As Little as Possible, as Much as Necessary: Detecting Over- and Undertranslations with Contrastive Conditioning

Jannis Vamvas\(^1\) and Rico Sennrich\(^1,2\)

\(^1\)Department of Computational Linguistics, University of Zurich
\(^2\)School of Informatics, University of Edinburgh
{vamvas, sennrich}@cl.uzh.ch

Abstract

Omission and addition of content is a typical issue in neural machine translation. We propose a method for detecting such phenomena with off-the-shelf translation models. Using contrastive conditioning, we compare the likelihood of a full sequence under a translation model to the likelihood of its parts, given the corresponding source or target sequence. This allows to pinpoint superfluous words in the translation and untranslated words in the source even in the absence of a reference translation. The accuracy of our method is comparable to a supervised method that requires a custom quality estimation model.

1 Introduction

Neural machine translation (NMT) is susceptible to coverage errors such as the addition of superfluous target words or the omission of important source content. Previous approaches to detecting such errors make use of reference translations (Yang et al., 2018) or employ a separate quality estimation (QE) model trained on synthetic data for a language pair (Tuan et al., 2021; Zhou et al., 2021).

In this paper, we propose a reference-free algorithm based on hypothetical reasoning. Our premise is that a translation has optimal coverage if it uses as little information as possible and as much information as necessary to convey the source sequence. Therefore, an addition error means that the source would be better conveyed by a translation containing less information. Conversely, an omission error means that the translation would be more adequate for a less informative source sequence.

Adapting our contrastive conditioning approach (Vamvas and Sennrich, 2021), we use probability scores of NMT models to approximate this concept of coverage. We create parse trees for both the source sequence and the translation, and treat their constituents as units of information. Omission errors are detected by systematically deleting constituents from the source and by estimating the probability of the translation conditioned on such a partial source sequence. If the probability score is higher than when the translation is conditioned on the full source, the deleted constituent might have no counterpart in the translation (Figure 1). We apply the same principle to the detection of addition errors by swapping the source and the target sequence.

When comparing the detected errors to human annotations of coverage errors on the segment level (Freitag et al., 2021), our approach surpasses a supervised QE baseline that was trained on a large number of synthetic coverage errors. Human raters find that word-level precision is higher for omissions than additions, with 39% of predicted error spans being precise for English–German translations, and 20% for Chinese–English. False positive predictions can occur especially in cases where the translation has different syntax than the source. We believe our algorithm could be a useful aid whenever humans remain in the loop, for example in a post-editing workflow.

We release the code and data to reproduce our findings, including a large-scale dataset of synthetic coverage errors in English–German and Chinese–English machine translations.\(^1\)

2 Related Work

Coverage errors in NMT Addition and omission of target words have been observed by human evaluation studies in various languages, with omission as the more frequent error type (Castilho et al., 2017; Zheng et al., 2018). They are included as typical translation issues in the Multidimensional Quality Metrics (MQM) framework (Lommel et al., 2014). Addition is defined as an accuracy issue where the target text includes text not present in the source, and omission is defined as an accuracy error.

---

\(^1\)https://github.com/ZurichNLP/coverage-contrastive-conditioning
issue where content is missing from the translation but is present in the source.²

Freitag et al. (2021) used MQM to manually re-annotate English–German and Chinese–English machine translations submitted to the WMT 2020 news translation task (Barrault et al., 2020). Their findings confirm that state-of-the-art NMT systems still erroneously add and omit target words, and that omission occurs more often than addition. Similar patterns can be found in English–French machine translations that have been annotated with fine-grained MQM labels for the document-level QE shared task (Specia et al., 2018; Fonseca et al., 2019; Specia et al., 2020).

Detecting and reducing coverage errors While reference-based approaches include measuring the n-gram overlap to the reference (Yang et al., 2018) and analyzing word alignment to the source (Kong et al., 2019), this work focuses on the reference-free detection of coverage errors.

Previous work has employed custom QE models trained on labeled parallel data. For example, Zhou et al. (2021) use synthetic hallucinations to learn a Transformer to predict the inserted spans. Similarly, Tuan et al. (2021) use an NMT model on synthetically noisy translations. In this paper, we propose a method that is based on off-the-shelf NMT models only.

