AI-Assisted Human Evaluation of Machine Translation

Vilém Zouhar[★1] **Tom Kocmi**[★2] **Mrinmaya Sachan**[1]

[1]ETH Zürich
{vzouhar,msachan}@inf.ethz.ch

[2]Microsoft
tomkocmi@microsoft.com

Abstract

Annually, research teams spend large amounts of money to evaluate the quality of machine translation systems (WMT, Kocmi et al., 2024a, inter alia). This is expensive because it requires a lot of expert human labor. In the recently adopted annotation protocol, Error Span Annotation (ESA), annotators mark erroneous parts of the translation and then assign a final score. A lot of the annotator time is spent on scanning the translation for possible errors. In our work, we help the annotators by pre-filling the error annotations with recall-oriented automatic quality estimation. With this AI assistance, we obtain annotations at the same quality level while cutting down the time per span annotation by half (71s/error span → 31s/error span). The biggest advantage of the ESA^{AI} protocol is an accurate priming of annotators (pre-filled error spans) before they assign the final score. This alleviates a potential automation bias, which we confirm to be low. In our experiments, we find that the annotation budget can be further reduced by almost 25% with filtering of examples that the AI deems to be likely to be correct.

1 Introduction

The quality of machine translation (MT) systems is periodically evaluated by academic and industry teams to measure progress and inform product deployment decisions. This undertaking at scale, such as the WMT campaigns (Kocmi et al., 2023, 2024a, inter alia), is extremely expensive. For high-quality systems, expensive high annotation quality is increasingly required to distinguish which system is truly better. Despite recent advancements in automated metrics (Freitag et al., 2023), the metrics remain misaligned with the ideal measure of text quality and human evaluation remains the most accurate, reliable, and ultimate standard.

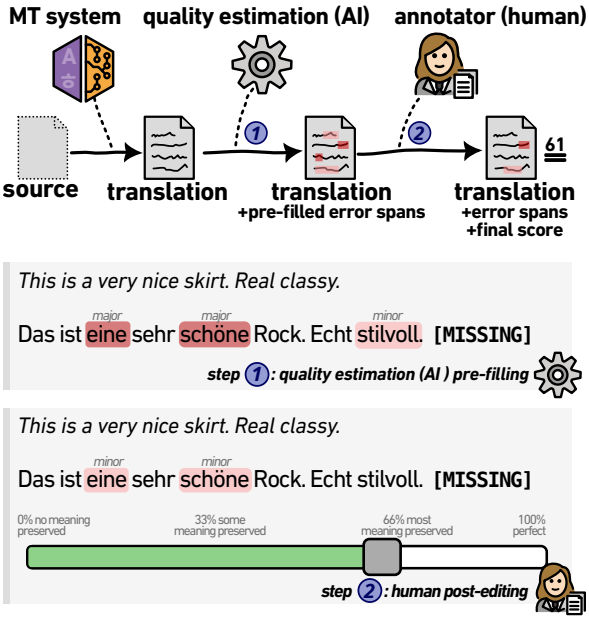

Figure 1: The pipeline (top) and annotation user interface (bottom) with Error Span Annotation pre-filled with AI. In the example, the user: (1) lowered the severity of the gender agreement error, (2) removed incorrectly marked error span, and (3) assigned the final score.

Human evaluation protocols range from ranking different system outputs against each other (Novikova et al., 2018), to assigning scores (direct assessment, DA, Graham et al., 2015), or marking specific error spans, types, and their severities (Multidimensional Quality Metrics, MQM, Lommel et al., 2014; Freitag et al., 2021). Kocmi et al. (2024b) simplified the MQM protocol into Error Span Annotation (ESA), which focuses on the error span severities and not the actual error types. At the end, the annotators additionally assign a final score to the translation. The ESA protocol thus combines the objective diagnostic qualities of MQM (error spans), with the speed and evaluation focus of DA (scoring). One of the problems of all the existing annotation protocols is either their very high cost or low quality. In this work, our aim is to make the MT evaluation process with ESA less expensive.

Human translation already benefits from human-

★Equal contributions.
[0]Code and collected data at:
github.com/wmt-conference/ErrorSpanAnnotation

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 4936–4950
April 29 - May 4, 2025

AI collaboration (Zouhar et al., 2021). In this work, we propose that human evaluation of MT can benefit from AI assistance in a similar way. Despite the risk of automation bias (blind acceptance of AI suggestions), human-AI collaboration can be faster and more accurate than human or AI alone (Bondi et al., 2022). Thus, instead of showing annotators just the source and the system translation, we prefill the translation with error annotations from an AI system (Figure 1 bottom). This is motivated by a lot of human labor being spent on finding possible errors, and we fill this with a recall-focused quality estimation system that produces error annotations. The users still edit the error spans, but now spend less effort scanning the translation for possible errors. This setup, which we call **ESA$^{\text{AI}}$**, is enabled by advancements in quality estimation systems (Guerreiro et al., 2023; Fernandes et al., 2023; Kocmi and Federmann, 2023), which provide accurate initial error spans. The advantage of ESA$^{\text{AI}}$ comes not only from the error span suggestions but also from priming the user with possible translation errors before assigning the final score.

To test our setup, we conduct an annotation campaign for translation evaluation with the ESA and ESA$^{\text{AI}}$ protocols. We compare these protocols on speed, inter-/intra-annotator agreement, quality control success, but also on a new meta-metric subset consistency accuracy.

Key findings

The ESA$^{\text{AI}}$ protocol yields on average 1.6 error spans per translation segment, in contrast to 0.5 for human-only ESA. Although the overall ESA$^{\text{AI}}$ annotation time is only slightly lower than that of ESA (58s→52s/segment), ESA$^{\text{AI}}$ halves the time per error span annotation (71s→31s/error span). This is because the output of ESA$^{\text{AI}}$ has more than three times the error spans per segment than in ESA.

In most of the cases where the AI did not predict any errors, the annotators did not add any new error span, confirming high recall of the AI. We also find that we can prefilter such examples from the evaluation, save up to 24% of the budget, and the evaluation results will be almost identical. In addition, because of the unified priming, the annotators also become more self-consistent and have higher inter-annotator agreement, suggesting higher annotation quality. Ultimately, this allows for a lower number of annotations required to arrive at the same system ranking (high subset consistency accuracy).

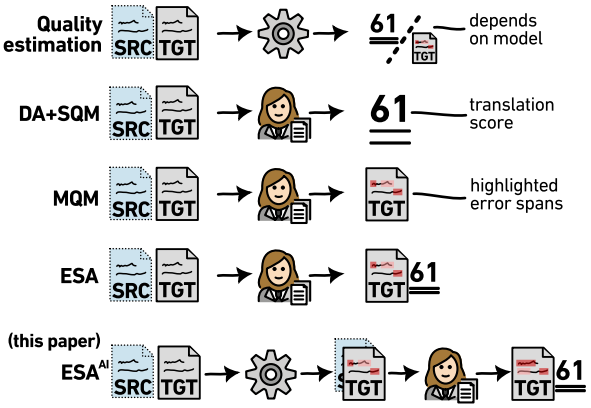

Figure 2: Overview of inputs and outputs of various MT evaluation approaches. Quality estimation (QE) is automated and produces for each segment either a single score or a list of errors. DA+SQM, MQM, ESA and ESA$^{\text{AI}}$ are human annotation protocols. ESA$^{\text{AI}}$ (this paper) is semi-automated and happens in two-steps: quality estimation pre-annotation and human annotation.

