Automatic Recognition and Classification of Errors in Human Translation

Luise Dürlich, Christian Hardmeier
Department of Linguistics and Philology
Uppsala University
luise.durlich.9577@student.uu.se christian.hardmeier@lingfil.uu.se

Abstract

Grading student assignments is a time-consuming part of teaching translation. Automatic tools that assist in this task would give teachers of professional translation more time to focus on other aspects of their job. With this project, we make a first attempt at both error recognition and classification for human-translated text employing both mono- and bilingual models: BERT – a pre-trained monolingual language model – and NuQE – a model adapted from the field of Quality Estimation for Machine Translation – are trained on a relatively small hand-annotated corpus of student translations. Due to the nature of the task, errors are quite rare compared to correctly translated tokens in the corpus. To account for this, we train the models with both over- and undersampled data. While both models detect errors with moderate success, NuQE adapts very poorly to the fine-grained classification setting. Despite low overall scores due to class imbalance and a lack of training data, we show that powerful monolingual language models in combination with over- and undersampling techniques can detect formal, lexical and translational errors with some success.

1 Introduction

Despite the advances of Machine Translation (MT) in the last few years, human translation remains the gold standard for professional language mediation. Skilled translators produce easily understandable and adequate translations. In order to achieve that, they go through years of language study and training, which involve copious amounts of writing, correcting and discussing translated text with their teachers. An important part in this is finding and explaining mistakes (Meyer, 1989), to raise awareness of the differences between target and source languages and common difficulties that arise during translation.

There are automatic and semi-automatic tools like translation memories to assist professional translators with the translation of terms and phrases. However, there is little automatic support for giving feedback on human translation quality. In fact, rather than being the subject of automatic evaluation, human translations are generally considered as reference for the assessment of MT systems (Miller and Beebe-Center, 1956; Papineni et al., 2002; Banerjee and Lavie, 2005).

An automatic system for the assessment and correction of student-translated text would be an important help for translation teachers. A first step in this would be recognising errors and determining their types. This project is a first attempt at such a tool using state-of-the-art machine learning techniques in the form of an established architecture for quality estimation (QE) for MT and a large pre-trained monolingual language model.

2 Related Work

Error tagging and error recognition have previously been explored in related contexts, such as learner texts and MT (Stymne, 2011; Popović, 2011; Zeman et al., 2011; Irvine et al., 2013; Lei et al., 2019). In learner text, errors and error types are mainly identified with rule-based systems (Bryant et al., 2017; Kutuzov and Kuzmenko, 2015; Boyd, 2018; Kempfert and Köhn, 2018), whereas error tagging in MT also featured machine learning in the form of decision trees (Martins and Caseli, 2014), random forests as meta classifiers over maximum entropy classification (Mehay et al., 2014).

A closely related field is that of QE, where MT system quality is assessed without the comparison to any human reference translation. On word level, QE annotates tokens as either "OK" or "BAD", i.e. acceptable translations or errors. In recent shared tasks on QE, the use of large pre-trained language

models such as BERT in combination with model ensembling has led to improvements in the word-level task (Fonseca et al., 2019).

3 Error Recognition and Classification

We approach error recognition like the word-level QE task as binary classification into either errors or correct translations and define error classification as a multi-class classification problem distinguishing between correct translations and multiple different types of errors.

The different error types are based on the annotations in the corpus used for this project: the KOPTE corpus (Wurm, 2016) is a collection of annotated student translations that were collected during French-to-German translation classes at Saarland University. In these translations, errors are either highlighted with no distinction of error type or assigned to one of eight categories: form, grammar, function, cohesion, lexis and semantics, stylistics and register, structure and translational problems[1].

4 Data

The data – 1,181 translated text with word tokens either falling into the correctly translated class or a single error class[2] for error recognition and 1,057 translations with multiple error classes for classification – is split into approximately 70% training, 20% test and 10% development data. Because the translations come from classes where many students translated the same text, we ensured that there was no overlap in source texts between the different data sets. Table 1 gives more details on the number of translations and source texts in each set as well as the distribution of the binary tags in the upper part and the different coarse error categories in the lower part. "Train (recognition*)" here refers to those translations where errors were only highlighted. As their annotation did not distinguish error type, they were not suitable for error classification and could only be used for the error recognition task.

