ParaCrawl: Web-Scale Acquisition of Parallel Corpora

Marta Bañón[†], Pinzhen Chen[‡], Barry Haddow[‡], Kenneth Heafield[‡], Hieu Hoang[‡]
Miquel Esplà-Gomis[★], Mikel Forcada[★], Amir Kamran[♦], Faheem Kirefu[‡]
Philipp Koehn[§], Sergio Ortiz-Rojas[†], Leopoldo Pla[★], Gema Ramírez-Sánchez[†]
Elsa Sarrías[★], Marek Strelec[‡], Brian Thompson[§], William Waites[‡], Dion Wiggins[▲]
Jaume Zaragoza[†]
[†]Prompsit, [‡]University of Edinburgh, [★]University of Alicante
[§]Johns Hopkins University, [♦]TAUS, [▲]Omniscien Technologies

Abstract

We report on methods to create the largest publicly available parallel corpora by crawling the web, using open source software. We empirically compare alternative methods and publish benchmark data sets for sentence alignment and sentence pair filtering. We also describe the parallel corpora released and evaluate their quality and their usefulness to create machine translation systems.

1 Introduction

Parallel corpora are essential for building high-quality machine translation systems and have found uses in many other natural language applications, such as learning paraphrases (Bannard and Callison-Burch, 2005; Hu et al., 2019) or cross-lingual projection of language tools (Yarowsky et al., 2001).

We report on work to create the largest publicly available parallel corpora by crawling hundreds of thousands of web sites, using open source tools. The processing pipeline consists of the steps: crawling, text extraction, document alignment, sentence alignment, and sentence pair filtering. We describe these steps in detail in Sections 4–8. For some of these steps we evaluate several methods empirically in terms of their impact on machine translation quality. We provide the data resources used in these evaluations as benchmarks for future research.

As part of these effort, several open source components have been developed. These are integrated into the open-source tool Bitextor,[1] a highly modular pipeline that allows harvesting parallel corpora from multilingual websites or from preexisting or historical web crawls such as the one available as part of the Internet Archive.[2]

The execution of the pipeline has focused on official European Union languages, but also targeted Russian, Sinhala, Nepali, Tagalog, Swahili, and Somali. We show that the obtained parallel corpora improve state-of-the-art results on common benchmarks, such as the WMT Shared Task on News Translation.

2 Related Work

While the idea of mining the web for parallel data has been already pursued in the 20th century (Resnik, 1999), the most serious efforts have been limited to large companies such as Google (Uszkoreit et al., 2010) and Microsoft (Rarrick et al., 2011), or targeted efforts on specific domains such as the Canadian Hansards and Europarl (Koehn, 2005). The book *Bitext Alignment* (Tiedemann, 2011) describes some of the challenges in greater detail.

2.1 Acquisition Efforts

Most publicly available parallel corpora are the result of targeted efforts to extract the translations from a specific source. The French–English Canadian Hansards[3] were used in the earliest work on statistical machine translation. A similar popular corpus is Europarl (Koehn, 2005), used throughout the WMT evaluation campaign.

Multi-lingual web sites are attractive targets. Rafalovitch and Dale (2009); Ziemski et al. (2015) extract data from the United Nations, Täger (2011) from European Patents, Lison and Tiedemann (2016) from a collection of TV and movie subtitles. Cettolo et al. (2012) explain the creation of a multilingual parallel corpus of subtitles from the TED Talks website which is popular due to its use in the IWSLT evaluation campaign.

[1] https://github.com/bitextor/bitextor
[2] https://archive.org/
[3] https://www.isi.edu/natural-language/download/hansard/

There are also various efforts targeted at a single language pair. Martin et al. (2003) build a parallel corpus for Inuktitut–English. Utiyama and Isahara (2003); Fukushima et al. (2006) worked on creating Japanese–English corpora. Uchiyama and Isahara (2007) report on the efforts to build a Japanese–English patent corpus and Macken et al. (2007) on efforts on a broad-based Dutch–English corpus. Li and Liu (2008) mine the web for a Chinese–English corpus. A large Czech–English corpus from various sources was collected (Bojar et al., 2010), linguistically annotated (Bojar et al., 2012), and has been continuously extended to over 300 million words (Bojar et al., 2016).

All these efforts rely on methods and implementations that are quite specific for each use case, not documented in great detail, and not publicly available. A discussion of the pitfalls during the construction of parallel corpora is given by Kaalep and Veskis (2007). A large collection of corpora is maintained at the OPUS web site[4] (Tiedemann, 2012).

2.2 Document Alignment

Document alignment can be defined as a matching task that takes a pair of documents and computes a score that reflects the likelihood that they are translations of each others. The task is typically limited to a single web domain (all web pages from `www.aaa.com` and `aaa.com`, possibly `aaa.de` but not `bbb.com`) for efficiency.

Matching may take the HTML structure into account, or purely rely on the textual content. Examples of structural matching is the use of edit-distance between linearized documents (Resnik and Smith, 2003) and probability of a probabilistic DOM-tree alignment model (Shi et al., 2006). Using the URL for matching is a very powerful indicator for some domains, typically by using a predefined set of patterns for language marking or simple Levenshtein distance (Le et al., 2016).

Content matching requires crossing the language barrier at some point, typically by using bilingual dictionaries or translating one of the documents into the other document's language (Uszkoreit et al., 2010).

Documents may be represented by vectors over word frequencies, typically td-idf-weighted. Vectors may also be constructed over bigrams (Dara and Lin, 2016) or even higher order n-grams (Uszkoreit et al., 2010). The vectors are then typically matched with cosine similarity (Buck and Koehn, 2016a). The raw vectors may be re-centered around the mean vector for a web domain (Germann, 2016)

Document alignment quality can be improved with additional features such ratio of shared links, similarity of link URLs, ratio of shared images, binary feature indicating if the documents are linked, DOM structure similarity (Esplà-Gomis et al., 2016), same numbers (Papavassiliou et al., 2016), or same named entities (Lohar et al., 2016).

Guo et al. (2019) introduce the use of document embeddings, constructed from sentence embeddings, to the document alignment task.