2 Related Work

Human evaluation. One of the goals of MT evaluation is to compare systems to inform decisions such as which system to deploy or which machine learning method works the best. There are two ways in which to evaluate translation quality: with automated metrics or with human labor. Reference-based metrics compare the system translation to the gold human translation. They do not always correspond to the human perception of quality and can also introduce evaluation bias (Freitag et al., 2020, 2023; Zouhar and Bojar, 2024). Reference-less approaches, known as **quality estimation (QE)**, do not have the reference bias problem, but also do not always correlate with human judgement (Freitag et al., 2023; Zouhar et al., 2024; Falcão et al., 2024) because the task is more difficult. In higher-stakes settings, human annotators are thus always employed to reliably judge the translation quality.

We depict the various human evaluation protocols in Figure 2. The simplest option for human evaluation is to show annotators the source and the translation and ask them to give a number from 0 to 100 (DA and DA+SQM, Graham et al., 2015; Kocmi et al., 2022). This has the issue of low reliability and agreement. To make annotations more objective, one can ask annotators to mark specific errors in the translation (Multidimensional Quality Metrics, **MQM**, Lommel et al., 2014; Freitag et al., 2021). The marking is done based on their **severity** (e.g. minor or major) but also type (e.g. "inconsistent terminology"). This requires well-trained

professional annotators and is thus expensive. In addition, this protocol does not yield scores, but only error spans, which are turned into the final score with a hand-crafted formula that can introduce additional problems. With some exceptions, the score computation from spans is a sum across all errors with -1 for minor and -5 for major.

Error Span Annotation (ESA). To simplify the MQM process and align it with the goal of objective translation quality assessment, Kocmi et al. (2024b) proposed **ESA**, which asks the annotators to provide only the error severity (not its type) but also a final translation score. Because the error type is not required, the whole annotation is faster and non-experts can be employed because the annotators do not need to know the error type ontology. This combines both DA and MQM in that the annotators are primed with their marked errors to provide high-quality final scoring. The modalities are depicted in the penultimate row in Figure 2.

AI Assistance. Previous work shows that annotators can benefit from AI assistance (Devarajan et al., 2023; Pavoni et al., 2022). However, the use of AI in evaluation is not straightforward because AI might bias the user or induce overreliance (Buçinca et al., 2021) where the annotator mindlessly accepts AI suggestions. This can happen because annotators usually have a financial incentive to optimize their work. In addition, Veselovsky et al. (2023) showed that up to 46% of the annotators used LLMs for summarization. Including AI assistance in the annotation directly could therefore decrease the use of undisclosed tools. See Figure 2 for a description of how this combines the quality estimation pipeline and the ESA protocol.

Quality estimation. Our AI assistance relies on a quality estimation (QE) system that marks error spans in the output. Specifically, given the source and only the translation (i.e., not the reference), the QE produces error span annotations (see Figure 2). Because it is not dependent on the reference, it can also be used in more setups, e.g. where the reference is only being created through this process. Despite the history of quality estimation, such an explainable QE has become popular only recently (Fomicheva et al., 2021). The most popular QE systems are xCOMET (Guerreiro et al., 2023), AutoMQM (Fernandes et al., 2023) and GEMBA (Kocmi and Federmann, 2023).

GEMBA, the QE system that we use, is based on prompting a GPT-4 model and therefore easily adaptable to new scenarios. With a one-shot example in the target language, the model is prompted to provide a list of MQM-like errors. We only use the error spans and severities of this model. See the full prompts in Appendix C.

The QE system is not always correct, but the output is vetted by a human annotator. Compared to humans, the QE system is recall-focused, thus erring on the side of highlighting spans that are not erroneous. Removing false positives is easier and faster for a human annotator than scanning the whole translation for false negatives. The QE thus still offloads some of the work that a human would do and better primes the annotators for final score evaluation.

3 Machine Translation evaluation with Human-AI collaboration

With high-quality machine translation systems, distinguishing which one is the best is increasingly difficult, requiring experts to annotate more and more samples. Some parts of the human expert evaluation do not require full attention or can be automated. With this, we reframe human evaluation as a computer-assisted annotation task to allow for future-proof scaling where competing systems' quality requires more evaluated samples.

We now describe the technical details needed for exact replication of the study.

Pipeline. We implement our study in Appraise (Federmann, 2018) and use GEMBA, a GPT-based quality estimation system. We adapt the Error Span Annotation (ESA) protocol (Section 2), where errors are marked on character level and annotated as either minor or major.[1] The initial error markings are done by the AI and then post-edited by annotators. Subsequently, the annotators manually assign a final score on the scale from 0% to 100% (not with AI) ranging from "no meaning preserved" to "perfect". See interface screenshot in Appendix Figure 10 and guidelines in Appendix B. The error annotation part thus works as a primer for the annotators to give more accurate scores. The complete pipeline is shown in Figure 1 (top). We run the ESA^{AI} setup twice with a different set of annotators to be able to determine the inter-annotator agreement and annotation stability. Finally, we request about 30% of annotators to redo their work

[1] **Minor**: style/grammar/lexical choice could be better; **Major**: changes meaning, lowers usability. See Appendix B.

two months later to estimate the intra-annotator agreement, also known as self-consistency. We hire 21 annotators that are professional translators and native in the target language, German.

Dataset and collected data. We use the data of WMT23 Metrics Shared Task (Freitag et al., 2023) which has been annotated with MQM and ESA. The Conference on Machine Translation annually asks research and industry teams to submit their machine translation systems. These systems are then evaluated with human experts to determine the final system ranking. This ranking is useful, among other things, for measuring research trends and overall improvements. However, because the submitted systems are state-of-the-art and recently close to human quality, arriving at the ranking requires more and more annotations. This motivates finding annotation protocols that speed up the annotations without sacrificing quality.

For maximum compatibility, we use the set-up of Kocmi et al. (2024b). We focus on English→German where 13 translation sets were submitted, one of which is the human reference translation and others machine translation systems. For each set, we have 207 segments (average 18 words per segment) from 74 source documents. Each annotator is assigned a number of segments from various sets and evaluates them with ESA or ESA^{AI}.

4 Analyzing ESA^{AI} Efficacy

To evaluate the new ESA^{AI} annotation pipeline, we consider two main aspects: (1) the annotation process, including its reliability and human effort, and (2) its usefulness for machine translation systems comparison and costs.

4.1 ESA^{AI} Evaluation Process

Collected data distribution. We first examine the high-level distribution of the data collected in Table 1. For ESA^{AI}, the total number of annotated error spans is three times higher than for ESA, which is due to the high number of annotations suggested by the QE system. The split between minor and major errors is similar, although ESA^{AI} annotators prefer major errors as opposed to ESA, even slightly more than those produced by the QE system. Finally, the overall translation score is lower for ESA^{AI} than for ESA alone. This is potentially caused by the priming effect of initially annotated error spans by the QE which highlight the negative aspects of the translation.

	#errors	Minor/Major	Score
ESA	0.45	63% / 37%	81.8
ESA^{AI}	1.63	54% / 46%	76.7
QE (automated)	1.51	55% / 45%	×

Table 1: Average number of error spans and scores across ESA, ESA^{AI}, and the QE system (automated). Because of the pre-annotations, the output of ESA^{AI} is much more errors than ESA alone.