Other related work has focused on improving coverage during decoding or training, for example via attention (Tu et al., 2016; Wu et al., 2016; Li et al., 2018; among others). More recently, Yang et al. (2019) found that contrastive fine-tuning on references with synthetic omissions reduces coverage errors produced by an NMT system.

3 Approach

Contrastive Conditioning Properties of a translation can be inferred by estimating its probability conditioned on contrastive source sequences (Vamvas and Sennrich, 2021). For example, if a certain translation is more probable under an NMT model when conditioned on a counterfactual source sequence, the translation might be inadequate.

Application to Omission Errors Figure 1 illustrates how contrastive conditioning can be directly applied to the detection of omission errors. We construct partial source sequences by systematically deleting constituents from the source. If the probability score of the translation (average token log-probability) is higher when conditioned on such a partial source, the deleted constituent is taken to be missing from the translation.

To compute the probability score for a translation \( Y \) given a source sequence \( X \), we sum up the log-probabilities for every target token and normalize the sum by the number of target tokens:

\[
\text{score}(Y | X) = \frac{1}{|Y|} \sum_{i=0}^{|Y|} \log p_\theta(y_i | X, y_{<i})
\]

Application to Addition Errors We apply the same method to addition detection, but swap the source and target languages. Namely, we use an NMT model for the reverse translation direction, and we score the source sequence conditioned on the full translation and a set of partial translations.³

³Another possibility would be to leave the translation direction unreversed and to score the partial translations con-
**Potential Error Spans**  In its most basic form, our algorithm does not require any linguistic resources apart from tokenization. For a source sentence of $n$ tokens one could create $n$ partial source sequences with the $i$th token deleted. However, such an approach would rely on a radical assumption of compositionality, treating all tokens as independent constituents.

We thus propose to extract potential error spans from parse trees, specifically from dependency trees predicted by Universal Dependency parsers (de Marneffe et al., 2021), which are widely available. This allows (a) to skip function words and (b) to include a reasonable number of multi-word spans in the set of potential error spans. Formally, we consider word spans that satisfy the following conditions:

1. A potential error span is a complete subtree of the dependency tree.
2. It covers a contiguous subsequence.
3. It contains a part of speech of interest.

For every potential error span, we create a partial sequence by deleting the span from the original sequence. This is still a simplified notion of constituency, since some partial sequences will be ungrammatical. Our assumption is that NMT models can produce reliable probability estimates despite the ungrammatical input.

**4 Experimental Setup**

In this section we describe the data and tools that we use to implement and evaluate our approach.

**Scoring model**  We use mBART50 (Tang et al., 2021), which is a sequence-to-sequence Transformer pre-trained on monolingual corpora in many languages using the BART objective (Lewis et al., 2020; Liu et al., 2020) that was fine-tuned on English-centric multilingual MT in 50 languages. Sequence-level probability scores are computed by averaging the log-probabilities of all target tokens. We use the one-to-many mBART50 model if English is the source language, and the many-to-one model if English is the target language.

**Error spans**  We use Stanza (Qi et al., 2020) for dependency parsing, a neural pipeline for various languages trained on data from Universal Dependencies (de Marneffe et al., 2021). We make use of universal part-of-speech tags (UPOS) to define

---

**Gold Standard Data**  We use state-of-the-art English–German and Chinese–English machine translations for evaluation, which have been annotated by Freitag et al. (2021) with translation errors. We set aside translations by the system *Online-B* as a development set, and use the other systems as a test set, excluding translations by humans. The development set was used to identify the typical parts-of-speech of coverage error spans, listed in the paragraph above.

**Synthetic Data**  We also create synthetic coverage errors, which we use for training a supervised baseline QE system. We propose a data creation process that is inspired by previous work (Yang et al., 2019; Zhou et al., 2021; Tuan et al., 2021) but is defined such that it works for both additions and omissions, and produces fluent translations.

