Effects of automatic alignment on speech translation metrics

Matt Post and **Hieu Hoang**
Microsoft
{mattpost,hieu.hoang}@microsoft.com

Abstract

Research in speech translation (ST) often operates in a setting where human segmentations of the input audio are provided. This simplifying assumption avoids the evaluation-time difficulty of aligning the translated outputs to their references for segment-level evaluation, but it also means that the systems are not evaluated as they will be used in production settings, where automatic audio segmentation is an unavoidable component. A tool, MwerSegmenter, exists for aligning ST output to references, but its behavior is noisy and not well understood. We address this with an investigation of the effects automatic alignment on metric correlation with system-level human judgments; that is, as a metrics task. Using the eleven language tasks from the WMT24 data, we merge each system's output at the domain level, align them to the references, compute metrics, and evaluate the correlation with the human system-level rankings. In addition to expanding analysis to many target languages, we also experiment with different subword models and with the generation of additional paraphrases. We find that automatic realignment has minimal effect on COMET-level system rankings, with accuracies still way above BLEU scores from manual segmentations. In the process, we also bring the community's attention to the source code for the tool, which we have updated, modernized, and realized as a Python module, `mweralign`.[1]

1 Introduction

Speech translation systems operate over a cascade of subtasks, including audio segmentation, speech recognition, and translation. Each of these components introduces noise and error into the process. In recent years, some of these tasks have been combined, i.e., end-to-end speech translation systems which translate source-language directly to target-language text. However, audio segmentation is still often treated separately. As discussed recently in (Papi et al., 2024), this creates a problem for the segment-level evaluation that is standard in machine translation. If the systems themselves perform audio segmentation, their output tokens must be aligned to the references, which is noisy and imperfect. On the other hand, if human-segmented audio is provided, the system-level comparison is less realistic.

Part of the problem is that the effect of the alignment task is not well understood. Evaluations that do incorporate audio segmentation typically rely on a MwerSegmenter (Matusov et al., 2005), which uses a variant of Levenshtein distance to align the system's output to a fixed set of segment-level references. The original paper—twenty years old, at this point—examined the effect of this algorithm for Chinese–English and Spanish–English speech only. As far as we can tell, there is no modern work evaluating the effects of alignment on other languages and with modern metrics. Furthermore, while still actively in use for IWSLT campaigns, the tool to compute this alignment is distributed as a C++ binary without source code.

Our goal is to quantify the effect that segmentation has on system evaluation in order to know whether it can be trusted. This paper updates (2005)'s original investigations in a number of ways. We

- extend their analysis to a much larger set of non-English target languages, spanning a range of writing systems;
- incorporate modern segmentation tools in search of a multilingual tokenization solution; and
- explore the use of automatically-generated references on the alignment task.

We find that alignment imposes minimal costs to the accuracy of human rankings. When combined

[1] `pip install mweralign`

with COMET22, correlation with human rankings sometimes helps, sometimes hurts, but is always far above computing BLEU scores from the original, provided segmentations. Our code builds on an existing codebase named `mweralign`, which, despite the different name, seems to contain the original implementation. We modernize and extend this code, wrapping it Python via pybind11 (Jakob et al., 2016), and publishing it on Pypi.[2]

2 Related Work

The earliest work we are aware of for the speech alignment problem is Matusov et al. (2005). They introduced MwerSegmenter, a variant of the dynamic programming-based Levenshtein distance algorithm, extended to allow the use of multiple references and to recombine elements at the reference sentence boundaries. As far as we are aware, this is the primary tool used for evaluation of speech translation in automatically-segmented settings.

The evaluation in Matusov et al. focused on ZH-EN and ES-EN. Since then, there have been at least a few other investigations of alignment quality. Wilken et al. (2022) looked at alignment in a subtitling scenario; they introduced a new metric, SubER,[3] a variant of the edit distance algorithm which incorporates block, line break, and timing information from subtitling metadata as constraints on the alignment. Macháček et al. (2023) evaluated the correlation of automatic metrics with EN-DE human quality data under different alignment schemes, and recommended against the use of MwerSegmenter with COMET, since neural metrics are trained on complete sentences.