Operation		Frequency
Severity change		12.0%
	Increase severity	60.0%
	Decrease severity	40.0%
Move span ≤5		13.1%
Move span ≤10		17.2%
Move span ≤20		23.3%
Resize	Increase error span size	21.5%
	Decrease error span size	78.5%

Table 2: Distribution of two ESA^{AI} post-editing types: changing the severity, and moving the error span. A span is considered to be *moved* if the distance between old and new endpoints is at most 5, 10, or 20 characters. Many of the QE errors are only misplaced or have the wrong severity. See specific cases in Example 1.

What post-edits do annotators make? Not all post-editing operations are of equal value. For example, moving the error span by a few characters to the left is less important than adding a new error span for a missing translation. We point out two post-editing types: (1) changing the error span severity, and (2) editing the error span boundaries (Table 2). In 11% of the cases, the users only changed the severity. This is important from the workflow perspective because it only requires clicking on the error span. In many cases, the error span was only moved. Time-wise, this is more expensive because it requires the original error span to be removed and a new one created in its place. This operation can be skipped because it does not contribute to the ESA score. Therefore, the annotators could be instructed more specifically not to try to post-edit errors as long as they are approximately correct. See Example 1 for post-editing types.

Do annotators blindly accept AI hints? Gradual overreliance (Holford, 2022) is a type of habituation or automation bias that arises through repetition of non-problematic examples, such as cancer

Increase severity	**Source**	The physics are terrible and the people that created the game won't do anything about it
	QE	Die Physik ist schrecklich und die Leute, die das Spiel entwickelt haben, werden nichts dagegen tun
	ESA^{AI}	Die Physik ist schrecklich und die Leute, die das Spiel entwickelt haben, werden nichts dagegen tun
Decrease severity	**Source**	Will not buy Mr. Coffee again
	QE	Ich kaufe Mr. Kaffee nicht mehr.
	ESA^{AI}	Ich kaufe Mr. Kaffee nicht mehr.
Move	**Source**	However, I hate classes on fine arts and literature, and my school history bears it out.
	QE	Aber ich hasse Kunst und Literatur, und meine Schulgeschichte bestätigt es.
	ESA^{AI}	Aber ich hasse Kunst und Literatur, und meine Schulgeschichte bestätigt es. [missing]
Resize	**Source**	[. . .] I'm not sure if that would work for this.
	QE	[. . .] ich bin mir nicht sicher, ob das für diesen Zweck funktionieren würde.
	ESA^{AI}	[. . .] ich bin mir nicht sicher, ob das für diesen Zweck funktionieren würde.

Example 1: Several post-editing operations from the collected data. Changing the severity (minor and major) is a very fast operation (only clicking the span), while moving and resizing are slow (removing the error span and creating a new one in its place takes up more of the annotator's time).

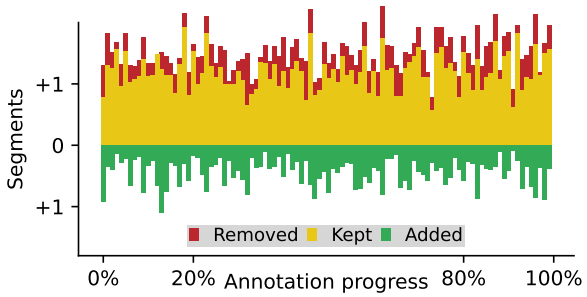

Figure 3: Number of removed/kept/added error spans from the QE system with respect to annotator progress. The amount and type of work remains constant.

SRC: *Sie haben gestern das Treffen wieder verschoben.*
TGT: *He postponed the meeting again yesterday.*
TGT^{P}: *He postponed the meeting squirrels are never.*

Example 2: An example of a perturbed translation **TGT^{P}** based on the original system translation **TGT**. The QE system correctly annotated the error span he (correctly the pronoun is *they*) but the perturbed part is left intentionally unannotated as an attention check.

diagnosis which is dominantly negative. Especially when there are no immediate repercussions, the annotator might be tempted to only confirm the AI suggestion without actually doing any post-editing work. As a result, they would either confirm a span that is not an error or miss part of the translation that is not highlighted by AI but is, in fact, erroneous. We first examine this through the perspective of changes in annotator's behavior through the annotation progression. In Figure 3 we show that the annotators make the same number of edits at the beginning as at the end of the task. This excludes the possibility of a learned automation bias. Next, we examine whether annotators are not already overreliant on AI from the beginning.

Do annotators pay attention? Attention checks, stimuli where we apriori know the correct annotation, are a mechanism to verify that the annotator does the expected job. We use attention checks where the translation is intentionally malformed, but the QE system does not show an error (Example 2). Per each 100 segments to annotate, there are 12 attention checks in total, each with one perturbed span. Each annotator sees both the attention check and the translation original, randomly shuffled. This way, we can compare the annotator's score between the perturbed and non-perturbed versions. The range for passing attention checks (on score, error count, or error highlight level) for ESA is 65% and for ESA^{AI} 69% (Table 3, similar to Kocmi et al., 2023). This is despite ESA^{AI} being at a disadvantage because the segments, as in Example 2, contain errors that are strictly not highlighted by the QE system. Therefore, the pertrubed examples were even more out-of-distribution and the attention of the annotators in-distribution is likely higher.

Do AI mistakes affect annotators? Showing incongruent examples, where AI predictions are clearly wrong, has the potential to reduce the user's trust in AI and their subsequent collaboration (Dhuliawala et al., 2023). In our case, such examples are the attention checks in which the AI intentionally misses the perturbed part. To measure the effect of incongruent examples, we look at docu-

		Original	Perturbed	OK
ESA	Score	79.5	52.6	86%
	Span count	0.85	1.86	54%
	Perturbation marked			56%
ESA^{AI}	Score	75.8	52.6	76%
	Span count	2.19	4.48	61%
	Perturbation marked			71%

Table 3: Annotations assigned to perturbed attention check items (either scores or number of spans). **OK** is percentage in how many cases the non-perturbed item received a higher score or had fewer error spans, and how often the pertrubed span was marked by the annotator.

ments directly preceding and following the attention check. In the document directly before 84% of AI-suggested spans are accepted. In contrast, documents directly after the attention check have only 73% of acceptance of AI-suggested spans. This is a slight decrease in trust, but does not render the collaboration ineffective. It also shows that the annotators are sensitive to possible AI mistakes.

How long do annotations take? One of the motivations for the AI-assisted setup is to speed up annotations and reduce costs. The variance in individual annotator time can be explained by how much they post-edited the QE system's error span annotation (see Figure 4). In this aspect, ESA^{AI} has more variance than ESA, but can also be more controlled and constrained by instructing the annotators what the expected post-editing level is. Per segment, ESA^{AI} annotators required 52s while ESA required 58s. In addition, the time is 71s per single error span for ESA but 31s per single error span for ESA^{AI}, making the latter more efficient in detailed annotation.

Do annotators agree? For a robust and objective annotation protocol, the scores by two independent annotators should be similar and not subjective. To test this, we ran the annotations again with different annotators. Table 4 shows that ESA^{AI} has a much larger agreement between the annotators. For the MQM-like score computation from spans, this is due to the bias by the pre-filled error spans. Still, the agreement is much higher also for the direct scoring, likely due to the unified priming of the annotators. This is consistent with much higher ESA^{AI} *intra*-annotator agreement, i.e. how much annotators agree with themselves.