2.3 Sentence Alignment

Early sentence aligners (Brown et al., 1991; Gale and Church, 1993) use scoring functions based only on the number of words or characters in each sentence and alignment algorithms based on dynamic programming. Europarl, for example, used metadata to align paragraphs, typically consisting of 2-5 sentences, and using Gale and Church (1993)'s method to align sentences within corresponding paragraphs. Later work added lexical features and heuristics to speed up search, such as limiting the search space to be near the diagonal (Moore, 2002; Varga et al., 2005).

More recent work introduced scoring methods that use MT to get both documents into the same language (Sennrich and Volk, 2010) or use pruned phrase tables from a statistical MT system (Gomes and Lopes, 2016). Both methods "anchor" high-probability 1–1 alignments in the search space and then fill in and refine alignments. They later propose an extension (Sennrich and Volk, 2011) in which an SMT system is bootstrapped from an initial alignment and then used in Bleualign.

Vecalign (Thompson and Koehn, 2019) is a sentence alignment method that relies on bilingual sentence embeddings and achieves linear run time with a coarse-to-fine dynamic programming algorithm.

2.4 Sentence Pair Filtering

Parallel corpora that have been crawled from unverified web sites and processed by error-prone extraction and alignment methods are likely to contain noise, such as random text fragments, text in the wrong language, translations produced by machine translation tools or bad translators, and

[4] http://opus.lingfil.uu.se/

misaligned sentence pairs. Such noise is specially harmful for neural machine translation (Khayrallah and Koehn, 2018), so filtering it out is an essential processing step.

There is a robust body of work on filtering out noise in parallel data but most recently this topic has gained a lot of momentum, partly due to the lack of robustness of neural models and fostered by recent shared tasks on parallel corpus filtering under high-resource (Koehn et al., 2018) and low-resource data conditions (Koehn et al., 2019).

Most participants in these shared tasks used three components: pre-filtering rules, scoring functions for sentence pairs, and a classifier that learned weights for feature functions.

Pre-filtering rules. Some of the training data can be discarded based on simple deterministic filtering rules. This may remove over 80% of the data (Kurfalı and Östling, 2019; Soares and Costa-jussà, 2019). Such rules remove too short or too long sentences, sentences that have too few words (tokens with letters instead of just special characters), either absolute or relative to the total number of tokens, sentences whose average token length is too short or too long, sentence pairs with mismatched lengths in terms of number of tokens, sentence pairs where names, numbers, dates, email addresses, URLs do not match between both sides, sentence pairs that are too similar, indicating simple copying instead of translating, and sentences where language identifier do not detect the required language.

Scoring functions. Sentence pairs that pass the pre-filtering stage are assessed with scoring functions which provide scores that hopefully correlate with quality of sentence pairs. Participants used a variety of such scoring functions, including n-gram or neural language models on clean data (Rossenbach et al., 2018), language models trained on the provided raw data as contrast, neural translation models (Junczys-Dowmunt, 2018), bag-of-words lexical translation probabilities (González-Rubio, 2019), or even existing off-the-shelf tools like Zipporah and Bicleaner (Chaudhary et al., 2019).

Learning weights for scoring functions. Given a large number of scoring functions, simply averaging their resulting scores may be inadequate. Learning weights to optimize machine translation system quality is computationally intractable due to the high cost of training these systems to evaluate different weight settings. A few participants used instead a classifier that learns how to distinguish between good and bad sentence pairs (where bad sentence pairs are either synthesized by scrambling good sentence pairs or selected from the raw crawled data).

A novel method that was central to the best-performing submission in WMT 2019 was the use of cross-lingual sentence embeddings that were directly trained from parallel sentence pairs (Chaudhary et al., 2019). Other submissions used monolingual word embeddings (Soares and Costa-jussà, 2019; Kurfalı and Östling, 2019; Bernier-Colborne and Lo, 2019).

Another approach is to first train a translation system on the clean data, then use it to translate the non-English side into English and use monolingual matching methods to compare it against the English side of the parallel corpus. Different matching metrics were used: METEOR (Erdmann and Gwinnup, 2019), Levenshtein distance (Sen et al., 2019), or BLEU (Parcheta et al., 2019),

As Rarrick et al. (2011) point out, one type of noise in parallel corpora extracted from the web are translations that have been created by machine translation. Venugopal et al. (2011) propose a method to watermark the output of machine translation systems to aid this distinction, with a negligible loss of quality. Antonova and Misyurev (2011) report that rule-based machine translation output can be detected due to certain word choices, and statistical machine translation output can be detected due to lack of reordering. Rarrick et al. (2011) train a classifier to learn the distinction and show that removing such data leads to better translation quality.

2.5 Comparable Corpus Mining

Our work exploits web sites that provide roughly the same content in multiple languages, leading us to the assumption to find pairs of web pages which are translations of each other, with translated sentences following the same order. This assumption does not hold in less consistently translated web content such as Wikipedia, or accidental parallel sentence found in news stories about the same subject matter written in multiple languages.

There have been increasing efforts to mine sentence pairs from large pools of multi-lingual text, which are treated as unstructured bags of sen-

tences. Munteanu and Marcu (2005) use document retrieval and a maximum entropy classifier to identify parallel sentence pairs in a multi-lingual collection of news stories.

Bilingual sentence embeddings (Guo et al., 2018) and multilingual sentence embeddings (Artetxe and Schwenk, 2018) were tested on their ability to reconstruct parallel corpora. This lead to work to construct WikiMatrix, a large corpus of parallel sentences from Wikipedia (Schwenk et al., 2019) based on cosine distance of their cross-lingual sentence embeddings.

3 Identifying Multi-Lingual Web Sites

Since the start of the collection effort in 2015, we identified potential web sites to crawl in various ways, but mainly by exploiting statistics from CommonCrawl. By splitting this large collection of crawled web pages by web domain and running text extraction and language identification (Buck et al., 2014), we can extract statistics on what language content exists on each of them. Web domains with sufficient content in a targeted language and English are selected for crawling.