Figure 2 illustrates the process. We start from the original source sentences and create partial sources by deleting randomly selected constituents. Specifically, we delete each constituent with a probability of 15%. We then machine-translate both the original and the partial sources, yielding full and partial machine translations. We retain only samples where the full machine translation is different from the partial one, and can be constructed by addition.

This allows us to treat the full translations as overtranslations of the partial sources, and the added words as addition errors. Conversely, the partial translations are treated as undertranslations of the original sources. Negative examples are cre-
TABLE 1: Segment-level comparison of coverage error detection methods on the gold dataset by Freitag et al. (2021).

| Approach                  | Detection of additions | Detection of omissions |
|---------------------------|------------------------|------------------------|
|                           | Precision  | Recall | F1   | Precision | Recall | F1    |
| **EN–DE**                 |            |        |      |            |        |      |
| Supervised baseline       | 6.9±1.9    | 2.9±0.9| 4.0±1.3| 40.3±5.2  | 6.1±0.1| 10.6±0.2|
| Our approach              | 4.0        | 15.0   | **6.3** | 22.3       | 18.8   | **20.4** |
| **ZH–EN**                 |            |        |      |            |        |      |
| Supervised baseline       | 4.3±0.6    | 4.7±0.7| **4.5±0.6** | 49.6±0.6  | 9.4±1.0| 15.9±1.4|
| Our approach              | 1.7        | 40.6   | 3.4  | 25.8       | 62.0   | **36.5** |

5 We perform a segment-level evaluation and do not quantify word-level accuracy in this section since the dataset does not contain consistently annotated spans for coverage errors.
### Evaluation Design

We employed two linguistic experts per language pair as raters. Each rater was shown around 700 randomly sampled positive predictions across both types of coverage errors.

Raters were shown the source sequence, the machine translation, and the predicted error span. They were asked whether the highlighted span was indeed translated badly, and were asked to perform a fine-grained analysis based on a list of predefined answer options (Figures 3 and 4 in the Appendix).

A part of the samples were annotated by both raters. The agreement was moderate for the main question, with a Cohen’s kappa of 0.54 for English–German and 0.45 for Chinese–English. Agreement on the more subjective follow-up question was lower (0.32 / 0.13).

### Results

The fine-grained answers allow us to quantify the word-level precision of the spans highlighted by our approach, both with respect to coverage errors in particular and to translation errors in general (Table 2). Precision is higher than expected when detecting omission errors in English–German translations, but is still low for additions. The distribution of the detailed answers (Figures 3 and 4 in the Appendix) suggests that syntactical differences between the source and target language contribute to the false positives regarding additions. Example predictions are provided in Appendix F, which include cases where all three raters of Freitag et al. (2021) had overlooked the coverage error.

Finally, Table 2 shows that many of the predicted error spans are in fact translation errors, but not coverage errors in a narrow sense. For example, more than 10% of the spans marked in Chinese–English translations were classified by our raters as a different type of accuracy error, such as mistranslation.

### Limitations and Future Work

We hope that the automatic detection of coverage errors could be an aid to translators and post-editors, given that manually detecting such errors is tedious. Our results on omissions are encouraging, and user studies are recommended in order to validate the usefulness of the predictions to practitioners. Further work needs to be done to improve the detection of additions, of which the real-world data contain few examples. Higher accuracy would be necessary for word-level QE to be helpful (Shenoy et al., 2021), and so with regard to detecting addition errors, the practical utility of both the baseline and of our approach remains limited.

Inference time should also be discussed. In Appendix C we perform a comparison, finding that on a long sentence pair contrastive conditioning can take up to ten times longer than a forward pass of the baseline. However, this is still a fraction of the time needed for generating a translation in the first place. In addition, restricting the potential error spans that are considered could further improve efficiency.

### Conclusion

We have proposed a reference-free method to automatically detect coverage errors in translations. Derived from contrastive conditioning, our method relies on hypothetical reasoning over the likelihood of partial sequences. Since any off-the-shelf NMT model can be used to estimate conditional likelihood, no access to the original translation system or to a quality estimation model is needed. Evaluation on real machine translations shows that our approach outperforms a supervised baseline in the detection of omissions. Future work could address the low precision on addition errors, which are relatively rare in the datasets we used for evaluation.

