结果集，并计算两个结果集的重叠率。实验结果如表 2 所列。

表 2 余弦相似性和边缘分数 1.06 下的对齐重叠率

Table 2 Alignment overlap rates with different cosine thresholds and margin score of 1.06

余弦阈值	0.50	0.55	0.60	0.65	0.70	0.75
ZH-EN	0.84	0.88	0.92	0.97	0.93	0.91
ZH-RU	0.82	0.87	0.91	0.96	0.91	0.90
ZH-KO	0.83	0.86	0.93	0.95	0.91	0.89
ZH-VI	0.81	0.87	0.93	0.96	0.92	0.91
ZH-TA	0.79	0.86	0.94	0.93	0.90	0.85
ZH-SW	0.80	0.87	0.95	0.94	0.9	0.86

从表 2 可以看出，余弦相似性阈值取 0.6～0.7 时，对齐结果和计算代价较大的边缘分数对齐结果可以取得 94%以上的重叠，而且重叠率与具体语言对关系不大。基于这个原因，在实验中使用余弦相似性对句子多语嵌入向量进行对齐。

4.2 抽取效率

与全局方法一样，我们提出的中间方法需要计算句子间语义相似性来抽取平行句对。为了比较这两种方法的抽取效率，即抽取到的平行句对数和句对相似性分数计算次数之比，进行以下实验。

对于全局方法和中间方法，任意给定语言 L1 和语言 L2，从所有抽取的 L1 语言和 L2 语言的句子集合中各采样 20 万个句子，计算两类语言的句子之间两两相似性分数（全局方法使用边缘分数，中间方法使用余弦相似性分数）。若相似性分数大于给定阈值（全局方法阈值 1.06，中间方法阈值 0.60），则它们互为翻译。对于以上 6 种语言对，进行 10 次采样，计算采样结果中发现的互为翻译的句对数的平均值，并将平均值除以相似性分数计算次数，得到表 3 所列的全局方法和中间方法每百万相似性分数计算发现的互为翻译的句对数。

表 3 全局方法和中间方法每百万相似性计算抽取到互为翻译的句对数

Table 3 Numbers of sentence pairs found to be translations of each other per million similarity computations by global and intermediate methods

语言对	全局方法	中间方法
ZH-EN	0.86	9.71
ZH-RU	0.53	8.42
ZH-KO	0.16	7.07
ZH-VI	0.21	7.25
ZH-TA	0.05	5.91
ZH-AF	0.06	6.06

可以看出，相较于高资源语言对，低资源语言对在相同的语义相似性计算代价下发现的平行句对更少，即抽取效率较低。但是，中间方法只在多语域名内部进行对齐，尽管抽取效率随着资源丰富程度的降低而降低，但降低的程度明显好于全局方法，降低不到 50%（从 9.71 降到 5.91），而全局方法降低超过 90%（从 0.86 降到 0.06）。因此，中间方法的抽取效率明显高于全局方法。

此外，相较于局部方法中间方法，可以发现双语网页内部互为翻译的句子。为了验证这一点，首先通过域名下网站的各种语言文本长度分布自动统计实验发现的双语网页（包含两种语言文本的网页）数，计算双语网页比例；然后从双语网页中随机采样 1 万个网页，人工筛选出包含双语翻译的网页数，计算双语网页中双语翻译网页（同时包含两种语言翻译文本的网页）的比例。统计结果如表 4 所列。

表 4 双语网页比例

Table 4 Percentages of bilingualweb pages

(%)

语言对	双语网页比例	双语翻译网页比例
ZH-EN	0.009	15
ZH-RU	0.007	18
ZH-KO	0.011	22
ZH-VI	0.003	16
ZH-TA	0.00001	30
ZH-AF	0.0005	22

从表 4 可以看出，双语网页比例与语言对资源丰富程度有关，低资源语言对包含更少的双语网页；但是，双语翻译网页比例与语言对资源丰富程度基本无关，最低资源的 ZH-TA 方向的双语翻译网页比例反而最大。尽管双语网页比例很低，但相较于局部方法，我们的中间方法可以从这类网页抽取平行句对，从而提高抽取方法的查全率。