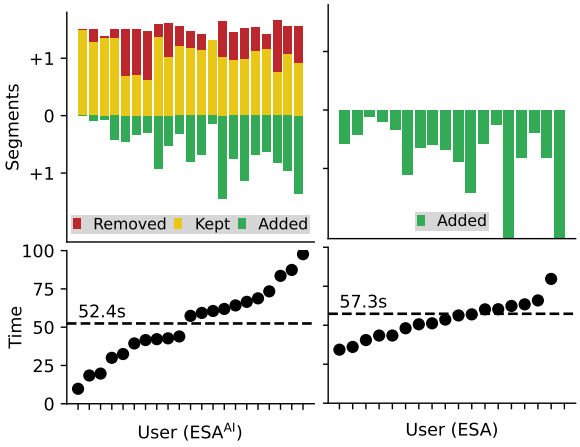

Figure 4: Annotation actions (remove/keep/add an error span) and time per segment. Each dot and bar is an annotator (sorted by time).

	inter-annotator		intra-annotator	
Scoring	**ESA**	**ESA^{AI}**	**ESA**	**ESA^{AI}**
direct score	0.376	0.533	0.222	0.486
from spans	0.327	0.671	0.282	0.689

Table 4: Inter-annotator and intra-annotator agreement with direct scores and scores computed from error spans with MQM formula, as measured with Spearman correlation. ESA^{AI} from spans have the highest inter-annotator agreement, which is however caused by the the QE system's pre-filling. Still, the scores from ESA^{AI}, solely by humans, have the highest inter-annotator and intra-annotator agreement. See visualization in Appendix Figure 9.

Do annotators become faster? With most annotations tasks, the annotators *learn* to be faster. Although the speed-up occurs throughout the entire annotation, it is mostly present in the first 15% of segments (green box in Figure 5). The ESA^{AI} annotators get 1.87s faster with every segment, which is comparable to ESA. This effect is present despite the ESA annotators being at an advantage because there were more ESA^{AI} annotators, and thus each ESA^{AI} annotator individually processed fewer segments, having less time to learn. In addition, users in the post-editing task are more consistent. For ESA, the user's absolute deviation from their personal average is 43.3s, while for ESA^{AI} this is only 32.1s. This makes the human effort more consistent and predictable but also shows that the nature of the annotation task changes.

Why do some segments take longer than others? Being able to predict the expected annotation time for a segment can lead to a more efficient distri-

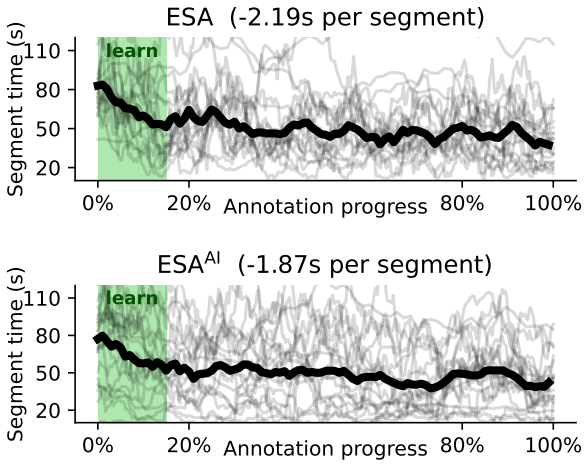

Figure 5: Time per segment with respect to progression in the annontation. Each annotator is the gray faint line and their average is in black. The lines are smoothed with a window of size 15 segments. We also compute the average speed at the beginning and at the end, which yields the *learned speedup*. This is how much the annotator speeds up per working on one segment.

	MQM	ESA	ESA^{AI}
Progress	-0.12	-0.13	-0.13
Translation word count	0.30	0.19	0.16
QE error spans	0.12	0.07	0.12
Error spans	0.06	0.04	0.12
Score	-0.07	-0.03	-0.08
Document size	-0.14	-0.17	-0.17

Table 5: Individual Pearson correlation between features and annotation times. The higher the absolute value, the more it affects the annotation time.

bution and planning, for example, when selecting which segments to annotate at all. We examine the correlations between the features and the segment-level time in Table 5. The number of words in the translation, together with the number of error spans, is a strong predictor of annotation time. For MQM this is the highest, which can be explained by each error span requiring the most work in the MQM annotation scheme because the annotators have to also assign the error type. The longer the whole document (number of surrounding translation segments), the lower the annotation time, which is likely due to shared context, so the annotator does not have to switch between domains and contexts. In contrast, the ESA^{AI} annotators are slightly less affected by the translation length in contrast to ESA because the error spans are pre-highlighted.

	ESA	ESA^{AI}	MQM	QE
MQM^{WMT}	0.240	0.292	0.239	0.416

Table 6: Kendall τ_c segment-level correlations between evaluation protocols. ESA and ESA^{AI} use direct scores.

4.2 ESA^{AI} for Evaluation of WMT Systems

Our goal is for ESA^{AI} to be as reliable as or more reliable than ESA in ranking MT systems. We consider MQM^{WMT} collected by Freitag et al. (2023) as the human gold standard and show the system-level correlations with our protocol in Appendix Figure 8. Both ESA and ESA^{AI} have similar correlations with MQM^{WMT}, justifying our setup. In Table 6 we show that this protocol does not stray far from existing ones in terms of segment-level rating. Many of these cross-protocol correlations are on part with inter-annotator agreement, which is naturally the upper bound. In particular, ESA^{AI} has a higher correlation than ESA or MQM by Kocmi et al. (2024b) alone.

Can cost be further lowered? The goal is to speed up the annotation process without sacrificing quality. This can be achieved by removing, or automating, redundant decisions and actions on the annotator's side. In this segment, we do so by skipping high-quality translations for which we can predict that the annotator's would not mark any error spans.

Our QE system, GEMBA, is recall-focused, and therefore the occurrence of "false positive" error spans is low. In 89% of the cases, the QE marked the spans as having 0 errors and retained 0 errors after the annotation (first row in Table 7), and these segments have an average score of 95. This makes it possible to also use the QE as a prefiltering step. If we replace all such segments with 100 (not to overfit), all but one system comparisons remain the same (Figure 6, left). Alternatively, one can also exclude segments for which the QE marks 0 errors for most systems, which has the advantage that we do not alter the data. For this method, again all but one system comparison would be the same (Figure 6, right). Pre-filtering can thus result in almost 25% budget saving (~52 segments per system).

How many annotations are needed? Comparing the quality of two annotation protocols is not straightforward because of the absence of a gold standard. We now take a practical perspective

QE	——Removed——			No edit	——Added——			
#err. (freq.)	=2	=1	=0		=0	=1	=2	≥3
0 (23.8%)	0%	0%	100%	88%	88%	8%	2%	2%
1 (38.0%)	0%	28%	72%	62%	81%	14%	3%	3%
2 (18.8%)	15%	16%	69%	54%	71%	13%	9%	7%
3 (10.4%)	11%	20%	62%	51%	68%	16%	7%	10%
4 (8.9%)	11%	13%	69%	54%	65%	13%	10%	12%

Table 7: Distribution of error span post-editing based on original QE-reported error spans (2nd column). Percentages in the table are proportions within the number of the QE error spans. For example, second row shows that 62% of segments with exactly one QE error span received no post-editing from annotators and in 28% the annotators removed the single error. ESA is comparable to ESA$^{\text{AI}}$.