The thresholds of what constitutes sufficient content varied depending on language. Typically, we require minimum amounts of content in the targeted language and English (measured in bytes of text), and consider the ratio between the two. For instance, we identified 19,616 web domains with at least 100KB of content in German and English (max ratio 10), but only 438 web domains with at least 20KB of content in Maltese and English (max ratio 10).

It is worth noting that by targeted crawling of web sites we are able to collect many more web pages than present in CommonCrawl. In an exploratory study, only 5% of a collection of web pages with useful content were found in CommonCrawl. This may have improved with recent more extensive crawls by CommonCrawl but there is still a strong argument for targeted crawling.

4 Crawling

Crawling is the initial step of the pipeline. It entails downloading documents from a number of websites and looking for any documents that contain text. These documents are stored as single or multi-domain Web ARChive (WARC) files. WARC is an archiving format for crawled data originally proposed by the Internet Archive

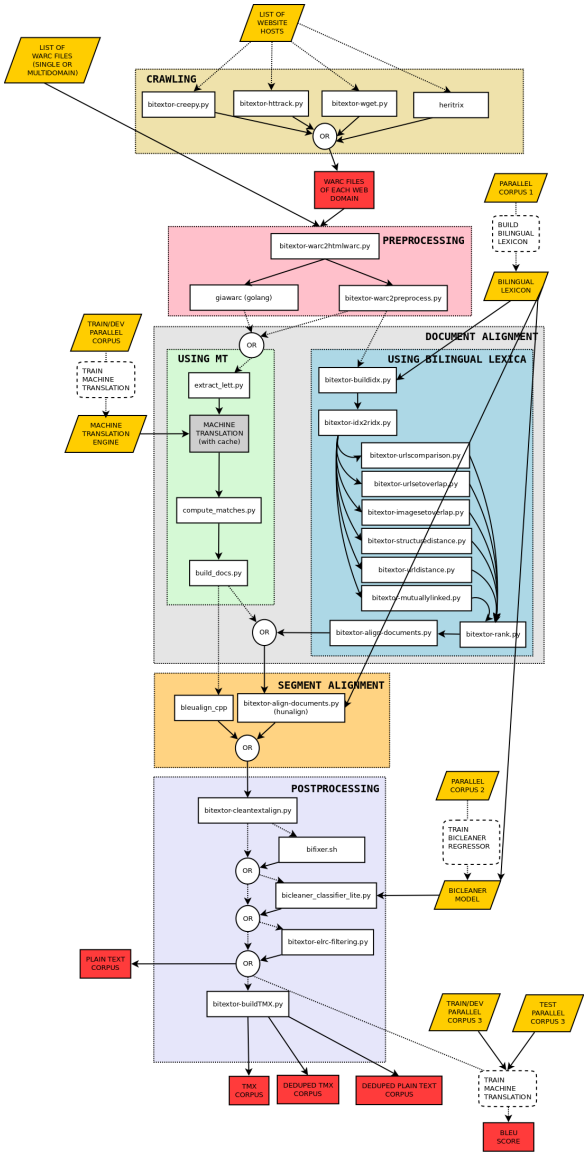

Figure 1: Workflow diagram of Bitextor

and developed by a consortium of libraries and archives into the ISO 28500:2009 standard (ISO, 2009). It consists of a list of gzip-compressed records, each comprising a header with metadata and a crawled document.

Four different crawling tools are currently supported in Bitextor:

HTTrack[5] Well-known multi-platform tool for crawling. It has been for long time in Bitextor, even though it is now deprecated as the support for the tool is discontinued.

Heritrix[6] Internet Archive's web crawler; it is fully compatible with WARC format and supports

[5] https://www.httrack.com/

[6] https://github.com/internetarchive/heritrix3

a variety of options that make it one of the most suitable options for large scale data crawling.

Creepy[7] Python library with basic resources for crawling. A crawler has been implemented on top of it, and is currently experimental.

Wget One of the most popular tools for retrieving files through HTTP and HTTPS in Unix systems. It is fully compatible with WARC format.

Most of our crawling in ParaCrawl has been done using HTTrack. To deal with the I/O-intensive process of writing small files with high frequency, data is first stored on local SSD drives and then transferred to a network file system for subsequent processing.

5 Text Extraction

After crawling, all documents are pre-processed to extract and normalize the text and identify their language. The resulting cleaned and sorted text is the input for the subsequent steps of document and segment alignment (see Sections 6 and 7).

Conversion to HTML WARC files contain one web-crawled document per record. The documents can be in a variety of formats that contain text: plain text, HTML, Open Document Format[8] (".odt"), Office Open XML[9] (".docx") or PDF files containing text. With the exception of the small number of documents that are already in plain text format, the `bitextor-warc2htmlwarc.py` module converts any of these formats to HTML (see fig. 1) and produces WARC files containing only HTML or plain text documents.

Text extraction from HTML Given WARC files containing HTML, we extract the text content. We preserve sentence breaks indicated by HTML tags such as <p> or
 (paragraph and line break), but remove formatting tags such as <b> (for bold text) without a trace.

Language identification with `cld2` and text extraction are currently performed by Python module `bitextor-warc2preprocess.py`; as text extraction is a rather intensive operation, an alternative workflow uses an experimental module written in the Go language, `giawarc`.

[7] https://github.com/aitjcize/creepy

[8] https://www.oasis-open.org/standards#opendocumentv1.2

[9] http://www.ecma-international.org/publications/standards/Ecma-376.htm

6 Document Alignment

There are two main workflows for document alignment.

Using bilingual lexica The traditional workflow in Bitextor until version 5 used bilingual lexica. Module `bitextor-buildidx.py` builds indexes of documents containing, for each word in the lexicon for each language, the documents containing it. Then `bitextor-idx2ridx` uses the bilingual lexica to translate these words and build reverse indexes where each document is paired to a list of documents and bag-of-words-based overlap scores in the other language. A series of modules (`bitextor-urlscomparison.py`, `bitextor-urlsetoverlap.py`, `bitextor-imagestooverlap.py`, etc.), compute a series of features for each language direction based on mutual linking and the comparison of document URLs, the set of outgoing URLs, HTML structure and image content; these features are integrated by `bitextor-rank.py` into two new reverse-index file with new scores, which are used to obtain the final document alignment.

