

TransQuest at WMT2020: Sentence-Level Direct Assessment

Tharindu Ranasinghe[◇], Constantin Orăsan[♡] and Ruslan Mitkov[◇]

[◇]Research Group in Computational Linguistics, University of Wolverhampton, UK

[♡]Centre for Translation Studies, University of Surrey, UK

{t.d.ranasinghehettiarachchige, r.mitkov}@wlv.ac.uk

c.orasan@surrey.ac.uk

Abstract

This paper presents the team TransQuest’s participation in Sentence-Level Direct Assessment shared task in WMT 2020. We introduce a simple QE framework based on cross-lingual transformers, and we use it to implement and evaluate two different neural architectures. The proposed methods achieve state-of-the-art results surpassing the results obtained by OpenKiwi, the baseline used in the shared task. We further fine tune the QE framework by performing ensemble and data augmentation. Our approach is the winning solution in all of the language pairs according to the WMT 2020 official results.

1 Introduction

The goal of quality estimation (QE) systems is to determine the quality of a translation without having access to a reference translation. This makes it very useful in translation workflows where it can be used to determine whether an automatically translated sentence is good enough to be used for a given purpose, or if it needs to be shown to a human translator for translation from scratch or postediting (Kepler et al., 2019). Quality estimation can be done at different levels: document level, sentence level and word level (Ive et al., 2018). This paper presents TransQuest, a sentence-level quality estimation framework which is the winning solution in all the language pairs in the WMT 2020 Sentence-Level Direct Assessment shared task (Specia et al., 2020).

In the past, high performing quality estimation systems such as QuEst (Specia et al., 2013) and QuEst++ (Specia et al., 2015) were heavily dependent on linguistic processing and feature engineering. These features were fed into traditional machine-learning algorithms like support vector regression and randomised decision trees (Specia et al., 2013), which then determined

the quality of a translation. Even though, these approaches provide good results, they are no longer the state of the art, being replaced in recent years by neural-based QE systems which usually rely on little or no linguistic processing. For example the best-performing system at the WMT 2017 shared task on QE was POSTECH, which is purely neural and does not rely on feature engineering at all (Kim et al., 2017).

In order to achieve high results, approaches such as POSTECH require extensive pre-training, which means they depend on large parallel data and are computationally intensive (Ive et al., 2018). TransQuest, our QE framework removes this dependency on large parallel data by using crosslingual embeddings (Ranasinghe et al., 2020) that are already fine-tuned to reflect properties between languages (Ruder et al., 2019). Ranasinghe et al. (2020) show that by using them, TransQuest eases the burden of having complex neural network architectures, which in turn entails a reduction of the computational resources. That paper also shows that TransQuest performs well in transfer learning settings where it can be trained on language pairs for which we have resources and applied successfully on less resourced language pairs.

The remainder of the paper is structured as follows. The dataset used in the competition is briefly discussed in Section 2. In Section 3 we present the TransQuest framework and the methodology employed to train it. This is followed by the evaluation results and their discussion in Section 4. The paper finishes with conclusions and ideas for future research directions.

2 Dataset

The dataset for the Sentence-Level Direct Assessment shared task is composed of data

extracted from Wikipedia for six language pairs, consisting of high-resource languages English-German (En-De) and English-Chinese (En-Zh), medium-resource languages Romanian-English (Ro-En) and Estonian-English (Et-En), and low-resource languages Sinhala-English (Si-En) and Nepalese-English (Ne-En), as well as a Russian-English (Ru-En) dataset which combines articles from Wikipedia and Reddit (Specia et al., 2020). Each language pair has 7,000 sentence pairs in the training set, 1,000 sentence pairs in the development set and another 1,000 sentence pairs in the testing set. Each translation was rated with a score between 0 and 100 according to the perceived translation quality by at least three translators (Fomicheva et al., 2020). The DA scores were standardised using the z-score. The quality estimation systems have to predict the mean DA z-scores of the test sentence pairs (Specia et al., 2020).

3 Methodology

This section presents the methodology used to develop our quality estimation methods. Our methodology is based on TransQuest our recently introduced QE framework (Ranasinghe et al., 2020). We first briefly describe the neural network architectures TransQuest proposed, followed by the training details. More details about the framework can be found in (Ranasinghe et al., 2020).