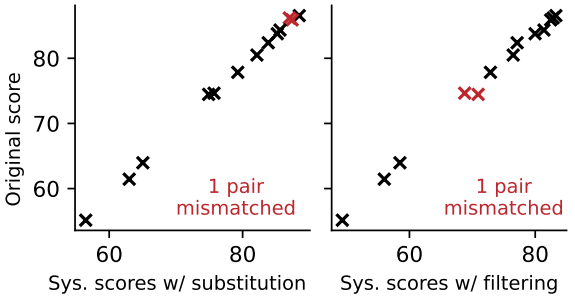

Figure 6: Average system scores with either substitution or filtering of segments with no QE errors. Each cross is a single system. The two red pairs of crosses are the single pairs of systems whose ordering changes when substitution or filtering is applied.

where we desire a protocol that finds the true system ordering with as few examples as possible. This does not require any gold standard to compare with. Instead, this approach compares a subset of the annotations of a protocol with the full set of annotations from the same protocol.

With a sufficiently large evaluation, even noisy annotation schemes yield the true system ordering. Conversely, only robust annotation schemes yield this ordering on a small scale. We formalize this in Appendix A to show that ESA$^{\text{AI}}$ leads to better annotations than ESA or MQM. We measure the accuracy of the ordering $(m_1 >_I m_2)$ of systems $(\mathcal{M})$ computed on a subset of segments (I) against the ordering given by the full data $(a_{m_1} > a_{m_2})$:

$$\text{ACC}(I) \stackrel{\text{def}}{=} \sum_{m_1, m_2 \in \mathcal{M}} \frac{\mathbb{1}[(m_1 >_I m_2) \Leftrightarrow (a_{m_1} > a_{m_2})]}{|\mathcal{M}|^2} \quad (1)$$

This simulates the setup where we wish to save costs with fewer number of annotations but care about arriving at the true system ordering. We refer to this metric as **subset consistency accuracy**, similar to split-half reliability, and in Appendix A justify that annotation protocols with lower noise reach higher accuracies. For each subset size, e.g. 30 source sentences, we select 1000 random subsets and compute the system ranking accuracy.

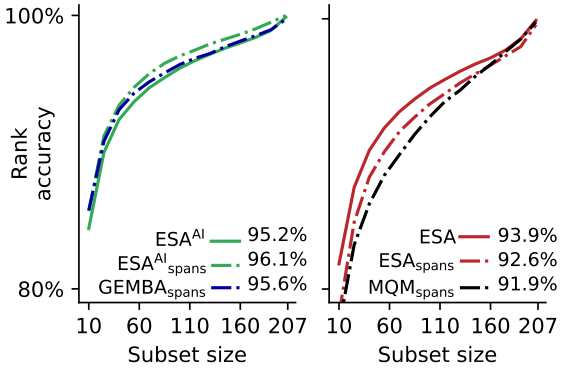

Figure 7: Subset consistency accuracy of a system ranking only on a subset against ranking on full data. Percentages are averages across visible subset sizes, which corresponds to the area under the curve. See figure in tabular form in Appendix Table 8.

The results in Figure 7 suggest that the QE system, GEMBA, alone very consistent. However, this is solely because it can be perceived as a single annotator and thus there is no inter-annotator confusion. Overall and among human evaluation protocols, ESA$^{\text{AI}}$ with scores has the highest stability and quality of scores. This is not the result of automation bias because only the annotators themselves assign the final score. In practice, this would mean that one can annotate fewer examples (e.g. 2000 for ESA$^{\text{AI}}$) to obtain the same system-level accuracy as lower-quality protocol (e.g. 2500 for ESA), thus further lowering the costs.

5 Conclusion

Our AI-assisted protocol of human evaluation of MT is faster and cheaper. This protocol is more robust and self-consistent and increases inter-annotator agreement by priming the annotators with pre-annotated error spans. Our analysis also shows that the annotators did not overrely on the AI and were able to maintain evaluation quality. The inclusion of AI in evaluation also opens many options for further evaluation economy by reducing the test set size requirements. To this end, we introduce subset consistency accuracy, which quantifies how many annotations could be saved while arriving at a similar final system ordering.

Limitations

Despite the advantages in lower costs per error span of the presented setup, we urge practitioners not to use this approach when metrics evaluation is one of the expected tasks due to the particular bias to the used metric in the setup. The intended application of this pipeline is purely a more efficient evaluation of the quality of the machine translation system.

Both ESA^{AI} and GEMBA rank GPT-4-5shot as the best system, a system that uses the same LLM to translate sentences as we use to generate for GEMBA. This indicated a weakness that our approach is biased towards systems built on top of the same underlying LLM. Liu et al. (2023) described this phenomenon when the same system used to generate output should not be used to also evaluate them. This issue could be mitigated by using two different LLMs to generate error spans.

Lastly, for QE we use GEMBA, a GPT4-based system, for the quality estimation and work with WMT 2023 data. Unfortunately, we cannot exclude the possibility of the QE system being trained on this data, though the texts and scores are kept in two separate large files with non-linear mappings.

Ethics Statement

The annotators were paid a standard commercial translator wage in the respective country. No personal data was collected and the data shown to the annotators was screened for potentially disturbing content.

References

Elizabeth Bondi, Raphael Koster, Hannah Sheahan, Martin Chadwick, Yoram Bachrach, Taylan Cemgil, Ulrich Paquet, and Krishnamurthy Dvijotham. 2022. Role of human-AI interaction in selective prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36/5, 5286–5294.

Zana Buçinca, Maja Barbara Malaya, and Krzysztof Z Gajos. 2021. To trust or to think: Cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–21.

Ganesh Gopal Devarajan, Senthil Murugan Nagarajan, Sardar Irfanullah Amanullah, SA Sahaaya Arul Mary, and Ali Kashif Bashir. 2023. AI-Assisted deep NLP-Based approach for prediction of fake news from social media users. *IEEE Transactions on Computational Social Systems*.

Shehzaad Dhuliawala, Vilém Zouhar, Mennatallah El-Assady, and Mrinmaya Sachan. 2023. A diachronic perspective on user trust in AI under uncertainty. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 5567–5580. Association for Computational Linguistics.

Júlia Falcão, Claudia Borg, Nora Aranberri, and Kurt Abela. 2024. COMET for low-resource machine translation evaluation: A case study of English-Maltese and Spanish-Basque. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 3553–3565, Torino, Italia. ELRA and ICCL.

Christian Federmann. 2018. Appraise evaluation framework for machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, 86–88. Association for Computational Linguistics.

Patrick Fernandes, Daniel Deutsch, Mara Finkelstein, Parker Riley, André Martins, Graham Neubig, Ankush Garg, Jonathan Clark, Markus Freitag, and Orhan Firat. 2023. The devil is in the errors: Leveraging large language models for fine-grained machine translation evaluation. In *Proceedings of the Eighth Conference on Machine Translation*, 1066–1083. Association for Computational Linguistics.

Marina Fomicheva, Piyawat Lertvittayakumjorn, Wei Zhao, Steffen Eger, and Yang Gao. 2021. The Eval4NLP shared task on explainable quality estimation: Overview and results. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, 165–178. Association for Computational Linguistics.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 61–71. Association for Computational Linguistics.

Markus Freitag, Nitika Mathur, Chi-kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Tom Kocmi, Frederic Blain, Daniel Deutsch, Craig Stewart, Chrysoula Zerva, Sheila Castilho, Alon Lavie, and George Foster. 2023. Results of WMT23 metrics shared task: Metrics might be guilty but references are not innocent. In *Proceedings of the Eighth Conference on Machine Translation*, 578–628. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, and Nitika Mathur. 2015. Accurate evaluation of segment-level machine translation metrics. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies, 1183–1191. Association for Computational Linguistics.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2023. xCOMET: Transparent machine translation evaluation through fine-grained error detection.