### Acknowledgments

This work was funded by the Swiss National Science Foundation (project MUTAMUR; no. 176727). We would like to thank Xin Sennrich for facilitating the recruitment of annotators, and Chantal Amrhein as well as the anonymous reviewers for helpful feedback.

### References

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette...
Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanos, Tom Kočmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In Proceedings of the Fifth Conference on Machine Translation, pages 1–55, Online. Association for Computational Linguistics.

Sheila Castilho, Joss Moorkens, Federico Gaspari, Rico Sennrich, Vilemíni Sosoni, Yota Georgakopoulou, Pintu Lohar, Andy Way, Antonino Miceli Barone, and Maria Gialama. 2017. A comparative quality evaluation of PBSMT and NMT using professional translators. 16th Machine Translation Summit 2017, pages 116–131.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8440–8451, Online. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Christopher D. Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal Dependencies. Computational Linguistics, 47(2):255–308.

Erick Fonseca, Lisa Yankovskaya, André F. T. Martins, Mark Fishel, and Christian Federmann. 2019. Findings of the WMT 2019 shared tasks on quality estimation. In Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2), pages 1–10, Florence, Italy. 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.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Hyun Kim, Jong-Hyek Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In Proceedings of the Second Conference on Machine Translation, pages 562–568, Copenhagen, Denmark. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Xiang Kong, Zhaopeng Tu, Shuming Shi, Eduard Hovy, and Tong Zhang. 2019. Neural machine translation with adequacy-oriented learning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 6618–6625.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7871–7880, Online. Association for Computational Linguistics.

Yanyang Li, Tong Xiao, Yinqiao Li, Qiang Wang, Changming Xu, and Jingbo Zhu. 2018. A simple and effective approach to coverage-aware neural machine translation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 292–297, Melbourne, Australia. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual Denoising Pre-training for Neural Machine Translation. Transactions of the Association for Computational Linguistics, 8:726–742.

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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In International Conference on Learning Representations.

João Moura, Miguel Vera, Daan van Stigt, Fabio Kepler, and André F. T. Martins. 2020. IST-unbabel participation in the WMT20 quality estimation shared task. In Proceedings of the Fifth Conference on Machine Translation, pages 1029–1036, Online. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.
Lucia Specia, Frédéric Blain, Marina Fomicheva, Er- 
Raksha Shenoy, Nico Herbig, Antonio Krüger, and 
Josef van Genabith. 2021. Investigating the help- 
fulness of word-level quality estimation for post-
editing machine translation output. In Proceedings 
of the 2021 Conference on Empirical Methods in 
Natural Language Processing, pages 10173–10185, 
Online and Punta Cana, Dominican Republic. Asso-
ciation for Computational Linguistics.

Lucia Specia, Frédéric Blain, Varvara Logacheva, 
Ramón F. Astudillo, and André F. T. Martins. 2020. Findings of the 
WMT 2020 shared task on quality estimation. In Proceedings of the Fifth Conference on Machine 
Translation, pages 743–764, Online. Association for Compu-
tational Linguistics.

Lucia Specia, Frédéric Blain, Marina Fomicheva, Er-
rick Fonseca, Vishrav Chaudhary, Francisco Guzmán, 
and André F. T. Martins. 2020. Findings of the 
WMT 2018 shared task on quality estimation. In Proceedings of the Third Conference on Machine Translation: Shared Task Papers, pages 689–709, Belgium, Brussels. Association for Compu-
tational Linguistics.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Na-
man Goyal, Vishrav Chaudhary, Jiatao Gu, and An-
abela Fan. 2021. Multilingual translation from de-
oising pre-training. In Proceedings of the Association for Compu-
tational Linguistics: ACL-IJCNLP 2021, pages 3450–3466, Online. Association for Compu-
tional Linguistics.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, 
and Hang Li. 2016. Modeling coverage for neural 
machine translation. In Proceedings of the 54th An-
ual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 76–85, Berlin, Germany. Association for Computational Linguistics.