Translation-to-source statistics

Set	Translations	Source Texts	Tags OK		Tags other		Tokens
Train (classification)	723	65	219,969	(95.39%)	10,622	(4.61%)	230,591
Train (recognition*)	124	21	47,532	(89.70%)	5,460	(10.30%)	52,992
Dev	115	7	35,242	(93.58%)	2,418	(6.42%)	37,660
Test	219	16	64,713	(94.55%)	3,729	(5.45%)	68,442

Error categories

Set	form	grammar	function	cohesion	lexis	stylistics	structure	transl.
Train	1,593	1,634	0	622	5,297	395	123	958
Dev	372	146	0	110	1,553	80	27	130
Test	639	498	0	205	1,853	253	16	265

Table 1: Statistics on the Data splits

4.1 Data Sampling

A problem with both versions of the training corpus is the overall rarity of errors: in the training corpus that distinguishes error types, only about 5% of all tokens are errors. This imbalance causes naive models to predict only the majority class. To prevent this, we try both over- and undersampling. For undersampling, we discard any translated sentence that does not contain an error. To oversample the data, we sample over the error categories. For each category we sample full sentences containing that type of error until we reach a certain threshold of error instances[3]. This threshold is set to 1,400, which is approximately the number of occurrences of the second and third most frequent error class.

Figure 1 shows the class proportion for the different error classes on the original error classification training set as well as its over- and undersampled versions.

As we can see, both over- and undersampling help increase the proportion of the different error classes. Still, as most sentences contain at least some correctly translated words, the correct class still accounts for the majority of instances in the over- and undersampled sets.

5 Baselines

As baseline models we approach the problem from two perspectives: First, we consider the translations with a purely monolingual model. A pre-trained BERT (Devlin et al., 2019) model is fitted for token classification on the German translations. We use *bert-base-german-cased*[4], a monolingual German BERT model trained on 12 GB of data, namely

[1] These are also the categories used in error classification in the following. Because of page limitations, we will not go into further detail on what they encode, but refer to the typology in the appendix of Wurm (2016).

[2] This includes all the translations with fine-grained annotation. For the recognition scenario, the different error tags were simply all mapped to the "BAD" tag.

[3] Note that the number of instances for an already sampled class can still increase at a later time, because the sentences may contain more than one different type of error. Since most sentences containing an error still contain instances of correctly translated words, we do not sample over that class and in doing so, just like in undersampling exclude sentences without errors.

[4] https://deepset.ai/german-bert

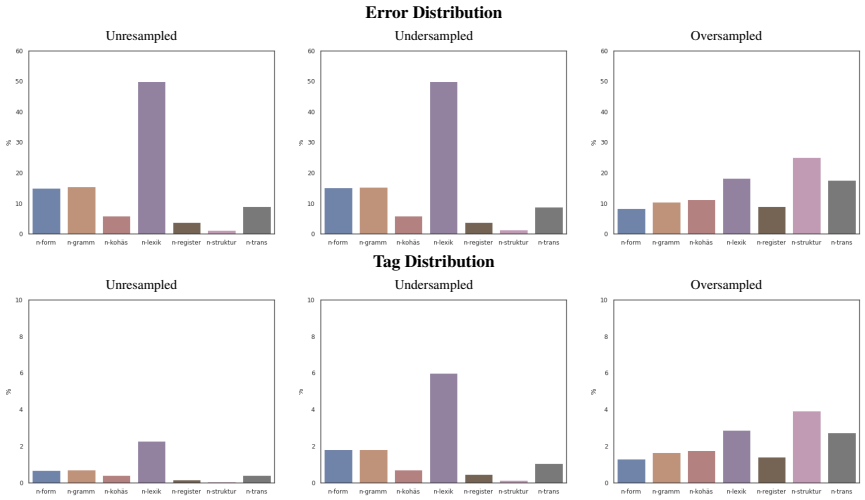

Figure 1: Statistics on the sampled training data – for the overall tag distribution the percentage of correctly translated tokens ("OK") is 95.30, 88.01 and 84.30 respectively.