W David Holford. 2022. Design-for-responsible algorithmic decision-making systems: A question of ethical judgement and human meaningful control. *AI and Ethics*, 2(4):827–836.

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, Mariya Shmatova, Steinthór Steingrímsson, and Vilém Zouhar. 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*, 1–46. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, 1–42. 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)*, 1–45. Association for Computational Linguistics.

Tom Kocmi and Christian Federmann. 2023. GEMBA-MQM: Detecting translation quality error spans with GPT-4. In *Proceedings of the Eighth Conference on Machine Translation*, 768–775. Association for Computational Linguistics.

Tom Kocmi, Vilém Zouhar, Eleftherios Avramidis, Roman Grundkiewicz, Marzena Karpinska, Maja Popović, Mrinmaya Sachan, and Mariya Shmatova. 2024b. Error span annotation: A balanced approach for human evaluation of machine translation. In *Proceedings of the Ninth Conference on Machine Translation*, 1440–1453. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2511–2522. Association for Computational Linguistics.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014. Multidimensional quality metrics (MQM): A framework for declaring and describing translation quality metrics. *Tradumàtica*, 12:0455–463.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2018. RankME: Reliable human ratings for natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 72–78. Association for Computational Linguistics.

Gaia Pavoni, Massimiliano Corsini, Federico Ponchio, Alessandro Muntoni, Clinton Edwards, Nicole Pedersen, Stuart Sandin, and Paolo Cignoni. 2022. Taglab: AI-assisted annotation for the fast and accurate semantic segmentation of coral reef orthoimages. *Journal of Field Robotics*, 39(3):246–262.

Parker Riley, Daniel Deutsch, George Foster, Viresh Ratnakar, Ali Dabirmoghaddam, and Markus Freitag. 2024. Finding replicable human evaluations via stable ranking probability. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 4908–4919. Association for Computational Linguistics.

Veniamin Veselovsky, Manoel Horta Ribeiro, and Robert West. 2023. Artificial artificial artificial intelligence: Crowd workers widely use large language models for text production tasks.

Vilém Zouhar and Ondřej Bojar. 2024. Quality and quantity of machine translation references for automatic metrics. In *Proceedings of the Fourth Workshop on Human Evaluation of NLP Systems (HumEval) @ LREC-COLING 2024*, 1–11. ELRA and ICCL.

Vilém Zouhar, Shuoyang Ding, Anna Currey, Tatyana Badeka, Jenyuan Wang, and Brian Thompson. 2024. Fine-tuned machine translation metrics struggle in unseen domains. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 488–500. Association for Computational Linguistics.

Vilém Zouhar, Martin Popel, Ondřej Bojar, and Aleš Tamchyna. 2021. Neural machine translation quality and post-editing performance. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 10204–10214. Association for Computational Linguistics.

Protocol/ method	Subset size			
	10	40	115	190
ESA^{AI}	84.41%	92.38%	96.69%	98.88%
$\text{ESA}^{\text{AI}}_{\text{spans}}$	85.69%	93.43%	97.46%	99.49%
$\text{GEMBA}_{\text{spans}}$	85.73%	93.10%	96.86%	98.94%
ESA	81.86%	90.26%	95.52%	98.52%
$\text{ESA}_{\text{spans}}$	78.11%	88.28%	94.48%	97.94%
$\text{MQM}_{\text{spans}}$	77.19%	86.30%	93.89%	98.50%

Table 8: Specific values of Figure 7. Subset accuracy across annotation schemes. $\text{ESA}^{\text{AI}}{}_{\text{spans}}$ has the highest subset consistency, though this is likely biased by the spans from GEMBA, which as 100% inter-annotator agreement. However, ESA^{AI} (direct scores) is based solely on human scorings, which has the second-highest subset consistency of any protocol.

A Subset Consistency Formalization

This section justifies the setup in Section 4.2 and is reminiscent of the work of Riley et al. (2024) or split-half reliability. A key distinction is that we are considering ranking stability with respect to the protocol itself. We do so by bootstrapping subsets of the data.

Our goal is tho show that a protocol with lower annotation error has higher system-level ranking accuracy. We assume that the annotation schemes are not biased towards a particular system but are noisy. We also assume a simplified model of system performance, where the annotation output $y_{m,i}$ of system m on segment i can be approximated by the system ability a_m (e.g. average across a real life distribution) from which segment-specific variance d_i is subtracted and error term ϵ is added. The annotation output $y_{m,i}$ is dependent on the specific annotation scheme, which is not indicated for brevity. We would like to find the system abilities a_m but we only have access to $y_{m,i}$. This notation can also be extended to a collection of segments I:

$$y_{m,i} = a_m - d_i + \epsilon_{m,i} \tag{2}$$

$$Y_{m,I} = \frac{\sum_{i\in I} y_{m,i}}{|I|} \tag{3}$$

$$= a_m - \frac{\sum_{i\in I} d_i}{|I|} + \frac{\sum_{i\in I} \epsilon_{m,i}}{|I|} \tag{4}$$

On a large enough set of segments with the law of large numbers, we can assume $\frac{\sum_{i\in I} \epsilon_{m,i}}{|I|} \approx 0$ as ϵ is unbiased. If we want to estimate $\epsilon_{m,i}$, we could subtract from sample i the average from all dataset, $Y_{m,D}$. Unfortunately, this would still leave the segment-specific difference d_i:

$$y_{m,i} - Y_{m,D} = -d_i + \epsilon_{m,i} \tag{5}$$

To separate $\epsilon_{m,i}$, we could consider subsets $I \subsetneq D$ for which $\frac{\sum_{i\in I} d_{m,i}}{|I|} \approx 0$ but $\frac{\sum_{i\in I} \epsilon_{m,i}}{|I|} \not\approx 0$. Apart from the difficulty of finding such subsets, our goal is to have a good estimation of the ranking of the systems. For this, we define system ordering $>_I$ given by the observed subset I:

$$m_1 >_I m_2$$

$$\overset{\text{def}}{\Leftrightarrow} \frac{\sum_{i\in I} y_{i,m_1}}{|I|} > \frac{\sum_{i\in I} y_{i,m_2}}{|I|} \tag{6}$$

$$\Leftrightarrow \sum_{i\in I} y_{i,m_1} > \sum_{i\in I} y_{i,m_2} \tag{7}$$

$$\Leftrightarrow a_{m_1} - \sum_{i\in I} d_i + \sum_{i\in I} \epsilon_{i,m_1} > a_{m_2} - \sum_{i\in I} d_i + \sum_{i\in I} \epsilon_{i,m_2} \tag{8}$$

$$\Leftrightarrow a_{m_1} + \sum_{i\in I} \epsilon_{i,m_1} > a_{m_2} + \sum_{i\in I} \epsilon_{i,m_2} \tag{9}$$

Notice that $>_I$ is independent of the segment-specific term d_i because both systems are evaluated on the same segments. We compare this empirical ordering with that of the true system ranking. This is done across a set of systems $\mathcal{M}$ using pairwise accuracy, i.e. how many system pairs are ranked in the same way as by the true system ranking:

$$\text{ACC}(I) \overset{\text{def}}{=} \sum_{m_1,m_2\in\mathcal{M}} \frac{\mathbb{1}[(m_1 >_I m_2) \Leftrightarrow (a_{m_1} > a_{m_2})]}{|\mathcal{M}|^2} \tag{10}$$

With higher accuracy we can assume that the relative ϵ is lower, at least for the purposes of ordering. This is because **if** the accumulated error terms are low (11), the indicator in Equation (10) is true (12), which is **equivalent** to high accuracy (13):

$$\sum_{i\in I} \epsilon_{i,m_1} \to 0 \wedge \sum_{i\in I} \epsilon_{i,m_2} \to 0 \quad \Rightarrow \tag{11}$$

$$\left(a_{m_1} + \sum_{i\in I} \epsilon_{i,m_1} > a_{m_2} + \sum_{i\in I} \epsilon_{i,m_2} \Leftrightarrow a_{m_1} > a_{m_2}\right) \tag{12}$$

$$\Leftrightarrow \quad \text{ACC}(I) \to 1 \tag{13}$$

To obtain ACC, we would need to know if $a_{m_1} > a_{m_2}$. In our setup, we do not know this true ranking and obtaining it would require large-scale super-human annotations. However, for large-enough I, we can assume that $\frac{\sum_{i\in I} \epsilon_{m,i}}{|I|} \approx 0$.

