“Bilingual Expert” Can Find Translation Errors

Kai Fan, Bo Li, Fengming Zhou, Jiayi Wang
{k.fan,shiji.lb,zfm104435,joanne.wjy}@alibaba-inc.com
Alibaba Group Inc.

Abstract

Recent advances in statistical machine translation via the adoption of neural sequence-to-sequence models empower the end-to-end system to achieve state-of-the-art in many WMT benchmarks. The performance of such machine translation (MT) system is usually evaluated by automatic metric BLEU when the golden references are provided for validation. However, for model inference or production deployment, the golden references are prohibitively available or require expensive human annotation with bilingual expertise. In order to address the issue of quality evaluation (QE) without reference, we propose a general framework for automatic evaluation of translation output for most WMT quality evaluation tasks. We first build a conditional target language model with a novel bidirectional transformer, named ***neural bilingual expert*** model, which is pre-trained on large parallel corpora for feature extraction. For QE inference, the bilingual expert model can simultaneously produce the joint latent representation between the source and the translation, and real-valued measurements of possible erroneous tokens based on the prior knowledge learned from parallel data. Subsequently, the features will further be fed into a simple Bi-LSTM predictive model for quality evaluation. The experimental results show that our approach achieves the state-of-the-art performance in the quality estimation track of WMT 2017/2018.

The neural machine translation (NMT) in a sequence-to-sequence fashion, providing an end-to-end learning approach for automatic translation system, has accomplished great success to potentially overcome many of the weaknesses of conventional phrase-based translation, and claimed achieving human parity for certain language pairs (Wu et al. 2016; Hassan et al. 2018). However, current MT systems are still not perfect to meet the real-world applications without human post-editing (an example is illustrate in Fig. 1). Apparently, additional error correction is needed for even such a simple sentence. A possible solution to take the advantage of the existing MT technologies is to collaborate with human translators within a computer-assisted translation (CAT) (Barrachina et al. 2009). In such cases, translation quality estimation (QE) plays a critical role in CAT to reduce human efforts, thereby increasing productivity (Specia 2011). Either the global sentence score or the fine-grained word tags can guide the CAT as an evidence to indicate whether a machine translation output requires further manual post-editing, even which particular token needs special correction.

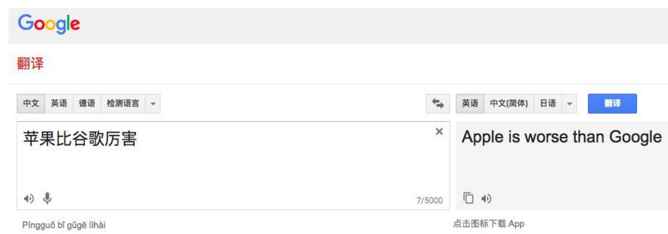


Figure 1: Chinese to English translation test at Google online system on 04/03/2018: the correct translation should be “Apple is better than Google”. Notice this bug has been fixed.

One traditional direction for translation quality estimation is to formulate the sentence level score or word level tags prediction as a constraint regression or sequence labeling problem respectively (Bojar et al. 2017). The classical baseline model is to use the QuEst++ (Specia, Paetzold, and Scarton 2015) with two modules: rule based feature extractor and scikit-learn [1] algorithms. Similarly, the recent predictor-estimator model (Kim et al. 2017) is a recurrent neural network (RNN) based feature extractor and quality estimation model, ranking first place at WMT 2017.

Another promising direction is to build a multi-task learning model to incorporate quality estimation task with automatic post-editing (APE) together (Hokamp 2017; Tan et al. 2017), achieving the goal of CAT eventually. In this paper, we will first adopt the traditional single task framework to describe our model. In the experimental section, we also propose an extension to support multi-task learning, different from using QE ahead of APE (Chatterjee et al. 2018).

However, the final prediction model for scoring or tagging is not the main contribution in our work. Since there are many publicly available bilingual corpora, we can readily build a conditional language model as a more robust feature extractor. The high level joint latent representation of two parallel sentences between the source and the target can hopefully capture either the alignment or semantic information. In contrast, when a source and a low-quality machine

[1] http://scikit-learn.org/

translation are fed into the pre-trained language model, the distribution of latent features is very likely to be different from the one of grammatically correct target. Intuitively, most people learn the foreign language from reading the correct translation to their native language. Gradually, they may acquire the ability to be aware of the abnormality when errors appear in an unseen sentence. Additionally, we design four-dimensional token mis-matching features from the pre-trained model, measuring the difference between what the bilingual expert model will predict and the actual token of machine translation output.