Using machine translation This workflow uses machine translation to decide whether two documents have to be aligned, and is the one that has been used for the parallel data releases of the project (Buck and Koehn, 2016b). After `extract-lett.py` extracts plain-text documents in each language, a machine translation system translates each document from language A to B. We then generate a (sparse) matrix of tf-idf scores between machine translated versions of documents in language A and documents in language B. These scores are used by `compute_matches.py` to compute a list of document pairs (score, source URL, target URL).

Document pairs are stored in a file in which each line contains the URLs of both documents and their plain-text content encoded in base64.

7 Sentence Alignment

During the ParaCrawl project, we made use of a few sentence alignment tools. In this paper, we compare their performance on five language pairs. The sentence aligners are:

Hunalign (Varga et al., 2005) is a widely used tool that relies on a bilingual dictionary that we

Language	Web Domains	Document Pairs	English Tokens
German	21,806	17,109,018	10,788,923,009
Czech	12,179	6,661,650	4,089,806,440
Hungarian	5,560	2,770,432	1,504,698,348
Estonian	5,129	2,301,309	1,427,328,440
Maltese	933	303,198	134,232,546

Table 1: Corpus statistics for data used in the sentence alignment evaluation. Number of English tokens is computed with the Unix command `wc`.

Language	Hunalign	Vecalign	Bleualign
German	35.1 (100m)	**35.8 (150m)**	35.0 (100m)
Czech	21.0 (50m)	**21.2 (50m)**	21.0 (50m)
Hungarian	16.5 (30m)	**16.8 (30m)**	16.6 (15m)
Estonian	**21.8 (20m)**	21.6 (20m)	21.4 (20m)
Maltese	33.5 (5m)	**34.1 (7m)**	30.3 (2m)

Table 2: BLEU scores for systems trained on corpora generated by different sentence aligners. Different subsets are selected based on Bicleaner scores, size of the subsets is given in number of million English tokens.

generated from the Europarl corpus or other available parallel corpora.

Bleualign (Sennrich and Volk, 2010) aligns an English translation of the foreign sentences and the English sentences based on their similarity, as measured by a variant of the BLEU score. We implemented a faster version of Bleualign in C++.

Vecalign (Thompson and Koehn, 2019) is a new sentence aligner based on sentence embeddings, using an efficient coarse-to-fine algorithm with linear run time. We used pre-trained LASER embeddings[10] which cover all the languages of ParaCrawl, except for Irish.

We compared the quality of the sentence pairs extracted from document pairs for these tools. To our knowledge, this is the first evaluation of sentence aligners on large-scale real-world web-crawled data. We selected five languages, ranging from low resource (Maltese) over mid-resource (Estonian, Hungarian) to high-resource (Czech, German). We selected a subset of web domains, for details see Table 1.

The data is provided as document pairs from the usual upstream ParaCrawl processing. The text of web pages needs to be further split into sentences, and then aligned using the different sentence aligners. The resulting sentence pairs are deduplicated are assessed for quality using Bicleaner (more on sentence pair filtering in the next section).

Since different sentence aligners generate different amounts of data (for instance, Bleualign filters quite aggressively for noise), we selected differently sized subsets of the data for evaluation by selecting the best sentence pairs according to Bicleaner quality scores. We built neural machine translation models on these subsets using Fairseq and evaluated them on test sets drawn from the WMT news translation task (newstest2018 for German, Czech, Estonian; newstest2009 for Hungarian) and the EU Bookshop[11] corpus (Maltese).

See Table 2 for the BLEU scores and corpus sizes for the best-performing subsets for each sentence aligner and language. Vecalign gives the best results for 4 of the languages, and is slightly behind Hunalign for Estonian.

We published the document pairs to be aligned, as well as the testing environment[12] to promote the evaluation of novel sentence alignment methods.

8 Sentence Pair Filtering

Our processing pipeline is aimed at high recall at the cost of precision, thus creating large but very noisy corpora. So, as a last processing step, we aim to filter out sentence pairs that are not useful as training data for machine translation or any other purpose.

This is especially important since training on noisy corpora is a challenge for neural machine translation which motivated the organization of two shared tasks in 2018 and 2019, on the high resource language German–English and the low resource languages Sinhala and Nepali, respectively. Here, we extend this evaluation to European languages with medium sized resources.

Building on the data sets generated by the sentence alignment evaluation of the previous section, we compared three sentence pair filtering methods used in the ParaCrawl effort: Zipporah (Xu and Koehn, 2017), Bicleaner (Sánchez-Cartagena et al., 2018), and LASER (Chaudhary et al., 2019).

We carried out the evaluation (see Table 3) in the same fashion, as in the previous section. Filtering by LASER scores gives the best results except for Maltese (for which the publicly available

[10] https://engineering.fb.com/ai-research/laser-multilingual-sentence-embeddings/

[11] http://opus.nlpl.eu/EUbookshop.php

[12] http://www.statmt.org/paracrawl-benchmarks/

Setup	Zipporah	Bicleaner	LASER
de, Hunalign	34.4 (100m)	35.1 (100m)	**36.0 (100m)**
de, Vecalign	34.6 (100m)	35.8 (100m)	**36.3 (50m)**
cs, Hunalign	19.1 (15m)	21.0 (50m)	**22.2 (30m)**
cs, Vecalign	21.4 (30m)	21.2 (50m)	**22.2 (30m)**
hu, Hunalign	16.2 (10m)	16.5 (30m)	**17.2 (10m)**
hu, Vecalign	16.9 (15m)	16.8 (30m)	**17.2 (15m)**
et, Hunalign	21.2 (15m)	21.8 (20m)	**22.1 (15m)**
et, Vecalign	21.3 (20m)	21.6 (20m)	**22.9 (20m)**
mt, Hunalign	32.8 (5m)	**33.5 (7m)**	32.6 (7m)
mt, Vecalign	33.8 (5m)	**34.1 (5m)**	30.2 (7m)

Table 3: BLEU scores for systems trained on subsets of the data selected by different sentence pair filtering methods. The size of the subsets in millions of English words is also reported.

LASER model has not been trained). Moreover, in almost all settings, we achieve better results with Bicleaner than Zipporah.