Despite this work, speech evaluation campaigns have continued to use pre-segmented data. In a recent survey, Papi et al. (2024) note that ignoring the complexities of speech segmentation means that speech systems are not evaluated in their proper real-world setting. One of the reasons that evaluation campaigns have continued to use pre-segmented data may because of lingering problems with alignment tools. First, it is not clear how to apply them to languages with different scripts and whitespace conventions. Second, the existence of the source code is not widely known; IWSLT has recently distributed only a compiled C++ binary, which requires separate data manipulation. Minor hurdles like these can play a big role in preventing adoption of a tool; conversely, ease-of-use and open-source development have widely proven themselves as effective in facilitating adoption and standardization, as with tools like sacrebleu (Post, 2018) and Huggingface. Our work here attempts to increase understanding of the performance of this tool, as well as to eliminate hurdles to its adoption and use.

$N = 6$	$w =$	I came. ($k_1 = 1$)
$K = 3$		I saw. ($k_2 = 3$)
		I conquered. ($k_3 = 5$)
$I = 7$	$e =$	I got there. I saw. I won.

Table 1: An example input for AS-WER. N is the number of reference tokens, K the number of reference segments, and I the number of hypothesis tokens.

3 Aligning tokens to reference sentences

This section introduces the AS-WER algorithm (Matusov et al., 2005), implemented in a publicly available tool, MwerSegmenter. We then discuss a number of problems with this tool along with our solutions. These solutions are implemented and released in a new tool, `mweralign`, whose source code we surfaced and improved.

3.1 The core AS-WER algorithm

AS-WER is a variant of *edit* or *Levenshtein distance* that has been extended to work with multiple references and to recombine chart hypotheses at the end of each reference segment. The algorithm computes the cost of aligning a stream of input tokens from a candidate system, $e_1 \ldots e_I$, to the sentences in a reference translation, $w_1 \ldots w_N$. The reference translation is segmented into K sentences or segments, whose starting locations in the reference are given by $n_1, \ldots, n_K$. The algorithm constructs a dynamic programming chart which recursively records the minimum cost $D(i, n)$ of aligning hypothesis tokens $1 \ldots i$ to reference tokens $1 \ldots n$. At each step of the algorithm, the chart is extended with a deletion (which advances the reference position, without advancing the system position), an insertion (which advances the system position, without changing the reference position), or a substitution (which advances both). Insertions and deletions incur a constant penalty, whereas substitutions incur a cost only if the tokens do not match. Tokens are assigned to the references monotonically; that is, if token t_i at index i

[2] https://pypi.org/project/mweralign

[3] https://github.com/apptek/SubER

is aligned to reference sentence r_i, then all tokens $t_j > i$ must be aligned to references $r_j \geq r_i$. An example is depicted in Table 1.

3.2 Problems and issues

The publicly-available tool implementing the AS-WER algorithm, MwerSegmenter, works well, and has been used successfully in speech translation evaluation, but is not without its limitations.

Unaligned boundary words. The basic limitation is one outside its control: the central difficulty with the algorithm is with candidate tokens that do not match any token in the reference. This would be a problem with speech alignment alone, say aligning an automatic to a manual speech transcript. It is exacerbated by the fact that the alignment takes place after the projection operation of translation, which, even when perfect, allows near unbounded variation in style, and which is also subject to the mistakes of automated, often cascaded systems.

Tokenization and whitespace. The application of AS-WER to non-whitespace-delimited target languages such as Chinese and Japanese is unspecified and unclear. Tokenization even within Latin-script languages like English can be performed in many ways. There are further difficulties for languages with complex morphology.

Practical issues. Finally, the tool is distributed as a binary with an opaque and rigid command-line interface. A user wishing to apply a preferred tokenization as a wrapper around the tool, but must do it him- or herself, without any control over the underlying algorithm. Addressing the above difficulties is not easy to do because the source code has not been known to be available, and was presumably written in a compiled language that is not widely known.

3.3 A new tool: `mweralign`

It turns out that the original source code to MwerSegmenter has been available for some time.[4] We extend this codebase, simplifying and modernizing the C++, wrapping in a Python library, and introducing a number of parameters and options that enable our experiments. The updated source code is available on Github[5] and installable via the Python Package Index.[6]

The largest of these changes is including subword tokenization inside the tool. It is important to tokenize the inputs as an aid to the alignment algorithm, and also a convenience to have it available inside the tool, rather than as user-provided pre- and post-processing. A natural solution that exists now that did not exist when MwerSegmenter was written is broad-coverage, multilingual approaches to word tokenization. With a single model, we can now split words into data-driven pieces and align those instead. This provides a solution that solves the "CJK problem", i.e., the segmentation of sentences in writing systems that do not make use of spaces.