4.3 增量抽取的必要性

为了验证增量抽取的必要性，对 Web 上平行语料随着时间的推移是否有增长以及有多大的增长进行实验测试。首先累计在 Common Crawl 存档的处理过程中所发现包含 6 个语言对的多语域名数，结果如图 2 所示；然后累计 6 个语言对上抽取到的平行句对数量，结果如图 3 所示。

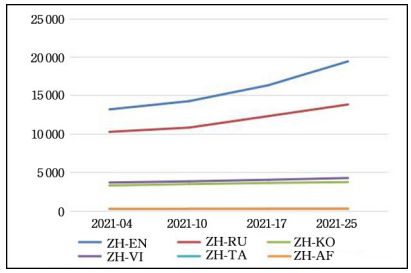

图 2 多语域名总数随时间的变化

Fig. 2 Changes in the numbers of multilingual domains over time

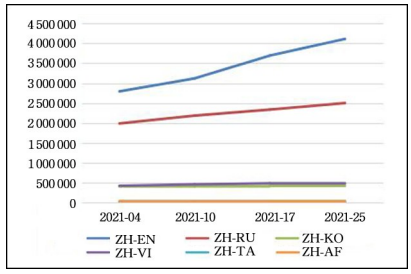

图 3 平行句对总数随时间的变化

Fig. 3 Changes of the numbers of parallel sentences over time

从图 2 和图 3 可以看出，所有语言方向的多语域名和抽取的句对数随着 Common Crawl 存档的增量处理或时间的推移都在增长，这意味着增量抽取是可以不断扩大平行语料规模的。另外，高资源语言对的多语域名数和平行句对数增长更加明显，低资源语言对的多语域名数和平行句对数增长相对比较缓慢。

接下来统计随着 Common Crawl 抓取存档的处理或时间的推移,抽取的各语言对平行句子和 CCMatrix V1 的重复率。统计结果如图 4 所示。

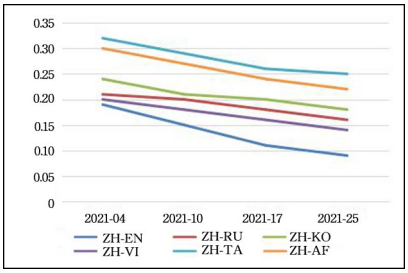

图 4 各语言对抽取平行句子和 CCMatix V1 的语料重复率

Fig. 4 Repetition rates of each extracted parallel sentence pair and CCMatrix V1

从图 4 可以看出,随着 Common Crawl 抓取存档的处理或时间的推移,抽取的各语言对平行句子和 CCMatrix V1 的重复率都在降低,这意味着抽取到越来越多的新的平行句对。一个有趣的现象是,活跃的高资源语言方向由于新的网页内容出现得更加快,重复率更小并且下降也越快。

4.4 抽取语料的质量

为了检验抽取的平行语料的质量,进行两个实验。第一个实验是采用前面介绍过的数据处理和过滤条件,对抽取的平行语料和其他从 Web 抽取的公开平行语料进行处理和过滤,统计平行句对过滤掉的比率或过滤率。过滤率越大,说明抽取的平行语料质量越差,包含的噪音更多。第二个实验是采用相同模型架构和训练条件在抽取的平行语料和其他语料上训练双语翻译模型,并在多个基准翻译评测集上度量翻译模型的质量。

针对第一个实验,考虑到其他语料数据量大小的限制,同时为了保证比较的公平性,在从 Web 抽取的开源平行语料和我们抽取的平行语料中各随机采样 1 万个句对,对句对进行规范化、去重和过滤,共采样 10 次,统计平均过滤率。统计结果如表 5 所列,黑体表示最好的结果。

从实验结果可以看出,随着语言对资源的减少,语料错误对齐的概率增大,过滤率也相应提高。在 6 个语言方向,我们抽取的语料在 4 个中低资源语言方向过滤率最低,且在 2 个低资源语言方向明显优于其他语料,在高资源 ZH-EN 和 ZH-RU 方向过滤率接近最好的 CCMatrix V1。