9 Released Corpora

Overall, the ParaCrawl corpus release v5.0 contains a total of 223 million filtered[13], unique sentence pairs from around 150k website domains and across 23 EU languages with English (see Table 5). However, the data release is highly imbalanced with 73% of sentence pairs comprising of just five languages: French, German, Spanish, Italian and Portuguese. The average (untokenised) English sentence length (over all languages) is 22.9 words, with some notable anomalies. For example, the low-resourced Irish-English pair (27.6 words) has over 50% of sentence pairs originating from the legal domain, where sentences are longer than usual. Furthermore, we noticed that filtered sentences which had been aligned using Hunalign were significantly shorter than those aligned by Bleualign (26.1 and 20.1 words respectively), although we are unsure of the exact reason for this discrepancy.

Our main motivation for creating the ParaCrawl corpus is to improve the quality of machine translation systems. To test this, we trained neural machine translation models where we added the corpus to existing data sets for language pairs that were tackled in the shared task on news translation at the Conference on Machine Translation (WMT) — which we consider a strong baseline.

[13]Sentence pairs with a Bicleaner score of less than 0.7 were discarded, but remain in the RAW release.

[14]sacreBLEU signatures:
`BLEU+case.mixed+lang.*-*+numrefs.1+smooth.exp+tok.13a+version.1.4.2`

Pair	BLEU [14] WMT	BLEU WMT+ParaCrawl-5
en-cs	19.0 (52m)	**19.8 (52m+5.3m)**
cs-en	25.0 (52m)	**25.7 (52m+5.3m)**
en-de	26.2 (5.8m)	**27.7 (5.8m+37m)**
de-en	31.2 (5.8m)	**34.0 (5.8m+37m)**
en-fi	19.9 (2.6m)	**23.3 (2.6m+3.0m)**
fi-en	24.2 (2.6m)	**29.9 (2.6m+3.0m)**
en-lv	12.8 (4.5m)	**16.2 (4.5m+1.0m)**
lv-en	16.2 (4.5m)	**20.2 (4.5m+1.0m)**
en-ro	26.5 (0.6m)	**28.6 (0.6m+2.8m)**
ro-en	30.2 (0.6m)	**35.7 (0.6m+2.8m)**

Table 4: BLEU scores for machine translation systems trained with WMT data adding ParaCrawl release v5.0 data. All the training and test sets are from WMT17 except for Romanian, taken from WMT16. The systems are transformer base trained with Marian using SentencePiece. Sentences are reported in millions.

We trained Transformer-Base models with Marian using SentencePiece. See Table 4 for results. For most language pairs, we see gains of several BLEU points (up to 6 BLEU points for English–Romanian). We even see gains for English–Czech, were ParaCrawl is quite a bit smaller than existing data sets (+0.7 BLEU when adding 5.3m sentence pairs to the existing set of 52m sentence pairs).

10 Computational Costs Concerns

Several of the steps involved in producing and evaluating the ParaCrawl corpora are computationally expensive. Even as some of the steps are embarrassingly parallel and amenable processing in a high-performance computing setting, even pre-processing of 100TB of source data to produce candidate documents consumes on the order of 50,000 CPU-hours equivalent to an estimated[15] 720kWh of power. Training of a neural network model for translating one of the more resource-rich languages such as German may take a week on a dozen GPUs again consuming about 750kWh. Translating 500 million German sentences to English for evaluation consumed roughly 7MWh. In practice, these computations are not simply performed once, they are performed many times as parameters are changed and different strategies tried.

This energy cost is significant. The Typical Domestic Consumption Values published by

[15]The datasheet of an Intel E5-2695 processor says that it uses 115W of power or about 9.5W/core. This estimate includes a 50% margin for main board power and other overhead.

Language Pair	Web domains	Raw Corpus		Clean Corpus	
		Sentence Pairs	English Words	Sentence Pairs	English Words
Bulgarian–English	4,762	248,555,951	1,564,051,100	2,586,277	55,725,444
Croatian–English	8,889	273,330,006	1,738,164,401	1,861,590	43,464,197
Czech–English	14,335	665,535,115	4,025,512,842	5,280,149	117,385,158
Danish–English	19,776	447,743,455	3,347,135,236	4,606,183	106,565,546
Dutch–English	17,887	1,101,087,006	6,792,400,704	10,596,717	233,087,345
Estonian–English	9,522	168,091,382	915,074,587	1,387,869	30,858,140
Finnish–English	11,028	460,181,215	2,731,068,033	3,097,223	66,385,933
French–English	48,498	4,273,819,421	24,983,683,983	51,316,168	1,178,317,233
German–English	67,977	5,038,103,659	27,994,213,177	36,936,714	929,818,868
Greek–English	11,343	640,502,801	3,768,712,672	3,830,643	88,669,279
Hungarian–English	9,522	461,181,772	3,208,285,083	4,187,051	104,292,635
Irish–English	1,283	64,628,733	667,211,260	782,769	21,909,039
Italian–English	31,518	2,251,771,798	13,150,606,108	22,100,078	533,512,632
Latvian–English	3,557	176,113,669	1,069,218,155	1,019,003	23,656,140
Lithuanian–English	4,678	198,101,611	963,384,230	1,270,933	27,214,054
Maltese–English	672	3,693,930	38,492,028	177,244	4,252,814
Polish–English	13,357	723,052,912	4,123,972,411	6,382,371	145,802,939
Portuguese–English	18,887	1,068,161,866	6,537,298,891	13,860,663	299,634,135
Romanian–English	9,335	510,209,923	3,034,045,929	2,870,687	62,189,306
Slovak–English	7,980	269,067,288	1,416,750,646	2,365,339	45,636,383
Slovenian–English	5,016	175,682,959	1,003,867,134	1,406,645	31,855,427
Spanish–English	36,211	2,674,900,280	16,598,620,402	38,971,348	897,891,704
Swedish–English	13,616	620,338,561	3,496,650,816	6,079,175	138,264,978
Russian–English	14,035	1,078,819,759	-	12,061,155	157,061,045
Dutch–French	7,700	38,164,560	Dutch: 770,141,393 French: 817,973,481	2,687,331	Dutch: 60,504,313 French: 64,650,034
Polish–German	5,549	11,060,105	Polish: 202,765,359 German: 198,442,547	916,522	Polish: 18,883,576 German: 20,271,637

Table 5: Size of corpus release 5. The corpus is released in two versions: **Raw** is very noisy data before the sentence pair filtering step. **Clean** has been proven to be useful for training machine translation systems. We release the raw corpus to allow use of other filtering methods, or different thresholds for quality cutoffs.