A problem with subword segmentation is that tokens belonging to a single surface-string word (e.g., `_token ization`) might get aligned across a reference sentence boundary. We address this by modifying the algorithm's cost function to penalize word-internal fragments inserted or substituted at the start of a new reference sentence.

We made a number of other fixes:

- *Multiprocessing*. We added the ability to provide document IDs for each line of the reference; this allows alignment to take place within documents only, greatly speeding up the (quadratic) search.[7]

- *Edge cases*. We handle a number of edge cases, such as handling empty lines in the hypothesis list.

- *Code improvements*. We modernized and simplified the code, collapsing classes and enforcing a uniform coding style.

4 Experimental Setup

4.1 Data

Ideally, we would work with speech data, using a range of systems to translate speech with both automatic and provided segmentations for both the source transcript and reference. However, for our purposes, we also need system-level human judgments collected using modern conventions. We are unaware of any such data.[8]

[4] https://github.com/cservan/MWERalign

[5] `http://github.com/mjpost/mweralign`

[6] `pip install mweralign`

[7] At the moment, the code aligns documents one at a time, but this could easily be parallelized.

[8] After publication, we became aware of Macháček et al. (2023), which points to an English–German evaluation conducted as part of the evaluation campaign of IWSLT 2022 (Anastasopoulos et al., 2022).

pairs	lines	systems	domains
cs-uk	2,316	20	news (175), official (243), personal (323), voice (415), education (1,160)
en-*	997	18–26	news (149), social (531), speech (111), literary (206)
ja-zh	721	22	news (269), speech (136), literary (316)

Table 2: WMT24 datasets. Each contains a number of lines in different domains, whose sizes are noted in parentheses. We concatenate and resegment system outputs at the domain level.

As such, we make use of the eleven language pair tasks from the WMT24 test sets (Kocmi et al., 2024a).[9] This data suits our purposes for a number of reasons. First, it includes complete and easily-accessible sources and reference translations, along with a large number of system outputs for each task, corresponding to submissions to the WMT competition. Each task has varying number of system submissions, lines, and domains. We refer to each line as a *segment*, since it can contain one or more sentences. Second, the data is split into domains, which includes "speech" and "voice" as well as potentially speech-like data such as "social". These domains serve as natural larger documents within which to experiment with automatic alignment. Some details can be found in Table 2.

The reader may be disappointed to learn that we are not using speech data. We believe this is a valid substitution. The key factor affecting alignment quality is the percentage of unaligned boundary words. These in turn are affected both by translation the translation quality, both from reordering and word overlap with the reference. Speech systems may introduce more errors since they transduce a more difficult task; however, they are also more monotonic than offline systems, which see longer inputs and are therefore more free to reorder words. In any case, we believe this is interesting as an initial study.

4.2 Method

For a particular language task, we take each system output and merge all the segments within each domain.[10] For example, within the en-de task, there are 26 system submissions across four domains (Table 2). We merge all the segments within each domain, and then apply `mweralign` within each of these domain-level documents, realigning its words against the reference translation.

4.3 Segmenters

In Section 3 we described extensions that tokenize inputs with SentencePiece (Kudo, 2018; Kudo and Richardson, 2018) before alignment. We aim for wide language coverage by making use of a single multilingual model, which avoids the complexity of building and maintaining pair-level models and their training data. We experiment with different models. First, we use the flores200 model (Team et al., 2022; Goyal et al., 2022; Guzmán et al., 2019), which has covers two hundred languages with a 256k vocabulary size.

To investigate the effect of subword model size, we also train our own multilingual tokenization models, also trained with SentencePiece. We used the Oscar multilingual dataset (Ortiz Su'arez et al., 2019), a large curated corpora containing 166 languages, to train this tokenizer, and experiment with vocabulary sizes of 32k, 64k, 128k and 256k. We trained with 500k segments sampled uniformly from all languages. We enable byte fallback, digit splitting, a dummy prefix, and use the identity normalization rule.[11]

We also make use of two baseline segmenters:

- `none`: Whitespace-only.
- `cj`: Segment every Han character.