第二个实验与 6 个语言方向上规模较大的 CCMatrix V1 和 OpenSubtitles2016 进行翻译质量比较。所有实验采用 Transformer Base 架构[18],并基于 OpenNMT-tf 2.26.0 实现,有效批大小 effective_batch_size 为 25 600 词条,批大小 batch_size 为 3 200 词条,优化器为 Adam,加热步为 4 000,初始学习率为 0.000 5,beam 大小为 5,每 1 000 步保存 1 次检查点,并对最后 5 个检查点取平均值。考虑到 OpenSubtitles2016 数据量的限制,为了保证相同大小的训练集,ZH-EN 和 ZH-RU 方向各语料随机采样 100 万句对,ZH-KO 和 ZH-VI 方向各语料随机采样 50 万句对,ZH-TA 和 ZH-AF 方向各语料随机采样 10 万句对。使用 SentencePiece[19] 对文本进行切分,其中,ZH-EN 和 ZH-RU 方向各语言的词汇表大小为 16 000,ZH-KO 和 ZH-VI 方向各语言的词汇表大小为 12 000,ZH-TA 和 ZH-AF 方向各语言的词汇表大小为 8 000。使用 TED 和 FLORES200[8] 的开发集作为训练开发集,当模型在开发集上 4 次更新内无改善时终止训练。对检查点取平均值后的模型在 TED 测试集和 FLORES200 开发测试集上进行评价,使用 SacreBLEU[20] 计算未切分 BLEU,结果如表 6 所列,其中,None 表示对应方向测试集不存在,黑体表示最好的结果。

表 5 各个开源 Web 平行语料过滤率比较

Table 5 Comparison of filter rates of different open source Web parallel corpora

(%)

语料	ZH-EN	ZH-RU	ZH-KO	ZH-VI	ZH-TA	ZH-AF
CCMatrix V1	**25**	**30**	37	35	80	77
XLEnt	32	35	42	40	85	79
MultiCCAglined V1.1	28	29	36	37	79	78
OpenSubtitles V2016	27	29	34	34	65	61
本文方法	26	31	**32**	**31**	**49**	**42**

表 6 基准集上翻译 BLEU 的比较

Table 6 Comparison of BLEUs on benchmark dataset

测试集	训练集	ZH-EN	ZH-RU	ZH-KO	ZH-VI	ZH-TA	ZH-AF
TED	CCMatrixV1	**12.5**	**12.3**	10.1	10.3	5.1	None
	OpenSubtitles2016	11.1	10.4	9.7	9.8	5.0	None
	本文方法	12.3	12.1	**10.2**	**10.4**	**5.8**	None
FLORES	CCMatrixV1	**14.7**	**13.9**	11.3	11.4	5.5	5.7
	OpenSubtitles2016	13.6	12.9	10.3	10.1	5.2	5.3
	本文方法	14.6	13.7	**11.4**	**11.5**	**5.9**	**6.0**

从实验结果可以看出,在中低资源语言 ZH-KO,ZH-VI,ZH-TA 和 ZH-AF 方向上,在我们的抽取语料上训练的模型翻译质量优于 CCMatrix V1 和 OpenSubtitles 上训练的模型的翻译质量,这也从另外一个角度说明我们在 4 个中低资源语言方向上抽取的平行语料质量优于 CCMatrix V1 和 OpenSubtitles2016,这一表现和表 5 给出的实验结论一致。在高资源语言 ZH-EN 和 ZH-RU 方向上,在我们的抽取语料上训练的模型翻译质量略差于 CCMatrix V1 上训练的模型的翻译质量,但优于 OpenSubtitles 上训练的模型的翻译质量。我们抽取的语料和 CCMatrix V1 一样都来自多样的 Web 网页,

如表5所列，我们的语料过滤率略高于CCMatrix V1，在测试集上BLEU也相应更低。尽管在ZH-RU上我们的过滤率略高于OpenSubtitles2016，在ZH-EN上过滤率低于它，但由于它的文本风格和主题多样性较差，因此在2个高资源语言方向上其BLEU都比我们的低。