Therefore, for the true ordering, we use the ordering by that particular annotation scheme on all data. Now we established a link between accumulated annotation noise, $\sum_{i \in I} \epsilon_{i,m}$, and accuracy, which we can measure.

The accuracy will be high if the error terms are low and therefore the annotations are of high quality. This can be used to measure the annotation protocol usefulness. In addition, this has practical implications as we could solicit fewer annotations to obtain the same results as if we had more.

B User Guidelines

The following are are annotation guidelines for our two local ESA$^{\text{AI}}$ campaigns, which is closely based on the setup of Kocmi et al. (2024b).

Highlighting errors: Highlight the text fragment where you have identified a translation error (drag or click start & end). Click repeatedly on the highlighted fragment to increase its severity level or to remove the selection.

- **Minor Severity:** Style/grammar/lexical choice could be better/more natural.
- **Major Severity:** Seriously changed meaning, difficult to read, decreases usability.

If something is missing from the text, mark it as an error on the **[MISSING]** word. The highlights do not have to have character-level precision. It's sufficient if you highlight the word or rough area where the error appears. Each error should have a separate highlight.

Score: After highlighting all errors, please set the overall segment translation scores. The quality levels associated with numerical scores on the slider:

- 0%: No meaning preserved: Nearly all information is lost in the translation.
- 33%: Some meaning preserved: Some of the meaning is preserved but significant parts are missing. The narrative is hard to follow due to errors. Grammar may be poor.
- 66%: Most meaning preserved and few grammar mistakes: The translation retains most of the meaning. It may have some grammar mistakes or minor inconsistencies.
- 100%: Perfect meaning and grammar: The meaning and grammar of the translation is completely consistent with the source.

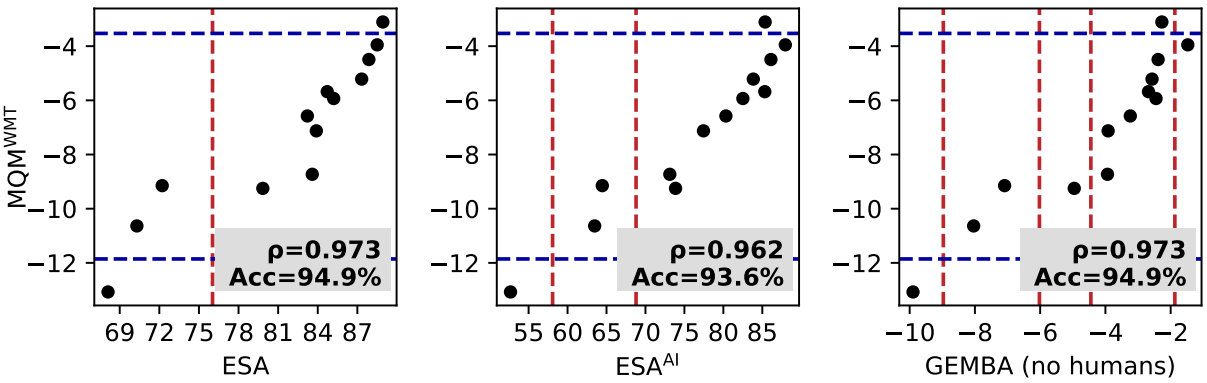

Figure 8: Each point is a system, with original MQM^{WMT} scores on the y-axis against ESA, ESA^{AI}, and GEMBA before post-editing. Stripped lines indicate cluster separations with alpha threshold 0.05. Numbers show Spearman's correlations between the specific protocol and MQM^{WMT}. ESA and ESA^{AI} have comparable system-level accuracy and correlations with MQM^{WMT}, making them equal in quality in this aspect.

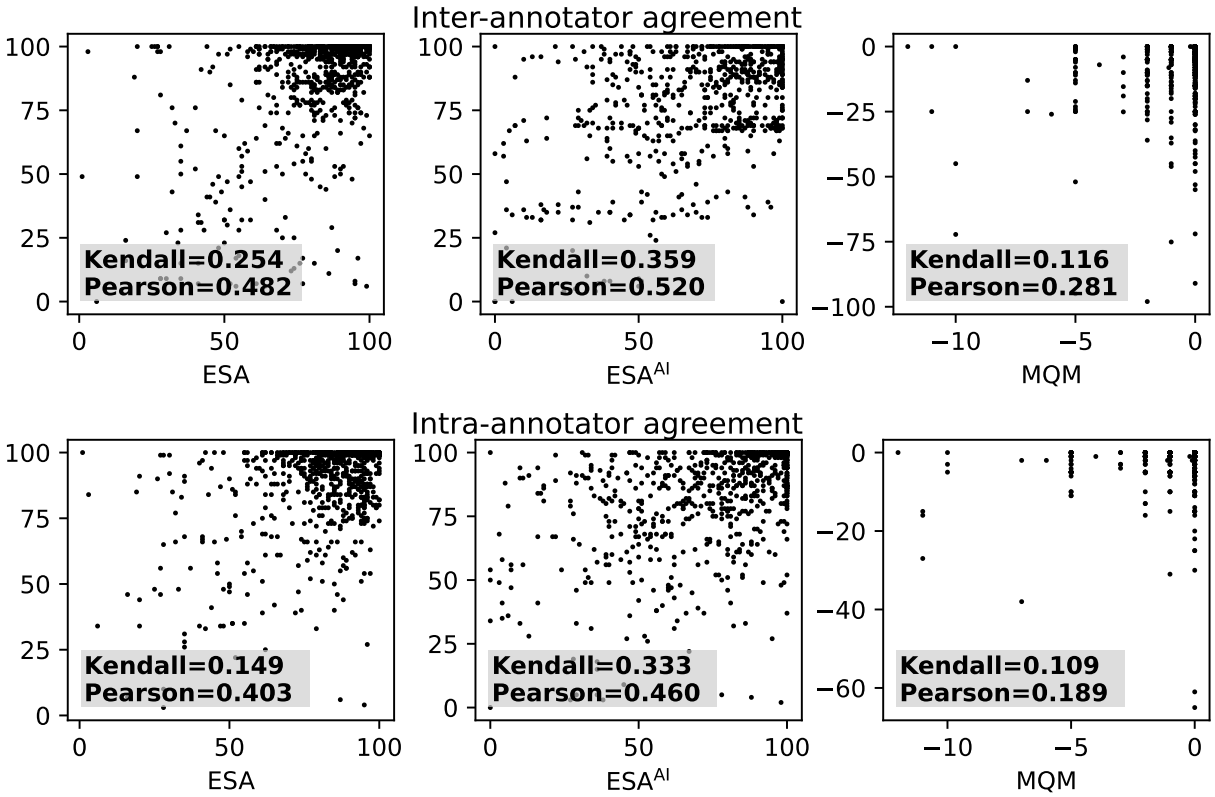

Figure 9: Inter-annotator and intra-annotator agreement. The-intra annotator agreement shows changes in scoring by the same annotator when evaluated again. Each point represents single annotated segment with x-axis being annotator's score assigned one month and y-axis their score assigned two months later. For ESA and ESA^{AI}, the scores are directly from annotators. For MQM, they are computed by the formula. ESA^{AI} has the highest *intra*-annotator and *inter*-annotator agreement, showing another positive aspect of being primed by GEMBA.