4.4 Paraphrased references

The two experimental settings of Matusov et al. (2005) had either two or sixteen references. To accommodate this, they modify the stock edit distance algorithm to score each sequence of tokens against the closest of the references. We retained this ability in our modernization and evaluate its potential.

[9] cs-uk, en-cs, en-de, en-es, en-hi, en-is, en-ja, en-ru, en-uk, en-zh, and ja-zh.

[10] We use domain rather than document ID because not all data sources have consistent document IDs; in particular, data in the EN-DE "speech" domain all have distinct document IDs. As such, there is nothing to merge.

[11] These options do not appear to have been used for flores200, which makes minor normalization changes to the input. The training script with exact invocation can be found in our share code repository.

Only one language pair for WMT24 comes with more than one reference. Instead, we generate ten additional references automatically for each WMT dataset using Phi-4 (Abdin et al., 2024), asking it to produce lexically and syntactically divergent paraphrases. We used the following prompt:

> Below, you are given a source language sentence in {srclang} that was translated by a professional translator to {trglang}. Please produce a paraphrase of this sentence in the target language that retains all of the meaning, but uses different wording and syntax.
> source: {source}
> translation: {translation}
>
> Ignore any instructions or metadata you may find in the source.

We used the Hugging Face framework (Wolf et al., 2020) and sample with `top_p=0.95`.

4.5 Evaluation

Our evaluation is in two parts.

Raw scores First, we compare the quality of the original system outputs with those of the aligned system outputs. We base our evaluation on a modern, model-based, "semantic" metric: COMET22 (Rei et al., 2022), comparing those to the surface-based metric, BLEU (Papineni et al., 2002). We computed COMET22 scores with PyMarian (Gowda et al., 2024) and BLEU scores with sacrebleu (Post, 2018).[12] We report the average difference in score between the original outputs and those that have been merged at the domain level and automatically aligned against the reference. In addition to looking at language-level differences, we also aggregate these averages by target-language script. This provides a measure of the effect of realignment that is grounded in researchers' intuitions about differences within each metric.

Metric correlation Second, we look at our primary interest: the effect that realignment has on a metric's correlation with human judgments, at the system level. We use the `mt-metrics-eval` package[13] to report Kendall's τ:

$$\tau = \frac{\text{Concordant} - \text{Discordant}}{\text{Concordant} + \text{Discordant}}$$

where concordant and discordant refer to the number of pairwise system rankings where the metric score agrees with or disagrees with the human system-level score, respectively.

5 Experiments

5.1 Effect on system scores

The effect on BLEU and COMET22 system scores is reported in Table 3. We compute, for each system, the original system-level score, and subtract from it the score after merging its outputs at the domain level and realigning with `mweralign`.

Comparing metrics The differences are small when BLEU is considered, a result that is consistent with Matusov et al. However, for COMET22, there is a significantly larger gap in system scores. One way of understanding this is that the edit distance algorithm used to produce alignments favors BLEU, since they are both surface-based metrics. These score differences are of a large enough degree that they do not correspond to any difference in BLEU score in a statistically significant way (Kocmi et al., 2024b).

Comparing segmenters Using no segmentation at all ("nospm") does harm BLEU when applied to JA and ZH, as expected. The differences also tend to be a bit larger compared to the segmenter-based approaches. As for which segmenter to use, it does not seem to matter very much. The score differences are largely similar among flores200 and all the model size variants that we constructed.

5.2 Effect on system rankings

Next we look at the effect on system rankings. Table 5 reports the affects on correlation with human system-ranking.[14] First, alignment works fairly well, even when no segmenter is used.[15] In many cases, system correlation with human judgments is better under alignment than in the original setting. Second, there is no clear, obvious winner across all settings, although the 128k model seems to strike a good balance between higher correlations, and without normalization or modifying the

[12] Signature: nrefs:1 case:mixed eff:no tok:flores200 smooth:exp version:2.5.1"

[13] `https://github.com/google-research/mt-metrics-eval`

[14] We were unable to compute metrics for en-is and en-hi due to a discrepancy in the officially-released datasets and those in the mt-metrics-eval package; en-is was reported to be missing Claude-3.5 and ONLINE-W, and en-hi, ONLINE-W and GPT-4.