3.1 Neural Network Architectures

The *TransQuest* framework that is used to implement the two architectures described here relies on the XLM-R transformer model (Conneau et al., 2020) to derive the representations of the input sentences (Ranasinghe et al., 2020). The XLM-R transformer model takes a sequence of no more than 512 tokens as input, and outputs the representation of the sequence. The first token of the sequence is always [CLS] which contains the special embedding to represent the whole sequence, followed by embeddings acquired for each word in the sequence. As shown below, proposed neural network architectures of TransQuest can utilise both the embedding for the [CLS] token and the embeddings generated for each word (Ranasinghe et al., 2020). The output of the transformer (or transformers for **SiameseTransQuest** described below), is fed into a simple output layer which is used to estimate the quality of translation. The

way the XLM-R transformer is used and the output layer are different in the two instantiations of the framework. We describe each of them below. The fact that TransQuest does not rely on a complex output layer makes training its architectures much less computationally intensive than alternative solutions. The *TransQuest* framework is open-source, which means researchers can easily propose alternative architectures to the ones TransQuest presents (Ranasinghe et al., 2020).

Both neural network architectures presented below use the pre-trained XLM-R models released by HuggingFace’s model repository (Wolf et al., 2019). There are two versions of the pre-trained XLM-R models named XLM-R-base and XLM-R-large. Both of these XLM-R models cover 104 languages (Conneau et al., 2020), potentially making it very useful to estimate the translation quality for a large number of language pairs.

TransQuest implements two different neural network architectures (Ranasinghe et al., 2020) to perform sentence-level translation quality estimation as described below. The architectures are presented in Figure 1.

1. **MonoTransQuest (MTransQuest)**: The first architecture proposed uses a single XLM-R transformer model and is shown in Figure 1a. The input of this model is a concatenation of the original sentence and its translation, separated by the [SEP] token. TransQuest proposes three pooling strategies for the output of the transformer model: using the output of the [CLS] token (CLS-strategy); computing the mean of all output vectors of the input words (MEAN-strategy); and computing a max-over-time of the output vectors of the input words (MAX-strategy) (Ranasinghe et al., 2020). The output of the pooling strategy is used as the input of a softmax layer that predicts the quality score of the translation. TransQuest used mean-squared-error loss as the objective function (Ranasinghe et al., 2020). Similar to Ranasinghe et al. (2020), the early experiments we carried out demonstrated that the CLS-strategy leads to better results than the other two strategies for this architecture. Therefore, we used the embedding of the [CLS] token as the input of a softmax layer.
2. **SiameseTransQuest (STransQuest)**: The second approach proposed in TransQuest

relies on the Siamese architecture depicted in Figure 1b which has shown promising results in monolingual semantic textual similarity tasks (Reimers and Gurevych, 2019; Ranasinghe et al., 2019). For this, we fed the original text and the translation into two separate XLM-R transformer models. Similarly to the previous architecture, we experimented with the same three pooling strategies for the outputs of the transformer models (Ranasinghe et al., 2020). TransQuest then calculates the cosine similarity between the two outputs of the pooling strategy. TransQuest used mean-squared-error loss as the objective function. Similar to Ranasinghe et al. (2020) in the initial experiments we carried out with this architecture the MEAN-strategy showed better results than the other two strategies. For this reason, we used the MEAN-strategy for our experiments. Therefore, cosine similarity is calculated between the mean of all output vectors of the input words produced by each transformer.

3.2 Training Details

We used the same set of configurations suggested in Ranasinghe et al. (2020) for all the language pairs evaluated in this paper in order to ensure consistency between all the languages. This also provides a good starting configuration for researchers who intend to use TransQuest on a new language pair. In both architectures, we used a batch-size of eight, Adam optimiser with learning rate $2e-5$, and a linear learning rate warm-up over 10% of the training data. The models were trained using only training data. Furthermore, they were evaluated while training using an evaluation set that had one fifth of the rows in training data. We performed early stopping if the evaluation loss did not improve over ten evaluation rounds. All of the models were trained for three epochs. For some of the experiments, we used an Nvidia Tesla K80 GPU, whilst for others we used an Nvidia Tesla T4 GPU. This was purely based on the availability of the hardware and it was not a methodological decision.

3.3 Implementation Details

The TransQuest framework was implemented using Python 3.7 and PyTorch 1.5.0. To integrate the functionalities of the transformers we used the

version 3.0.0 of the HuggingFace’s Transformers library. The implemented framework is available on GitHub¹.

4 Evaluation, Results and Discussion

This section presents the evaluation results of our architectures and the fine tuning strategies that can be used to improve the results. We first evaluate the TransQuest framework with the default setting (Section 4.1). Next we evaluate an ensemble setting of TransQuest in Section 4.2. We finally assess the performance of TransQuest with augmented data. We conclude the section with a discussion of the results.