结束语 文中介绍了一种灵活高效的增量式Web平行语料抽取方法，它允许我们灵活指定感兴趣的抽取语言对范围，实现比全局方法更高的抽取效率，也能抽取局部方法无法抽取的网页。此外，该抽取方法使得我们可以增量地更新抽取信息和不断获得新的平行语料。从实验结果看，所提方法抽取到的平行语料具有较少的噪音和较高的质量。下一步将利用该抽取方法构建一个以中文为中心的大规模平行语料库，特别是抽取机器翻译研究缺乏的低资源中文平行语料。

参 考 文 献

[1] SCHWENK H, WENZEK G, EDUNOV S, et al. CCMatrix: Mining billions of high-quality parallel sentences on the Web [C]// Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics. 2021: 6490-6500.

[2] EL-KISHKY A, CHAUDHARY V, GUZMAN F, et al. CCAligned: A massive collection of cross-lingual web-document pairs [C]// Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. 2020: 5960-5969.

[3] LISON P, TIEDEMANN J. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles [C]// Proceedings of the Tenth International Conference on Language Resources and Evaluation. 2016: 923-929.

[4] ZIEMSKI M, JUNCZYS-DOWMUNT M, POULIQUEN B. The United Nations parallel corpus v1. 0 [C]// Proceedings of the Tenth International Conference on Language Resources and Evaluation. 2016: 3530-3534.

[5] KOEHN P. Europarl: A parallel corpus for statistical machine translation [C]// Proceedings of Machine Translation Summit. 2005: 79-86.

[6] MORISHITA M, CHOUSA K, SUZUKI J, et al. JParaCrawl v3. 0: A Large-scale English-Japanese Parallel Corpus [C]// Proceedings of International Conference on Language Resources and Evaluation. 2022: 6704-6710.

[7] ESPLÀ-GOMIS M, FORCADA M, RAMÍREZ-SÁNCHEZ G, et al. Paracrawl: Web-scale parallel corpora for the languages of the EU [C]// Proceedings of Machine Translation Summit. 2019: 118-119.

[8] JUSSA C, CROSS M, ÇELEBI J, et al. No Language Left Behind: Scaling Human-Centered Machine Translation[J]. arXiv: 2207. 04672, 2022.

[9] TUFIS D, ION R, DANIEL S, et al. Wikipedia as an SMT training corpus [C]// Proceedings of the International Conference Recent Advances in Natural Language Processing. 2013: 702-709.

[10] SCHWENK H, CHAUDHARY V, SUN S, et al. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from Wikipedia [C]// Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics. 2021: 1351-1361.

[11] JOHNSON J, DOUZE M, JÉGOU H. Billion-scale similarity search with gpus [J]. IEEE Transactions on Big Data, 2019, 7(3): 535-547.

[12] ARTETXE M, SCHWENK H. Margin-based parallel corpus mining with multilingual sentence embeddings [C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 3197-3203.

[13] KVAPILÍKOVÁ I, ARTETXE M, LABAKA G, et al. Unsupervised multilingual sentence embeddings for parallel corpus mining[C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020: 255-262.

[14] RESNIK P. Mining the Web for Bilingual Text [C]// Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics. 1999: 527-534.

[15] BUCK C, KOEHN P. Findings of the WMT 2016 bilingual document alignment shared task [C]// Proceedings of the First Conference on Machine Translation. 2016: 554-563.

[16] FANG X Y, YANG Y F, CER D, et al. Language-agnostic BERT Sentence Embedding [C]// Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. 2022: 878-889.

[17] 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. 2018: 726-739.

[18] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need [C]// Proceedings of the 31st International Conference on Neural Information Processing Systems. 2017: 6000-6010.

[19] KUDO T, RICHARDSON J. SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing [C]// Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. 2018: 66-71.

[20] POST M. A Call for Clarity in Reporting BLEU Scores [C]// Proceedings of the Third Conference on Machine Translation. 2018: 186-191.

LIU Xiaofeng, born in 1974, Ph.D, associate professor, graduate supervisor. His main research interests include natural language processing based on deep learning and so on.

（责任编辑：柯颖）