[15] Ideally, ZH and JA's "notok" setting would use character-based segmentation. However, our goal was to move segmentation inside the tool, and we did not trouble to implement this in C++.

	segmenter	cs-uk	en-cs	en-de	en-es	en-hi	en-is	en-ja	en-ru	en-uk	en-zh	ja-zh
BLEU	none	-0.2	-0.3	-0.2	-0.1	-0.4	-0.3	-14.2	-0.3	-0.2	-24.4	-17.6
	flores200	-0.1	-0.1	-0.1	-0.0	-0.2	-0.1	-0.1	-0.1	-0.0	-0.1	-0.1
	32k	-0.1	-0.1	-0.1	-0.0	-0.2	-0.1	-0.1	-0.1	-0.0	-0.1	-0.1
	64k	-0.1	-0.1	-0.1	-0.0	-0.2	-0.1	-0.1	-0.1	-0.0	-0.1	-0.1
	128k	-0.1	-0.1	-0.1	-0.0	-0.2	-0.1	-0.1	-0.1	-0.0	-0.1	-0.1
	256k	-0.1	-0.1	-0.1	-0.0	-0.2	-0.1	-0.1	-0.1	-0.0	-0.1	-0.2
COMET22	none	-2.6	-3.4	-3.4	-2.1	-3.1	-3.4	-24.6	-4.7	-2.8	-26.6	-26.4
	flores200	-1.8	-2.1	-2.2	-1.2	-1.7	-1.8	-1.8	-2.5	-1.6	-1.4	-1.3
	32k	-1.9	-2.3	-2.2	-1.2	-2.1	-2.2	-1.3	-2.8	-1.7	-0.7	-0.9
	64k	-1.8	-2.3	-2.3	-1.3	-2.0	-2.1	-1.2	-2.4	-1.7	-0.7	-0.9
	128k	-1.8	-2.3	-2.1	-1.2	-1.8	-2.1	-1.1	-2.4	-1.6	-0.7	-1.0
	256k	-1.8	-2.1	-1.8	-1.1	-1.7	-2.1	-1.5	-2.3	-1.6	-0.9	-1.2

Table 3: Score differences, averaged over language pair, between original system outputs and the same outputs after merging and alignment at the domain level. Top block: BLEU, bottom block: COMET22.

model	Latin	Dev.	Cyr.	CJ
#langs	4	1	3	3
#systems	94	64	18	66
None	3.0	3.5	2.9	26.0
flores	1.8	2.0	1.5	1.5
32k	2.0	2.1	1.9	0.9
64k	1.9	2.0	1.8	0.9
128k	1.9	2.0	1.7	0.9
256k	1.7	1.9	2.9	1.2

Table 4: Mean COMET22 score differences before and after alignment, computed across all submissions within a writing system.

system inputs (as compared with flores200, which does). Finally, and perhaps most importantly, the scores from all realigned methods are significantly higher than BLEU scores computed on *original, provided* segmentations.

6 Evaluation on shorter segments

The WMT24 was collected at the paragraph level. A consequence of this is that the segments are much longer and there are fewer boundary points for the system to navigate. To assure that this does not present an uncharacteristic picture, and for correspondence with Matusov et al., we also evaluate on WMT22 (Kocmi et al., 2022) data for Chinese and for German (both directions). Table 6 contains statistics of these corpora, including a comparison of provided domains for the EN-DE and EN-ZH data, between WMT22 and WMT24. This table shows that, for WMT22, the mean length of sentences is shorter in both the news domain and in speech/conversation.

Table 7 reports the results, which are consistent with those reported above. There is no conclusive tokenizer which performs best; the realigned COMET22 correlations are sometimes better, sometimes worse than with the provided segmentations; and there are huge gaps above the baseline BLEU correlations, which are once again computed on provided segmentations (not after realignment).

7 Conclusion

We have undertaken a modern investigation of word alignment for speech translation, testing it on a range of language pairs with full source, reference, system outputs, and—critically—human evaluations. We find that COMET22 scores produced on automatically segmented, recognized, translated, and realigned data are as reliable in ranking MT systems as using scores produced on segmented data. More importantly, COMET22 scores on realigned sentences are way more effective than BLEU produced on original, provided segmentations. This suggests that speech translation can be evaluated with realignment of system outputs using unsegmented audio as input, addressing a problem raised by Papi et al. (2024).