Ofgem[16], the UK energy regulator, say that a high-consuming household with electric heating is expected to consume 7.1MWh/year. Does an increase of one or two BLEU points justify this cost? For ParaCrawl, we argue that yes, it does, because we are producing an enabling data set whose cost will, we hope, be amortised across many future experiments.

But there is a more general point to be made here: it is not currently the practice in the machine translation community to publish figures about the cost involved in achieving an increase in performance as measured with the standard metrics. It is not straightforward to evaluate when or if we, as a community, have reached a point of diminishing returns where small changes to a family of methods consume an ever-increasing amount of resources yielding only marginal improvements. We therefore suggest adopting a practice of disclosing energy use for experiments in machine translation alongside BLEU scores to make the cost-benefit trade-off explicit.

11 Conclusions

We released the largest publicly available parallel corpora for many language pairs and demonstrated their benefit to train machine translation systems. Going beyond providing data, the goals of this project include the creation of publicly available infrastructure to explore new research directions on parallel corpus mining by releasing open source code for the entire pipeline and public benchmarks for individual processing steps.

Each of the processing steps we describe here still have great potential for improvement, and we hope that our work contributes to the development of novel methods both in terms of better processing of raw parallel data sources, but also increasing the robustness of neural machine translation training when faced with noisy data.

We are especially interested in further extending this work into low resource languages where resources tend to be noisier.

[16] https://www.ofgem.gov.uk/electricity/retail-market/monitoring-data-and-statistics/typical-domestic-consumption-values

Acknowledgement

This work has been supported in part by three projects funded by the Connecting Europe Facility of the European Union (paracrawl.eu), two Google Faculty Research Awards to Philipp Koehn, a Mozilla research grant to Kenneth Heafield, and a donation from eBay to Kenneth Heafield. Hosting is provided by the AWS Public Dataset Program. This work was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (http://www.csd3.cam.ac.uk/), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/P020259/1), and DiRAC funding from the Science and Technology Facilities Council (www.dirac.ac.uk). This paper is the authors' opinion and not necessarily that of the funders.

References

Alexandra Antonova and Alexey Misyurev. 2011. Building a web-based parallel corpus and filtering out machine-translated text. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 136–144, Portland, Oregon. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2018. Margin-based parallel corpus mining with multilingual sentence embeddings. *CoRR*, abs/1811.01136.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 597–604, Ann Arbor, Michigan. Association for Computational Linguistics.

Gabriel Bernier-Colborne and Chi-kiu Lo. 2019. NRC parallel corpus filtering system for WMT 2019. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*.

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In *Text, Speech, and Dialogue: 19th International Conference, TSD 2016*, number 9924 in Lecture Notes in Computer Science, pages 231–238, Cham / Heidelberg / New York / Dordrecht / London. Masaryk University, Springer International Publishing.

Ondřej Bojar, Adam Liška, and Zdeněk Žabokrtský. 2010. Evaluating utility of data sources in a large parallel Czech-English corpus CzEng 0.9. In *Proceedings of LREC2010*.

Ondřej Bojar, Zdeněk Žabokrtský, Ondřej Dušek, Petra Galuščáková, Martin Majliš, David Mareček, Jiří Maršík, Michal Novák, Martin Popel, and Aleš Tamchyna. 2012. The Joy of Parallelism with CzEng 1.0. In *Proceedings of LREC2012*, Istanbul, Turkey. ELRA, European Language Resources Association.

Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. 1991. Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Meeting on Association for Computational Linguistics*, ACL '91, pages 169–176, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christian Buck, Kenneth Heafield, and Bas van Ooyen. 2014. N-gram counts and language models from the common crawl. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Reykjavík, Iceland.

Christian Buck and Philipp Koehn. 2016a. Findings of the wmt 2016 bilingual document alignment shared task. In *Proceedings of the First Conference on Machine Translation*, pages 554–563, Berlin, Germany. Association for Computational Linguistics.

Christian Buck and Philipp Koehn. 2016b. Quick and reliable document alignment via tf/idf-weighted cosine distance. In *Proceedings of the First Conference on Machine Translation*, pages 672–678, Berlin, Germany. Association for Computational Linguistics.

M. Cettolo, C. Girardi, and M. Federico. 2012. WIT3: Web inventory of transcribed and translated talks. In *Proceedings of th 16th International Conference of the European Association for Machine Translation (EAMT)*, pages 261–268.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*.

Aswarth Abhilash Dara and Yiu-Chang Lin. 2016. Yoda system for wmt16 shared task: Bilingual document alignment. In *Proceedings of the First Conference on Machine Translation*.

Grant Erdmann and Jeremy Gwinnup. 2019. Quality and coverage: The afrl submission to the wmt19 parallel corpus filtering for low-resource conditions task. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*.

Miquel Esplà-Gomis, Mikel Forcada, Sergio Ortiz Rojas, and Jorge Ferrández-Tordera. 2016. Bitextor's participation in wmt'16: shared task on document alignment. In *Proceedings of the First Conference on Machine Translation*, pages 685–691, Berlin,

Germany. Association for Computational Linguistics.

Ken'ichi Fukushima, Kenjiro Taura, and Takashi Chikayama. 2006. A fast and accurate method for detecting English-Japanese parallel texts. In *Proceedings of the Workshop on Multilingual Language Resources and Interoperability*, pages 60–67, Sydney, Australia. Association for Computational Linguistics.

William A Gale and Kenneth W Church. 1993. A program for aligning sentences in bilingual corpora. *Computational linguistics*, 19(1):75–102.

Ulrich Germann. 2016. Bilingual document alignment with latent semantic indexing. In *Proceedings of the First Conference on Machine Translation*, pages 692–696, Berlin, Germany. Association for Computational Linguistics.