Yi-Lin Tuan, Ahmed El-Kishky, Adithya Renduchin-
tala, Vishrav Chaudhary, Francisco Guzmán, and Lu-
cia Specia. 2021. Quality estimation without human-
labeled data. In Proceedings of the 16th Conference of the European Chapter of the Association for Compu-
tational Linguistics: Main Volume, pages 619–625, Online. Association for Computational Linguistics.

Jannis Vamvas and Rico Sennrich. 2021. Contrastive 
conditioning for assessing disambiguation in MT: A 
case study of distilled bias. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 10246–10265, Online 
and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V 
Le, Mohammad Norouzi, Wolfgang Macherey, 
Maxim Krikun, Yuan Cao, Qin Gao, Klaus 
Macherey, et al. 2016. Google’s neural machine 
translation system: Bridging the gap between hu-
man and machine translation. arXiv preprint arXiv:1609.08144.

Jing Yang, Biao Zhang, Yue Qin, Xiangwen Zhang, 
Qian Lin, and Jinsong Su. 2018. Otem&Utem: Over-
and under-translation evaluation metric for NMT. In Natural Language Processing and Chi-
inese Computing, pages 291–302, Cham. Springer In-
ternational Publishing.

Zonghan Yang, Yong Cheng, Yang Liu, and Maosong 
Sun. 2019. Reducing word omission errors in neu-
rnal machine translation: A contrastive learning ap-
proach. In Proceedings of the 57th Annual Meet-
ing of the Association for Computational Linguistics, pages 6919–6919, Florence, Italy. Association for Computational Linguistics.

Zaixiang Zheng, Hao Zhou, Shujian Huang, Lili Mou, 
Xinyu Dai, Jiajun Chen, and Zhaopeng Tu. 2018. Modeling Past and Future for Neural Machine Trans-
lation. Transactions of the Association for Compu-
tational Linguistics, 6:145–157.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona 
Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting halluci-
nated content in conditional neural sequence gener-
ation. In Findings of the Association for Compu-
tational Linguistics: ACL-IJCNLP 2021, pages 1393–1404, Online. Association for Computational Linguistics.

A Annotator Guidelines

You will be shown a series of source sentences and translations. One or several spans in the text are highlighted and it is claimed that the spans are translated badly. You are asked to determine whether the claim is true. The highlighted spans can be either in the source sequence or in the trans-
lation. If a span is in the source sentence, check whether it has been correctly translated. If a span is in the translation, check whether it correctly con-
veys the source. Sometimes, multiple spans are highlighted. In that case, focus your answer on the span that is most problematic for the trans-
lation. In a second step, you are asked to select an explanation. On the one hand, if you agree that the highlighted span is translated badly, please ex-
plain your reasoning by selecting your explanation. On the other hand, if you disagree and think that the span is well-translated, please select an expla-
nation why the span might have been marked as badly translated in the first place. Should multiple explanations be equally plausible, select the first from the top.
Table A1: Segment-level and word-level (MCC) evaluation based on a test set with synthetic coverage errors.

|                  | Detection of additions | Detection of omissions |
|------------------|------------------------|------------------------|
|                  | Prec. | Recall | F1   | MCC  | Prec. | Recall | F1   | MCC  |
| EN–DE Supervised |       |        |      |      |       |        |      |      |
| Baseline         | 98.8±0.4 | 98.0±2.0 | 98.4±2.0 | 96.8±1.1 | 94.0±1.3 | 96.6±0.4 | 95.3±0.5 | 90.5±2.0 |
| Ours             | 78.1  | 88.3   | 82.9 | 76.7 | 80.9  | 98.6   | 88.9 | 78.1 |
| ZH–EN Supervised |       |        |      |      |       |        |      |      |
| Baseline         | 87.2±1.5 | 75.7±6.0 | 81.0±3.0 | 72.6±6.0 | 67.3±1.3 | 68.0±1.2 | 67.7±9.0 | 53.8±3.0 |
| Ours             | 26.1  | 88.9   | 40.4 | 23.3 | 28.3  | 92.0   | 43.3 | 40.3 |