The evaluation metric used was the Pearson correlation (r) between the predictions and the gold standard from the test set, which is the most commonly used evaluation metric in WMT quality estimation shared tasks (Specia et al., 2018; Fonseca et al., 2019). We report the Pearson correlation values that we obtained from CodaLab, the hosting platform of the WMT 2020 QE shared task. As a baseline we compare our results with the performance of OpenKiwi as reported by the task organisers (Specia et al., 2020).

4.1 TransQuest with Default settings

The first evaluation we carried out was for the default configurations of the TransQuest framework where we used the training set of each language to build a quality estimation model using XLM-R-large transformer model and we evaluated it on a test set from the same language.

The results for each language with *default* settings are shown in row I of Table 1. The results indicate that both architectures proposed in *TransQuest* outperform the baseline, OpenKiwi, in all the language pairs. From the two architectures, *MTransQuest* performs slightly better than *STransQuest*.

As shown in Table 1, *MTransQuest* gained $\approx 0.2-0.3$ Pearson correlation boost over OpenKiwi in all the language pairs. Additionally, *MTransQuest* achieves ≈ 0.4 Pearson correlation boost over OpenKiwi in the low-resource language pair Ne-En.

¹TransQuest GitHub repository - <https://github.com/tharindudr/transquest>

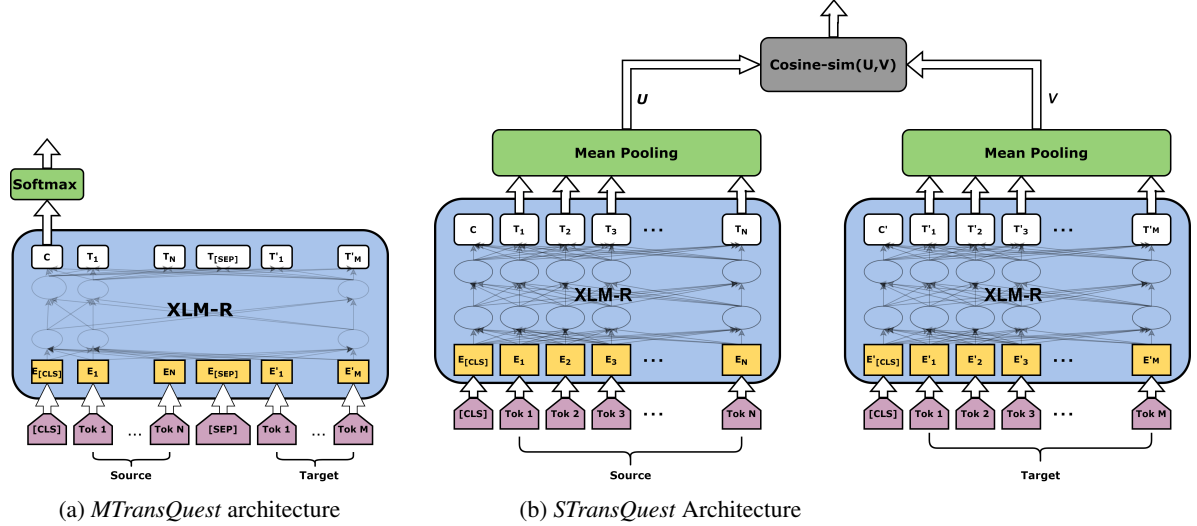


Figure 1: Two architectures of the *TransQuest* framework.

Method	Low-resource		Mid-resource			High-resource	
	Si-En	Ne-En	Et-En	Ro-En	Ru-En	En-De	En-Zh
I MTransQuest	0.6525	0.7914	0.7748	0.8982	0.7734	0.4669	0.4779
STransQuest	0.5957	0.7081	0.6804	0.8501	0.7126	0.3992	0.4067
II MTransQuest-base	0.6412	0.7823	0.7651	0.8715	0.7593	0.4421	0.4593
STransQuest-base	0.5773	0.6853	0.6692	0.8321	0.6962	0.3832	0.3975
III MTransQuest \otimes	0.6661	0.8023	0.7876	0.8988	0.7854	0.4862	0.4853
STransQuest \otimes	0.6001	0.7132	0.6901	0.8629	0.7248	0.4096	0.4159
IV MTransQuest \otimes - Aug	0.6849	0.8222	0.8240	0.9082	0.8082	0.5539	0.5373
STransQuest \otimes - Aug	0.6241	0.7354	0.7239	0.8621	0.7458	0.4457	0.4658
V OpenKiw	0.3737	0.3860	0.4770	0.6845	0.5479	0.1455	0.1902

Table 1: Pearson (r) correlation between *TransQuest* algorithm predictions and human DA judgments. Best results for each language (any method) are marked in bold. Rows I, II, III and IV indicate the different settings of *TransQuest*, explained in Sections 4.1-4.3. OpenKiw baseline results are in Row V.