Wikipedia text, news articles and a corpus of court decisions, and the Transformers library for Python (Wolf et al., 2019) to fit the token classification model. We fine-tune three versions of this model, each for 3 epochs with a learning rate of $3 \cdot 10^{-5}$: the first version (model (1) in the result section) is trained with default model parameters, for the second (model (2)) the weights of the cross entropy loss is adapted such that 99% of the weight is given to "BAD" for error recognition or divided up between the 7 error classes for error classification. Finally, the third version (model (3)) is also trained with default parameters, but the underlying BERT model undergoes another 40 epochs of continued pre-training with German news text from the WMT 2019 French-German news translation task data. We adapt the model more to the news domain, because a majority of the text in the KOPTE corpus is news text.

The second baseline (model (4) in the result section) is bilingual, the Neural Quality Estimation (NuQE) model (Martins et al., 2017). It consists of eight hidden layers – three sets of two feed-forward layers separated by bidirectional gated recurrent units (BiGRU) (Cho et al., 2014). In the version adopted here, the model accepts a window of three consecutive tokens of translated text and three tokens of source text, where the centre tokens are aligned. Each token is encoded in a 50 dimensional embedding space and the vectors of all six tokens are concatenated. The output is a sigmoid layer with a threshold of 0.5 to produce a "BAD" tag.

Following Martins et al. (2017), we train the model for 50 epochs with 0.5 dropout on the embedding layers and a loss weight factor of 3.0 for error classes.

6 Two-level Prediction

To counteract the overgeneration of error class tags, we train a sentence-level BERT model, again using the *bert-base-german-cased* model. This model is used to preselect sentences containing an error. For sentences that this model labels as correct, each token is assigned the "OK" label. All other sentences are passed to a baseline model to generate token labels. The resulting model is denoted "S + (N)", where S stands for the sentence-level model and (N) is the token-level model that is used.

7 Evaluation

We adopt several F_1-score-based metrics from previous WMT QE shared tasks to evaluate. For error recognition, we report F_1-Mult, the product of the F_1 scores of correctly and incorrectly translated tokens (Bojar et al., 2016): For error classification we look at two weighted measures from Bojar et al. (2014), the weighted averaged F_1 for all classes and the weighted averaged F_1 for errors only:

$$wF_{1,ALL} = \frac{1}{\sum_c N(c)} \sum_c N_c \times F_{1,c}$$

$$wF_{1,ERR} = \frac{1}{\sum_{c:c \neq OK} N(c)} \sum_{c:c \neq OK} N_c \times F_{1,c}$$

We also consider F_1 per class for both tasks. As the goal is successful error recognition and classification and we want to avoid pure majority class classification, we focus more on the results over errors than on the results on the correct class.

8 Results

Table 2 shows the results for error recognition, both using the designated recognition models in the upper part and when mapping the error classes predicted by the classification model to the "BAD" class. We can see that the recognition models trained on undersampled data – with the exception of model (3) – do better at identifying errors and in most cases also produce higher F_1-Mult scores. The best model is the combined model using token-level model (2) with an F_1-Mult of 16.00%. Considering the results of model (2) by itself, this suggests that the model, likely because of the adapted loss weights, overgenerates "BAD" tags and that this can be mitigated through the two-level approach, as suggested by the improvements on both classes.

Recognition Models

Undersampled				Oversampled			
Model	$F_{1,OK}$	$F_{1,BAD}$	F_1-Mult	Model	$F_{1,OK}$	$F_{1,BAD}$	F_1-Mult
(1)	95.30	14.05	13.39	(1)	94.96	13.85	13.15
(2)	80.16	16.96	13.60	(2)	89.53	16.52	14.79
(3)	95.24	13.04	12.42	(3)	94.20	14.50	13.66
(4)	73.74	14.46	10.66	(4)	77.08	12.69	9.78
S + (1)	**96.06**	11.59	11.13	S + (1)	**95.94**	11.49	11.02
S + (2)	88.53	**18.08**	**16.00**	S + (2)	93.38	**16.91**	**15.79**
S + (3)	**96.06**	10.85	10.85	S + (3)	95.59	13.06	12.48
S + (4)	85.82	16.34	14.23	S + (4)	86.89	13.69	11.90