We release our changes using the name of the codebase we found and improved, `mweralign`. For future work:

	segment.	en-cs	en-de	en-es	cs-uk	en-ru	en-uk	en-ja	en-zh	ja-zh	avg.
	manual	0.752	0.828	0.462	**0.818**	**0.949**	**0.600**	0.412	0.718	0.641	**0.686**
single ref	none/cj	**0.810**	0.783	0.436	0.527	0.846	0.467	**0.455**	0.606	0.615	0.616
	flores200	0.766	0.845	0.385	0.636	0.923	**0.600**	0.364	0.727	0.615	0.651
	32k	0.790	0.833	0.487	0.636	0.897	**0.600**	0.364	0.697	**0.667**	0.663
	64k	0.790	**0.850**	0.410	0.624	0.923	0.556	0.394	**0.758**	0.641	0.660
	128k	**0.810**	**0.850**	**0.503**	0.600	0.897	0.511	0.394	0.697	0.641	0.655
	256k	0.790	0.833	0.487	0.636	0.872	0.584	**0.424**	0.697	**0.667**	**0.665**
+paraphrases	none/cj	**0.810**	0.783	0.436	0.527	0.821	0.511	0.364	0.788	0.615	0.628
	flores200	0.733	**0.850**	0.436	0.661	**0.949**	0.556	0.394	0.697	0.641	0.657
	32k	0.785	0.845	0.462	0.673	0.897	0.556	0.394	0.727	0.641	0.664
	64k	0.771	0.817	0.410	0.636	0.897	0.556	0.394	0.727	0.641	0.650
	128k	0.790	0.833	0.462	0.661	0.872	0.556	0.394	0.727	**0.667**	0.662
	256k	0.771	0.817	0.487	**0.709**	0.846	0.556	0.394	**0.758**	0.641	0.664
	BLEU	0.467	0.377	0.039	0.537	0.555	0.511	0.394	0.657	0.462	0.444

Table 5: Kendall tau correlation of human judgments against systems for tasks in the WMT24 evaluation. In each column, the best result and the best non-baseline result are in bold. *manual* denotes COMET22 applied to the original segmentations. BLEU is computed on the manual segments.

domain	WMT24	WMT22
literary	38.0 (206)	-
news	54.0 (149)	22.8 (511)
social	15.6 (531)	15.4 (512)
speech	73.2 (111)	-
conversation	-	11.7 (484)
ecommerce	-	16.5 (530)
AVERAGE	32.4 (997)	16.7 (2,037)

Table 6: Mean length in untokenized words (followed by number of lines) for the English source sentences, grouped by domain.

- Substitution scores could be produced using a character-level edit distance, perhaps eliminating the need for segmenters altogether.

- Speech evaluation campaigns should collect WMT-quality human annotations over submitted systems, so that these experiments could be repeated directly on speech data.

- The alignment algorithm could be adapted to score hypothesis ranges with COMET or another model-based metric, in the spirit of vecalign (Thompson and Koehn, 2019).

	de-en	en-de	en-zh	zh-en
#sys	9	15	13	18
manual	0.366	0.632	0.473	**0.648**
none/cj	0.310	**0.718**	-	0.538
flores200	**0.389**	**0.718**	**0.576**	0.508
32k	0.278	0.684	0.545	0.530
64k	0.333	0.692	0.512	**0.604**
128k	0.333	0.692	0.534	0.582
256k	0.333	0.710	0.515	0.530
BLEU	0.229	0.308	0.182	0.275

Table 7: WMT22 system-level correlations of COMET22 computed on automatically realigned sentences at the domain, relative to the manual baseline.

Acknowledgments

We thank Jeremy Gwinnup for surfacing this repository in response to a discussion on social media.[16]

Limitations

Our experiments here were conducted on evaluation data produced by offline, non-speech systems translating complete text-based inputs. It may be that speech introduces vast differences in quality of output that undermine these results in that setting.

[16] https://x.com/mjpost/status/1775228566411620713

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Antonios Anastasopoulos, Loïc Barrault, Luisa Bentivogli, Marcely Zanon Boito, Ondřej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Clara Emmanuel, Yannick Estève, Marcello Federico, Christian Federmann, Souhir Gahbiche, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, and 24 others. 2022. Findings of the IWSLT 2022 evaluation campaign. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 98–157, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Thamme Gowda, Roman Grundkiewicz, Elijah Rippeth, Matt Post, and Marcin Junczys-Dowmunt. 2024. Py-Marian: Fast neural machine translation and evaluation in python. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 328–335, Miami, Florida, USA. Association for Computational Linguistics.