Table A2: Inference times when predicting on a short and a long sentence pair. Since we did not use a parser that is optimized for efficiency, we additionally report inference time without including the time needed for parsing.

|                  | Short sentence pair | Long sentence pair |
|------------------|---------------------|---------------------|
|                  | Additions | Omissions | Both | Additions | Omissions | Both |
| Supervised baseline | -         | -         | 25 ms | -         | -         | 25 ms |
| Our approach     | 40 ms    | 45 ms    | 83 ms | 165 ms   | 197 ms   | 365 ms |
| – excluding parser | 18 ms    | 21 ms    | 38 ms | 102 ms   | 144 ms   | 239 ms |

B Evaluation on Synthetic Errors

We used a test split held back from the synthetic data to perform an additional evaluation. On the segment level, we report Precision, Recall and F1-score. Like in Section 5.1, a prediction is treated as correct on the segment level if for a predicted coverage error there is indeed a coverage error of that type anywhere in the segment.

On the word level, we follow previous work on word-level QE (Specia et al., 2020) and report the Matthews correlation coefficient (MCC) across all the tokens in the test set.

Results Results are shown in Table A1. The supervised baseline has a high accuracy on English–German translations and a moderate accuracy on Chinese–English translations. In comparison, our approach performs clearly worse than the supervised baseline on the synthetic errors.

C Inference Time

Inference times are reported in Table A2. We measure the time needed to run the coverage error detection methods on a short sentence pair and on a long sentence pair for English–German. The short sentence pair is taken from Figure 1 and the long sentence pair has 40 tokens in the source sequence and 47 tokens in the target sequence. We average over 1000 repetitions on RTX 2080 Ti GPUs.

The higher inference times for our approach can be explained by the number of translation probabilities that need to be estimated. On average, we compute 30 scores per sentence in the English–German MQM dataset, and 44 per sentence in the Chinese–English MQM dataset. Still, the time needed for computing all these scores is only a fraction of the time it takes to generate a translation (254 ms for the short source sentence and 861 ms for the long sentence, assuming a beam size of 5).

The required number of scores could be reduced by considering fewer potential error spans. Furthermore, scoring could be parallelized across batches of multiple translations. Finally, using a more efficient parser, or no parser at all, could speed up inference.
### D Dataset Statistics

| Dataset split | Number of segments | Number of tokens |   |   |   |   |
|---------------|--------------------|-----------------|---|---|---|---|
|               | Total              | W/ addition     | W/ omission | Src. OK | Src. BAD | Tgt. OK | Tgt. BAD |
| EN–DE Train   | 135269             | 18423           | 18423        | 2185918 | 58378 | 2197843 | 53911 |
| EN–DE Dev     | 16984              | 2328            | 2328         | 273311  | 7398  | 275156  | 6781  |
| EN–DE Test    | 16984              | 2328            | 2328         | 273277  | 7701  | 275036  | 7032  |
| ZH–EN Train   | 110195             | 10697           | 10697        | 2576135 | 62311 | 1866567 | 37730 |
| ZH–EN Dev     | 14149              | 1383            | 1383         | 326743  | 7562  | 236685  | 4244  |
| ZH–EN Test    | 14026              | 1342            | 1342         | 322000  | 7566  | 234757  | 4882  |

Table A3: Statistics for the dataset of synthetic coverage errors described in Section 4.

| Dataset split | Number of segments | Number of segments |   |   |
|---------------|--------------------|--------------------|---|---|
|               | Total              | W/ addition error  | W/ omission error |   |   |
| EN–DE Dev     | 1418               | 77                 | 187           |   |   |
| EN–DE Test    | 8508               | 407                | 1057          |   |   |
| – without excluded segments | 4839 | 162 | 484 |
| ZH–EN Dev     | 1999               | 69                 | 516           |   |   |
| ZH–EN Test    | 13995              | 329                | 3360          |   |   |
| – without excluded segments | 8851 | 149 | 1569 |

Table A4: Statistics for the gold dataset by Freitag et al. (2021).

### E Examples of Synthetic Coverage Errors

#### English–German Example

**Addition error**

*Partial source:* But they haven’t played.