Luís Gomes and Gabriel Pereira Lopes. 2016. First steps towards coverage-based sentence alignment. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2228–2231, Portorož, Slovenia. European Language Resources Association (ELRA).

Jesús González-Rubio. 2019. Webinterpret submission to the wmt2019 shared task on parallel corpus filtering. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*.

Mandy Guo, Qinlan Shen, Yinfei Yang, Heming Ge, Daniel Cer, Gustavo Hernand ez Abrego, Keith Stevens, Noah Constant, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Effective parallel corpus mining using bilingual sentence embeddings. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 165–176, Belgium, Brussels. Association for Computational Linguistics.

Mandy Guo, Yinfei Yang, Keith Stevens, Daniel Cer, Heming Ge, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2019. Hierarchical document encoder for parallel corpus mining. In *Proceedings of the Fourth Conference on Machine Translation*, pages 64–72, Florence, Italy. Association for Computational Linguistics.

J. Edward Hu, Rachel Rudinger, Matt Post, and Benjamin Van Durme. 2019. Parabank: Monolingual bitext generation and sentential paraphrasing via lexically-constrained neural machine translation. In *Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*.

International Organization for Standardization ISO. 2009. ISO 28500:2009 information and documentation-WARC file format.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Heiki-Jaan Kaalep and Kaarel Veskis. 2007. Comparing parallel corpora and evaluating their quality. In *Proceedings of the MT Summit XI*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Tenth Machine Translation Summit (MT Summit X)*, Phuket, Thailand.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. 2019. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 54–72, Florence, Italy. Association for Computational Linguistics.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. 2018. Findings of the wmt 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, Belgium, Brussels. Association for Computational Linguistics.

Murathan Kurfalı and Robert Östling. 2019. Noisy parallel corpus filtering through projected word embeddings. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*.

Thanh Le, Hoa Trong Vu, Jonathan Oberländer, and Ondřej Bojar. 2016. Using term position similarity and language modeling for bilingual document alignment. In *Proceedings of the First Conference on Machine Translation*, pages 710–716, Berlin, Germany. Association for Computational Linguistics.

Bo Li and Juan Liu. 2008. Mining Chinese-English parallel corpora from the web. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing (IJCNLP)*.

Pierre Lison and Jörg Tiedemann. 2016. Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*.

Pintu Lohar, Haithem Afli, Chao-Hong Liu, and Andy Way. 2016. The adapt bilingual document alignment system at wmt16. In *Proceedings of the First Conference on Machine Translation*, pages 717–723, Berlin, Germany. Association for Computational Linguistics.

Lieve Macken, Julia Trushkina, and Lidia Rura. 2007. Dutch parallel corpus: MT corpus and translator's aid. In *Proceedings of the MT Summit XI*.

Joel Martin, Howard Johnson, Benoit Farley, and Anna Maclachlan. 2003. Aligning and using an English-Inuktitut parallel corpus. In *HLT-NAACL 2003 Workshop: Building and Using Parallel Texts: Data Driven Machine Translation and Beyond*, Edmonton, Alberta, Canada. Association for Computational Linguistics.

Robert C Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Conference of the Association for Machine Translation in the Americas*, pages 135–144. Springer.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4).

Vassilis Papavassiliou, Prokopis Prokopidis, and Stelios Piperidis. 2016. The ilsp/arc submission to the wmt 2016 bilingual document alignment shared task. In *Proceedings of the First Conference on Machine Translation*, pages 733–739, Berlin, Germany. Association for Computational Linguistics.

Zuzanna Parcheta, Germán Sanchis-Trilles, and Francisco Casacuberta. 2019. Filtering of noisy parallel corpora based on hypothesis generation. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*.

Alexandre Rafalovitch and Robert Dale. 2009. United Nations General Assembly resolutions: A six-language parallel corpus. In *Proceedings of the Twelfth Machine Translation Summit (MT Summit XII)*. International Association for Machine Translation.

Spencer Rarrick, Chris Quirk, and Will Lewis. 2011. MT detection in web-scraped parallel corpora. In *Proceedings of the 13th Machine Translation Summit (MT Summit XIII)*, pages 422–430. International Association for Machine Translation.

Philip Resnik. 1999. Mining the web for bilingual text. In *Proceedings of the 37th Annual Meeting of the Association of Computational Linguistics (ACL)*.

Philip Resnik and Noah A Smith. 2003. The web as a parallel corpus. *Computational Linguistics*, 29(3):349–380.

Nick Rossenbach, Jan Rosendahl, Yunsu Kim, Miguel Graça, Aman Gokrani, and Hermann Ney. 2018. The RWTH Aachen University filtering system for the WMT 2018 parallel corpus filtering task. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Víctor M. Sánchez-Cartagena, Marta Bañón, Sergio Ortiz Rojas, and Gema Ramírez. 2018. Prompsit's submission to wmt 2018 parallel corpus filtering shared task. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 955–962, Belgium, Brussels. Association for Computational Linguistics.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia. *CoRR*, abs/1907.05791.

Sukanta Sen, Asif Ekbal, and Pushpak Bhattacharyya. 2019. Parallel corpus filtering based on fuzzy string matching. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*.

Rico Sennrich and Martin Volk. 2010. MT-based sentence alignment for OCR-generated parallel texts. In *The Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010)*.

Rico Sennrich and Martin Volk. 2011. Iterative, MT-based sentence alignment of parallel texts. In *Proceedings of the 18th Nordic Conference of Computational Linguistics (NODALIDA 2011)*, pages 175–182.

Lei Shi, Cheng Niu, Ming Zhou, and Jianfeng Gao. 2006. A dom tree alignment model for mining parallel data from the web. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 489–496. Association for Computational Linguistics.

Felipe Soares and Marta R. Costa-jussà. 2019. Unsupervised corpus filtering and mining. In *Proceedings of the Fourth Conference on Machine Translation (WMT)*.

Wolfgang Täger. 2011. The sentence-aligned european patent corpus. In *Proceedings of the 15th International Conference of the European Association for Machine Translation (EAMT)*, pages 177–184.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Hong Kong. Association for Computational Linguistics.