4.2 TransQuest with Ensemble

Transformers have been proven to provide better results when experimented with ensemble techniques (Xu et al., 2020). In order to improve the results of *TransQuest* we too followed an ensemble approach which consisted of two steps. We conducted these steps for both architectures in *TransQuest*.

1. We train *TransQuest* using the pre-trained XLM-R-base transformer model instead of the XLM-R-large transformer model in the *TransQuest* default setting. We report the results from the two architectures from this step in row II of Table 1 as MTransQuest-base and STransQuest-base.

2. We perform a weighted average ensemble for the output of the default setting and the output we obtained from step 1. We experimented on weights 0.8:0.2, 0.6:0.4, 0.5:0.5 on the output of the default setting and output from the step 1 respectively. Since the results we got from XLM-R-base transformer model are slightly worse than the results we got from default setting we did not consider the weight combinations that gives higher weight to XLM-R-base transformer model results. We obtained best results when we used the weights 0.8:0.2. We report the results from the two architectures from this step in row III of Table 1 as MTransQuest \otimes and STransQuest \otimes .

As shown in Table 1 both architectures in TransQuest with ensemble setting gained ≈ 0.01 - 0.02 Pearson correlation boost over the default settings for all the language pairs.

4.3 TransQuest with Data Augmentation

All of the languages had 7,000 training instances that we used in the above mentioned settings in TransQuest. To experiment how TransQuest performs with more data, we trained TransQuest on a data augmented setting. Alongside the training, development and testing datasets, the shared task organisers also provided the parallel sentences which were used to train the neural machine translation system in each language. In the data augmentation setting, we added the sentence pairs from that neural machine translation system training file to training dataset we used to train TransQuest. In order to find the best setting for the data augmentation we experimented with adding 1000, 2000, 3000, up to 5000 sentence pairs randomly. Since the ensemble setting performed better than the default setting of TransQuest, we conducted this data augmentation experiment on the ensemble setting. We assumed that the sentence pairs added from the neural machine translation system training file have maximum translation quality.

Up to 2000 sentence pairs the results continued to get better. However, adding more than 2000 sentence pairs did not improve the results. We did not experiment with adding any further than 5000 sentence pairs to the training set since the timeline of the competition was tight. We were also aware that adding more sentence pairs with the maximum translation quality to the training file will make it imbalance and affect the performance of the machine learning models negatively. We report the results from the two architectures from this step in row IV of Table 1 as MTransQuest \otimes -Aug and STransQuest \otimes -Aug.

This setting provided the best results for both architectures in TransQuest for all of the language pairs. As shown in Table 1 both architectures in TransQuest with the data augmentation setting gained ≈ 0.01 - 0.09 Pearson correlation boost over the default settings for all the language pairs. Additionally, MTransQuest \otimes -Aug achieves ≈ 0.09 Pearson correlation boost over default MTransQuest in the high-resource language pair En-De.

4.4 Error analysis

In an attempt to better understand the performance and limitations of *TransQuest* we carried out an error analysis on the results obtained on Romanian - English and Sinhala - English. The choice of language pairs we analysed was determined by the availability of native speakers to perform this analysis. We focused on the cases where the difference between the predicted score and expected score was the greatest. This included both cases where the predicted score was underestimated and overestimated.

Analysis of the results does not reveal very clear patterns. The largest number of errors seem to be caused by the presence of named entities in the source sentences. In some cases these entities are mishandled during the translation. The resulting sentences are usually syntactically correct, but semantically odd. Typical examples are RO: *În urmă explorărilor Căpitanului James Cook, Australia și Noua Zeelandă au devenit ținte ale colonialismului britanic. (As a result of Captain James Cook's explorations, Australia and New Zealand have become the targets of British colonialism.)* - EN: *Captain James Cook, Australia and New Zealand have finally become the targets of British colonialism.* (expected: -1.2360, predicted: 0.2560) and RO: *O altă problemă importantă cu care trupele Antantei au fost obligate să se confrunte a fost malaria. (Another important problem that the Triple Entente troops had to face was malaria.)* - EN: *Another important problem that Antarctic troops had to face was malaria.* (expected: 0.2813, predicted: -0.9050). In the opinion of the authors of this paper, it is debatable whether the expected scores for these two pairs should be so different. Both of them have obvious problems and cannot be clearly understood without reading the source. For this reason, we would expect that both of them have low scores. Instances like this also occur in the training data. As a result of this, it may be that *TransQuest* learns contradictory information, which in turn leads to errors at the testing stage.