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzmán, and Angela Fan. 2022. The Flores-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 10:522–538.

Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. 2019. The FLORES evaluation datasets for low-resource machine translation: Nepali–English and Sinhala–English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6098–6111, Hong Kong, China. Association for Computational Linguistics.

Wenzel Jakob, Jason Rhinelander, and Dean Moldovan. 2016. pybind11 — seamless operability between c++11 and python. Https://github.com/pybind/pybind11.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, and 3 others. 2024a. Findings of the WMT24 general machine translation shared task: The LLM era is here but MT is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1–46, Miami, Florida, USA. Association for Computational Linguistics.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 1–45, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Tom Kocmi, Vilém Zouhar, Christian Federmann, and Matt Post. 2024b. Navigating the metrics maze: Reconciling score magnitudes and accuracies. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1999–2014, Bangkok, Thailand. Association for Computational Linguistics.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Dominik Macháček, Ondřej Bojar, and Raj Dabre. 2023. MT metrics correlate with human ratings of simultaneous speech translation. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, pages 169–179, Toronto, Canada (in-person and online). Association for Computational Linguistics.

Evgeny Matusov, Gregor Leusch, Oliver Bender, and Hermann Ney. 2005. Evaluating machine translation output with automatic sentence segmentation. In *Proceedings of the Second International Workshop on Spoken Language Translation*, Pittsburgh, Pennsylvania, USA.

Pedro Javier Ortiz Su'arez, Benoit Sagot, and Laurent Romary. 2019. Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures. Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-7) 2019. Cardiff, 22nd July 2019, pages 9 – 16, Mannheim. Leibniz-Institut f"ur Deutsche Sprache.

Sara Papi, Peter Polak, Ondřej Bojar, and Dominik Macháček. 2024. How "real" is your real-time simultaneous speech-to-text translation system? *Preprint*, arXiv:2412.18495.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1342–1348, Hong Kong, China. Association for Computational Linguistics.

Patrick Wilken, Panayota Georgakopoulou, and Evgeny Matusov. 2022. SubER - a metric for automatic evaluation of subtitle quality. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 1–10, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Perric Cistac, Clara Ma, Yacine Jernite, Julien Plu, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

	BLEU			
system	before	after	lines	chars
Unbabel-Tower70B	84.9	83.1	73.1	98.7
Dubformer	83.9	82.0	71.1	99.0
TranssionMT	83.5	81.8	72.5	97.1
GPT-4	83.5	81.8	71.0	98.9
ONLINE-B	83.4	81.8	72.6	97.9
Claude-3	83.3	81.2	69.8	98.5
ONLINE-W	83.0	81.1	71.0	98.9
CommandR-plus	83.0	81.3	70.4	98.4
Mistral-Large	82.7	80.4	66.9	98.1
IOL-Research	82.1	79.9	71.2	99.0
Gemini-1	82.1	80.4	69.1	96.7
ONLINE-A	81.5	79.4	71.2	99.0
Aya23	81.4	79.6	70.7	98.6
Llama3-70B	81.2	79.0	69.8	98.1
IKUN	80.6	77.9	63.5	98.7
ONLINE-G	80.2	78.1	71.8	98.9
Phi-3-Medium	79.7	77.5	67.2	99.0
IKUN-C	79.6	77.5	72.4	98.9
CUNI-NL	79.2	76.6	64.2	98.3
AIST-AIRC	73.4	71.0	69.8	98.9
NVIDIA-NeMo	71.3	68.8	60.5	98.7
Occiglot	69.3	64.7	41.3	88.0
MSLC	64.8	62.5	64.2	98.0
TSU-HITs	63.7	59.1	39.5	89.7
CycleL	42.0	40.5	36.4	93.8
CycleL2	42.0	40.5	36.4	93.8

Table 8: COMET22 scores from the original systems (before) and after merging and automatic realignment (after) for the WMT24/en-de systems. %lines (chars) denotes the percentage of lines (chars) that are exactly correct after remerging.

A System-level score detail

In Section 5.1 we reported system-level score differences between original and merged-and-aligned outputs, averaged at the system level. Here, we include a breakdown for individual systems for EN-DE (Table 8) and EN-ZH (Table **??**).