*Full machine translation:* Aber sie haben nicht gegen ein Team wie uns gespielt.

**Omission error**

*Full source:* But they haven’t played against a team like us.

*Partial machine translation:* Aber sie haben nicht gespielt.

#### Chinese–English Example

**Addition error**

*Partial source:* 医院和企业共研发相关检测试剂盒，惠及更多患者。

*Full translation:* Hospitals and enterprises jointly develop related test kits to benefit more cancer patients.

**Omission error**

*Full source:* 医院和企业共研发相关检测试剂盒，惠及更多肿瘤患者。

*Partial translation:* Hospitals and enterprises jointly develop related test kits to benefit more patients.
F Examples of Coverage Errors Predicted by Contrastive Conditioning

English–German Examples

**Predicted addition error**

Source: He added: "It’s backfired on him now, though, that’s the sad thing."

Machine translation: Er fügte hinzu: "Es ist jetzt auf ihn abgefeuert, aber das ist das Traurige."

Original MQM rating (Freitag et al., 2021): No related accuracy error marked by the three raters.

Answer by our human rater: The highlighted target span is not translated badly. It might have been highlighted because it is syntactically different from the source.

Meaning of highlighted span: hinzu = ‘additionally’

**Predicted omission error**

Source: The automaker is expected to report its quarterly vehicle deliveries in the next few days.

Machine translation: Der Autohersteller wird voraussichtlich in den nächsten Tagen seine vierteljährlichen Fahrzeugauslieferungen melden.

Original MQM rating: No related accuracy error marked by the three raters.

Answer by our human rater: The highlighted source span is not translated badly. The words in the span do not need to be translated.

**Chinese–English Examples**

**Predicted addition error**

Source: 美方指责伊朗制造了该袭击，并对伊朗实施制裁。

Machine translation: The US accused Iran of causing the attack and imposed new sanctions on Iran.

Original MQM rating (Freitag et al., 2021): No related accuracy error marked by the three raters.

Answer by our human rater: The highlighted target span is not translated badly. No phenomenon that might have caused the prediction was identified.

**Predicted omission error**

Source: 目前已收到来自俄罗斯农业企业的约50项申请。

Machine translation: About 50 applications have been received from Russian agricultural enterprises.

Original MQM rating: No accuracy error marked by the three raters.

Answer by our human rater: The highlighted source span is indeed translated badly. It contains information that is missing in the translation.

Meaning of highlighted span: 目前 = ‘at present’

**Predicted omission error**

Source: 他说，该系统目前在世界上有很大需求，但俄罗斯军队也需要它，其中包括在北极地区。

Machine translation: He said that the system is currently in great demand in the world, but the Russian army also needs it, including in the Arctic.

Original MQM rating: No accuracy error marked by the three raters.

Answer by our human rater: The highlighted source span is not translated badly. The words in the span do not need to be translated.

Meaning of highlighted span: 其中 = ‘among’
G Detailed Results of Human Evaluation

Correctly predicted additions
- The span adds unsupported information.
- The span adds information that is supported by the context or trivial.
- The span is badly translated because of an accuracy error.
- The span is badly translated because of a fluency error.

Falsely predicted additions
- The words in the span are redundant but fluent.
- The words in the span are supported by the context or trivial.
- The translation is syntactically different from the source.
- No phenomenon identified

Correctly predicted omissions
- The span contains information that is missing in the translation.
- The span contains information that is missing but can be inferred or is trivial.
- The span is badly translated because of an accuracy error.
- The span is badly translated because of a fluency error.

Falsely predicted omissions
- The words in the span do not need to be translated.
- The words in the span are supported by the context or trivial.
- The translation is syntactically different from the source.
- No phenomenon identified

Figure 3: Results for the human evaluation of predicted addition errors. If human raters answered that the highlighted span in the translation was indeed badly translated, they were offered the four explanation options on the left. Otherwise they chose from the four options on the right.

Figure 4: Results for the human evaluation of predicted omission errors. If human raters answered that the highlighted span in the source sequence was indeed badly translated, they were offered the four explanation options on the left. Otherwise they chose from the four options on the right.