A large number of problems are caused by incomplete source sentences or input sentences with noise. For example the pair RO: *thumbright250pxDrapelul cu fâșiile în poziție verticală (The flag with strips in upright position)* - EN: *ghtghntness 250pxDrapel with strips in upright position* has an expected score of 0.0595, but

our method predicts -0.9786. Given that only *ghtghness 250pxDrapel* is wrong in the translation, the predicted score is far too low. In an attempt to see how much this noise influences the result, we run the system with the pair *RO: Drapelul cu fâșiile în poziție verticală - EN: Drapel with strips in upright position*. The prediction is 0.42132, which is more in line with our expectations given that one of the words is not translated.

Similar to Ro-En, in Si-En the majority of problems seem to be caused by the presence of named entities in the source sentences. For an example in the English translation: *But the disguised Shiv will help them securely establish the statue*. (expected: 1.3618, predicted: -0.008), the correct English translation would be *But the disguised Shividru will help them securely establish the statue*. Only the named entity *Shividru* is translated incorrectly, therefore the annotators have annotated the translation with a high quality. However TransQuest fails to identify that. Similar scenarios can be found in English translations *Kamala Devi Chattopadhyay spoke at this meeting, Dr. Ann*. (expected:1.3177, predicted:-0.2999) and *The Warrior Falls are stone's, halting, heraldry and stonework rather than cottages. The cathedral manor is navigable places* (expected:0.1677, predicted:-0.7587). It is clear that the presence of the named entities seem to confuse the algorithm we used, hence it needs to handle named entities in a proper way.

5 Conclusion

In this paper we evaluated different settings of *TransQuest* in sentence-level direct quality assessment. We showed that ensemble results with XLM-R-base and XLM-R-large with data augmentation techniques can improve the performance of TransQuest framework.

The official results of the competition show that *TransQuest* won the first place in all the language pairs in Sentence-Level Direct Assessment task. *TransQuest* is the sole winner in En-Zh, Ne-En and Ru-En language pairs and the multilingual track. For the other language pairs (En-De, Ro-En, Et-En and Si-En) it shares the first place with another system, whose results are not statistically different from ours. The full results of the shared task can be seen in [Specia et al. \(2020\)](#).

In the future, we plan to experiment more with the data augmentation settings. We are

interested in augmenting the training file with semantically similar sentences to the test set rather than augmenting with random sentence pairs as we did in this paper. As shown in the error analysis in Section 4.4 the future releases of the framework need to handle named entities properly. We also hope to implement *TransQuest* in document level quality estimation too.

References

- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard 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.
- M Fomicheva, L Specia, S Sun, L Yankovskaya, F Blain, F Guzmán, M Fishel, N Aletras, and V Chaudhary. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, Vol 8 (2020).
- 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.
- Julia Ive, Frédéric Blain, and Lucia Specia. 2018. [deepQuest: A framework for neural-based quality estimation](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3146–3157, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- 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-Hyeok 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.
- Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2019. [Semantic textual similarity with](#)

- Siamese neural networks. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1004–1011, Varna, Bulgaria. INCOMA Ltd.
- Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. Transquest: Translation quality estimation with cross-lingual transformers. In *Proceedings of the 28th International Conference on Computational Linguistics*.
- Nils Reimers and Iryna Gurevych. 2019. **Sentence-BERT: Sentence embeddings using Siamese BERT-networks**. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. **A survey of cross-lingual word embedding models**. *J. Artif. Int. Res.*, 65(1):569–630.
- Lucia Specia, Frédéric Blain, Marina Fomicheva, Erick Fonseca, Vishrav Chaudhary, Francisco Guzmán, and André FT Martins. 2020. Findings of the wmt 2020 shared task on quality estimation. In *Proceedings of the Fifth Conference on Machine Translation: Shared Task Papers*.
- Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón Astudillo, and André F. T. Martins. 2018. **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 Computational Linguistics.
- Lucia Specia, Gustavo Paetzold, and Carolina Scarton. 2015. **Multi-level translation quality prediction with QuEst++**. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 115–120, Beijing, China. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.
- Lucia Specia, Kashif Shah, Jose G.C. de Souza, and Trevor Cohn. 2013. **QuEst - a translation quality estimation framework**. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84, Sofia, Bulgaria. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.
- Yige Xu, Xipeng Qiu, Ligao Zhou, and Xuanjing Huang. 2020. Improving bert fine-tuning via self-ensemble and self-distillation. *arXiv preprint arXiv:2002.10345*.