Classification Models

Undersampled				Oversampled			
Model	$F_{1,OK}$	$F_{1,BAD}$	F_1-Mult	Model	$F_{1,OK}$	$F_{1,BAD}$	F_1-Mult
(1)	95.79	13.27	12.71	(1)	95.52	**13.39**	**12.80**
(2)	92.76	**18.17**	**16.86**	(2)	95.19	13.13	12.50
(3)	95.60	9.81	9.38	(3)	95.35	10.66	10.16
(4)	**96.72**	0.00	0.00	(4)	94.97	5.40	5.13
S + (1)	96.35	10.66	10.27	S + (1)	**96.19**	10.71	10.30
S + (2)	94.85	17.16	16.28	S + (2)	96.01	10.59	10.17
S + (3)	96.24	6.98	6.72	S + (3)	96.07	8.91	8.56
S + (4)	**96.72**	0.00	0.00	S + (4)	95.86	4.48	4.29

Table 2: Error recognition results on the test set

Considering the classification models for recognition, training on oversampled data leads to better results with models (1), (3) and (4) than learning from the undersampled test set. This time, model (2) by itself produces the best results and even improves over the pure recognition model at 16.86% F_1-Mult.

Table 3 displays the averaged results for error classification. For model (3) and model (4) the oversampled training data leads to better results on errors. Model (4) with undersampled training data still only assigns the majority class, despite the sampling efforts. Looking at Table 4, which lists per-class F_1 scores, model (4) appears to at least recognise some form, grammar and lexical errors when trained on oversampled data. Again, the best model with respect to errors is undersampled model (2) at a weighted F_1 of 10.67%. This model manages to correctly identify cases of seven of the eight classes. One reason for the overall low scores on error classes could be that an error in the translation can constitute more than one type of error at once, which may lead to confusion over the type to assign.[5] Seeing as certain error types like structure, stylistics and cohesion remain hard to identify, the monolingual approach certainly has its limitations for instances for overall fluent translations that are flawed in the context of the source text.

Undersampled			Oversampled		
Model	$wF_{1,ALL}$	$wF_{1,ERR}$	Model	$wF_{1,ALL}$	$wF_{1,ERR}$
(1)	90.17	7.26	(1)	89.83	5.89
(2)	87.54	**10.67**	(2)	89.54	**6.28**
(3)	89.82	4.66	(3)	89.63	5.37
(4)	90.58	0.00	(4)	89.03	1.45
S + (1)	**90.59**	5.69	S + (1)	**90.39**	4.97
S + (2)	89.46	10.02	S + (2)	90.22	4.93
S + (3)	90.33	3.25	S + (3)	90.26	4.60
S + (4)	90.58	0.00	S + (4)	89.86	1.48

Table 3: Averaged classification results on the test set

Undersampled

Model	OK	form	grammar	cohesion	lexis	stylistics	structure	transl.
(1)	95.79	11.07	1.84	0.00	9.94	0.00	0.00	1.23
(2)	92.76	14.24	5.74	3.27	13.02	2.06	0.00	7.74
(3)	95.60	6.61	0.00	0.00	6.34	0.00	0.00	3.82
(4)	96.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00
S + (1)	96.35	8.58	1.56	0.00	7.73	0.00	0.00	1.34
S + (2)	94.85	12.55	5.31	3.02	12.36	1.06	0.00	9.15
S + (3)	96.24	4.55	0.00	0.00	4.36	0.00	0.00	3.13
S + (4)	96.72	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Oversampled

Model	OK	form	grammar	cohesion	lexis	stylistics	structure	transl.
(1)	95.52	7.70	3.67	0.00	6.84	1.51	0.00	6.75
(2)	95.19	8.08	3.72	0.48	7.32	1.64	0.00	7.22
(3)	95.35	6.55	1.16	0.61	6.46	0.49	0.00	9.37
(4)	94.97	0.32	0.57	0.00	2.65	0.00	0.00	0.00
S + (1)	96.19	6.41	3.61	0.00	5.61	0.00	0.00	7.10
S + (2)	96.01	5.00	3.37	0.00	5.68	0.00	0.00	9.55
S + (3)	96.07	4.64	1.37	0.70	5.59	0.00	0.00	9.41
S + (4)	95.86	0.34	0.78	0.00	2.65	0.00	0.00	0.00

Table 4: Error classification results: F_1 per class on the test set

9 Conclusion

In this paper, we investigated the use of mono- and bilingual models for error recognition and classification on a small set of human translations. Different sampling strategies were explored to address data imbalance and shown to work with different types of models. The results suggest that large pretrained monolingual models like BERT can outperform established bilingual QE approaches trained from scratch. Still, for more accurate identification of errors and classification of error types, more annotated data and adapted sampling methods are needed. Further, strictly monolingual perspectives from the target language are limited in the amount of information they can capture e.g. regarding aspects of semantics and form.