Jörg Tiedemann. 2011. *Bitext Alignment*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool, San Rafael, CA.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1246.

Masao Uchiyama and Hitoshi Isahara. 2007. A Japanese-English patent parallel corpus. In *Proceedings of the MT Summit XI*.

Jakob Uszkoreit, Jay Ponte, Ashok Popat, and Moshe Dubiner. 2010. Large scale parallel document mining for machine translation. In *Proceedings of the*

23rd International Conference on Computational Linguistics (Coling 2010), pages 1101–1109, Beijing, China. Coling 2010 Organizing Committee.

Masao Utiyama and Hitoshi Isahara. 2003. Reliable measures for aligning Japanese-English news articles and sentences. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 72–79.

Dániel Varga, Péter Halaácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón. 2005. Parallel corpora for medium density languages. In *Proceedings of the RANLP 2005 Conference*, pages 590–596.

Ashish Venugopal, Jakob Uszkoreit, David Talbot, Franz Och, and Juri Ganitkevitch. 2011. Watermarking the outputs of structured prediction with an application in statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1363–1372, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Hainan Xu and Philipp Koehn. 2017. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2935–2940. Association for Computational Linguistics.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*.

Michał Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2015. The united nations parallel corpus v1.0. In *International Conference on Language Resources and Evaluation (LREC)*.

Appendix: Detailed Sentence Alignment and Filtering Results

German	10m	20m	50m	70m	100m	150m	200m
Hunalign/Zipporah	29.9	32.1	33.8	34.3	**34.4**	34.1	33.6
Hunalign/Bicleaner	27.2	30.6	34.0	34.2	**35.1**	33.7	34.6
Hunalign/Laser	32.3	34.6	35.7	35.8	**36.0**	35.3	34.4
Vecalign/Zipporah	30.2	32.6	34.3	**34.6**	34.5	34.0	32.8
Vecalign/Bicleaner	28.1	31.7	34.3	35.0	35.4	**35.8**	35.1
Vecalign/Laser	32.4	34.4	**36.3**	36.1	36.1	35.9	34.7
Bleualign(NMT)/Bicleaner	27.9	30.9	34.5	34.7	**35.0**	34.6	33.1

Czech	10m	15m	20m	30m	50m	70m	100m
Hunalign/Zipporah	18.5	**19.1**	19.0	18.6	17.8	15.8	14.3
Hunalign/Bicleaner	16.2	17.7	18.7	20.2	**21.0**	20.9	19.1
Hunalign/Laser	20.6	21.6	21.8	**22.2**	21.0	20.7	19.6
Vecalign/Zipporah	19.2	20.1	20.9	**21.4**	21.3	20.5	19.7
Vecalign/Bicleaner	16.5	18.1	19.3	20.3	**21.2**	21.1	19.8
Vecalign/Laser	21.1	21.6	21.9	**22.2**	21.8	20.9	20.0
Bleualign(NMT)/Bicleaner	18.0	19.3	20.5	**21.0**	20.5	18.3	17.6
Bleualign(SMT)/Bicleaner	13.2	14.5	15.4	16.3	18.0	19.0	**19.6**

Hungarian	5m	7m	10m	15m	20m	30m	50m
Hunalign/Zipporah	15.4	15.9	**16.2**	15.3	15.0	13.9	12.8
Hunalign/Bicleaner	12.3	13.2	14.8	15.8	16.3	**16.5**	12.4
Hunalign/Laser	16.2	16.7	**17.2**	16.9	16.8	15.9	14.6
Vecalign/Zipporah	15.4	16.0	16.7	**16.9**	15.2	14.1	12.2
Vecalign/Bicleaner	12.4	13.8	14.0	16.1	16.8	**16.8**	13.4
Vecalign/Laser	16.3	16.9	17.0	**17.2**	17.1	16.7	15.6
Bleualign(NMT)/Bicleaner	14.0	15.2	16.2	**16.6**	16.2	14.6	14.7
Bleualign(SMT)/Bicleaner	7.3	9.0	10.1	11.9	13.1	**14.2**	14.2

Estonian	5m	7m	10m	15m	20m	30m	50m	70m
Hunalign/Zipporah	18.3	19.4	20.6	**21.2**	21.0	20.6	18.4	15.6
Hunalign/Bicleaner	17.2	18.0	19.7	20.9	**21.8**	21.0	17.8	15.1
Hunalign/Laser	19.6	20.5	21.2	**22.1**	21.9	20.7	18.4	18.1
Vecalign/Zipporah	18.7	19.7	20.4	21.3	**21.3**	21.3	17.3	15.5
Vecalign/Bicleaner	17.1	18.3	19.8	20.9	**21.6**	21.5	18.3	15.6
Vecalign/Laser	19.5	20.6	21.7	22.4	**22.9**	21.6	18.6	18.5
Bleualign(NMT)/Bicleaner	17.2	19.0	19.8	21.3	**21.4**	19.4	19.4	19.3
Bleualign(SMT)/Bicleaner	15.5	16.5	18.1	**19.9**	19.5	15.0	11.9	11.0

Maltese	1m	1.5m	2m	3m	5m	7m	10m
Hunalign/Zipporah	29.3	29.9	31.6	32.6	**32.8**	31.6	32.3
Hunalign/Bicleaner	29.0	30.1	30.1	31.8	32.7	**33.5**	31.3
Hunalign/Laser$^{\text{zero}}_{\text{shot}}$	29.0	30.2	30.7	31.9	**32.6**	32.6	32.1
Vecalign/Zipporah	27.0	31.9	32.5	33.5	**33.8**	33.0	32.0
Vecalign/Bicleaner	29.1	30.0	30.7	32.5	33.1	**34.1**	33.2
Vecalign/Laser$^{\text{zero}}_{\text{shot}}$	26.2	27.6	27.8	21.1	24.6	**30.2**	24.8
Bleualign(NMT)/Bicleaner	28.0	29.4	**30.3**	28.3	29.5	29.6	29.6
Bleualign(SMT)/Bicleaner	27.5	28.9	30.1	30.3	**30.4**	29.0	28.5