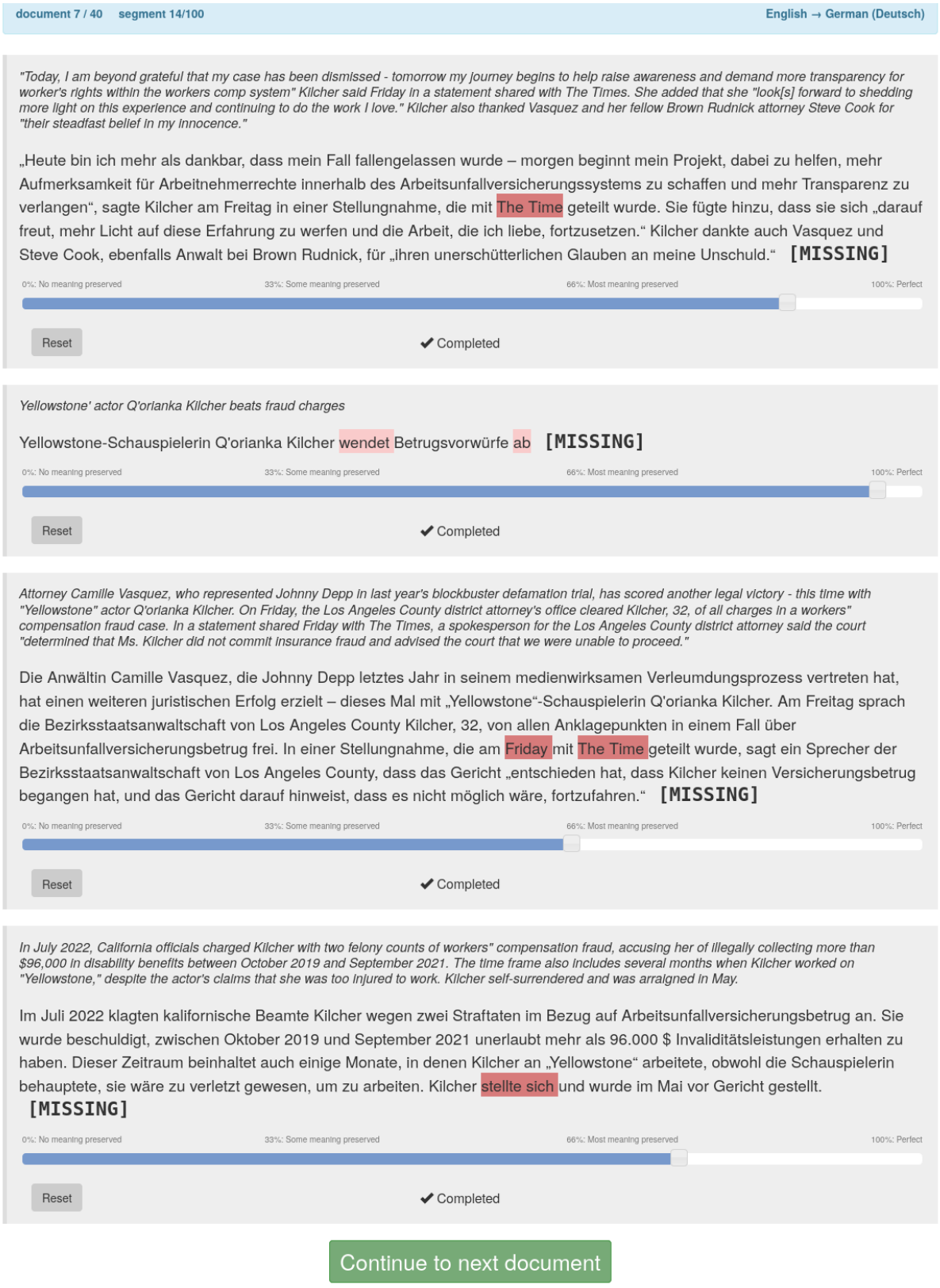

Figure 10: Screenshot of the study interface implemented for Appraise. Multiple segments from a document are shown together for context. The AI suggests the initial error spans which the annotator post-edits and finally adds final score judgment.

C GEMBA Quality Estimator Prompts

```
Your task is to identify machine translation errors and assess the tranlsation quality.

{source_lang} source:
```{source_seg}```
{target_lang} translation:
```{target_seg}```

Based on the source segment and machine translation surrounded with triple backticks,
identify error types in the translation and classify them. The categories of errors are:
accuracy (addition, mistranslation, omission, untranslated text), fluency (character enc-
oding, grammar, inconsistency, punctuation, register, spelling), style (awkward), termin-
ology (inappropriate for context, inconsistent use), non-translation, other, or no-error.

Each error is classified as one of two categories: major or minor. Major errors disrupt
the flow and make the understandability of text difficult or impossible. Minor errors
are errors that do not disrupt the flow significantly and what the text is trying to say
is still understandable.
```

(a) Prompt for annotating error spans (initial step).

```
"source_lang": "English",
"source_seg": "I do apologise about this, we must gain permission from the account holder
    to discuss an order with another person, I apologise if this was done previously,
    however, I would not be able to discuss this with yourself without the account
    holders permission.",
"target_lang": "German",
"target_seg": "Ich entschuldige mich dafuer, wir muessen die Erlaubnis einholen, um eine
    Bestellung mit einer anderen Person zu besprechen. Ich entschuldige mich, falls dies
    zuvor geschehen waere, aber ohne die Erlaubnis des Kontoinhabers waere ich nicht in
    der Lage, dies mit dir involvement.",

"answer: """\
Major:
accuracy/mistranslation - "involvement"
accuracy/omission - "the account holder"
Minor:
fluency/grammar - "waere"
fluency/register - "dir"
"""
```

(b) Prompt for scoring with prior annotations of error spans (example in prompt).

```
Given the translation from {source_lang} to {target_lang} and the annotated error
spans, assign a score on a continuous scale from 0 to 100. The scale has following
reference points: 0="No meaning preserved", 33="Some meaning preserved", 66="Most
meaning preserved and few grammar mistakes", up to 100="Perfect meaning and grammar".

Score the following translation from {source_lang} source:
```{source_seg}```
{target_lang} translation:
```{target_seg}```
Annotated error spans:
```{error_spans}```
Score (0-100):
```

(c) Prompt for scoring with prior annotations of error spans (final step).

Figure 11: Prompts for GEMBA with GPT-4. See the full GEMBA code for ESA. The prompts can be used to first prompt GEMBA to produce the list of translation errors, as in MQM, and then prompt again to score the segments hollistically. For the ESA^{AI} human pre-annotations we only use the first part and only the error severities.
```