[5] We did not approach the problem from a multi-label perspective as there are too few instances of errors and even fewer instances of multi-labelled tokens in the corpus.

Acknowledgements

Christian Hardmeier was supported by the Swedish Research Council under grant 2017-930.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the Association for Computational Linguistics 2005 Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 Workshop on Statistical Machine Translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Adriane Boyd. 2018. Using Wikipedia Edits in Low Resource Grammatical Error Correction. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84, Brussels, Belgium. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmirty Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of Empirical Methods in Natural Language Processing*, pages 1724–1734.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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 1–12, Florence, Italy. Association for Computational Linguistics.

Ann Irvine, John Morgan, Marine Carpuat, Hal Daumé III, and Dragos Munteanu. 2013. Measuring Machine Translation Errors in New Domains. *Transactions of the Association for Computational Linguistics*, 1:429–440.

Inga Kempfert and Christine Köhn. 2018. An Automatic Error Tagger for German. In *Proceedings of the 7th workshop on NLP for Computer Assisted Language Learning*, pages 32–40, Stockholm, Sweden.

Andrey Kutuzov and Elizaveta Kuzmenko. 2015. Semi-automated typical error annotation for learner English essays: integrating frameworks. In *Proceedings of the fourth workshop on NLP for computer-assisted language learning*, pages 35–41, Vilnius, Lithuania. LiU Electronic Press.

Wenqiang Lei, Weiwen Xu, Ai Ti Aw, Yuanxin Xiang, and Tat Seng Chua. 2019. Revisit Automatic Error Detection for Wrong and Missing Translation – A Supervised Approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 942–952, Hong Kong, China. Association for Computational Linguistics.

André F. T. Martins, Marcin Junczys-Dowmunt, Fabio N. Kepler, Ramón Astudillo, Chris Hokamp, and Roman Grundkiewicz. 2017. Pushing the Limits of Translation Quality Estimation. *Transactions of the Association for Computational Linguistics*, 5:205–218.

Débora Beatriz de Jesus Martins and Helena de Medeiros Caseli. 2014. Automatic Machine Translation Error Identification. *Machine Translation*, 29(1):1–24.

Dennis Mehay, Sankaranarayanan Ananthakrishnan, and Sanjika Hewavitharana. 2014. Lightly-Supervised Word Sense Translation Error Detection for an Interactive Conversational Spoken Language Translation System. In *Proceedings of the 14th Conference of the European Chapter of the Association*

for Computational Linguistics, volume 2: Short Papers, pages 54–58, Gothenburg, Sweden. Association for Computational Linguistics.

Ingrid Meyer. 1989. A Translation-Specific Writing Program: Justification and Description. In Peter W. Krawutschke, editor, *Translator and Interpreter Training and Foreign Language Pedagogy*, volume 3 of *American Translators Association scholarly monograph series*, pages 119–131. John Benjamins Publishing Company, Amsterdam/Philadelphia.

George A. Miller and J. G. Beebe-Center. 1956. Some Psychological Methods for Evaluating the Quality of Translation. *Mechanical Translation*, 3(3):73–80.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2011. Hjerson: An Open Source Tool for Automatic Error Classification of Machine Translation Output. *The Prague Bulletin of Mathematical Linguistics*, 96:59–68.

Sara Stymne. 2011. Blast: A Tool for Error Analysis of Machine Translation Output. In *Proceedings of the Association for Computational Linguistics Human Language Technologies 2011 System Demonstrations*, pages 56–61, Portland, Oregon, USA. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing.

Andrea Wurm. 2016. Presentation of the KOPTE Corpus – Version 2.

Daniel Zeman, Mark Fishel, Jan Berka, and Ondřej Bojar. 2011. Addicter: What is wrong with my translations? *The Prague Bulletin of Mathematical Linguistics*, 96:79–88.