Particularly, we use the recent proposed self-attention mechanism and transformer neural networks (Vaswani et al. 2017) to build our conditional language model – *neural bilingual expert*. The model consists of one transformer encoder for source sentence and a novel bidirectional transformer decoder for target sentence. The idea of constructing deep bidirectional language model comes from ELMO (Peters et al. 2018), which is originally pre-trained on a large monolingual corpus. In parallel, OpenAI recently applied the transformer decoder to train monolingual language model (Radford et al. 2018) [2]. However, the key difference is that only the single directional transformer decoder (Vaswani et al. 2017) is used. Approximately, our decoder can be considered as a combination of ELMO and OpenAI's work. Similar idea is discussed in (Shen et al. 2018). Empirically, we found such a pre-training strategy can improve the performance of the downstream quality estimation model as well, even a one-layer Bi-LSTM model.

Besides feature learning, the conditional language model can also play the role of automatic post-editing. Since shifts (word order errors) were not annotated as such (but rather as deletions and insertions) to avoid introducing noise in the annotation, missing tokens in the machine translations, as indicated by the TER tool (Snover et al. 2006) are annotated as follows: after each token in the sentence and at sentence start, a gap tag is placed. In this way, we can use the same network structure of conditional language model to enable the gap prediction (insertions) for missing token of translation output conditional on the source sentence. Using the deletion operation of word level together, we are literally trying to predict post-editing.

This paper makes the following contributions: i) we propose a bidirectional transformer based conditional language model and pre-train it on available large bilingual corpora, which can further be used as automatic post-editing model. ii) we address the importance of four-dimensional mis-matching features, and in the experiments of only using these features, we found that our approach can achieve comparable results with first ranked model in WMT 2017 QE task. iii) we develop a differentiable word-level quality estimation model to support data preprocessing with byte-pair-encoding (BPE) tokenization, bridging the gap between word and BPE token. iv) extensive experiments on real-world datasets (e.g., the corpus domain includes IT and pharmacy) demonstrate our method is effective and obtain state-of-the-art performance in many tasks.

[2]This paper is publicly released at 06/11/2018, and the WMT 2018 QE task deadline is 06/22/2018.

Quality Estimation for Machine Translation

Given the bilingual corpus, from the statistical view we can formulate the machine translation system as $p(\mathbf{t}|\mathbf{s}) = p(\mathbf{t}|\mathbf{z})p(\mathbf{z}|\mathbf{s})$, where $\mathbf{s}$ represents the tokens sequence of source sentence, $\mathbf{t}$ for target sentence, and $\mathbf{z}$ is the latent variable to represent the encoded source sentence. Therefore, $p(\mathbf{z}|\mathbf{s})$ and $p(\mathbf{t}|\mathbf{z})$ can be practically considered as the encoder and decoder. In the quality estimation task of machine translation, the machine translation system is agnostic and the training dataset is given in the format of triplet $(\mathbf{s}, \mathbf{m}, \mathbf{t})$, where $\mathbf{m}$ is the translation output from the unknown machine translation system with the input $\mathbf{s}$, and $\mathbf{t}$ represents the human post-edited sentence based on $\mathbf{s}$ and $\mathbf{m}$. Notice we abuse using notation $\mathbf{t}$ to refer both golden reference and human post-edited sentence.

In general, the quality of $\mathbf{m}$ can be evaluated either in the global sentence level or the fine-grained word level. The sentence level score is calculated by the percentage of edits needed to fix for $\mathbf{m}$, denoted as HTER. The word level evaluation is framed as the binary classification problem to distinguish between 'OK' and 'BAD' tokens for translation output. Particularly, the binary word-level labels are generated by using the alignments provided by the TER tool [3] between $\mathbf{m}$ and $\mathbf{t}$. Notice the sentence HTER and word labels can deterministically be calculated by the TER tool when $\mathbf{m}$ and $\mathbf{t}$ are both present. However, in inference only the source sentence $\mathbf{s}$ and machine translation $\mathbf{m}$ are avaiable, thus essentially requiring to develop an automatic method for quality estimation of machine translation output at runtime, without relying on any reference translation.

We can assume that the training data contains the tuple $(\mathbf{s}, \mathbf{m}, \mathbf{t}, h, \mathbf{y})$, where h is a scalar to represent HTER, and $\mathbf{y}$ is a binary vector to indicate the 'OK/BAD' labels of machine translation output. Considering the inference scenario, our task is to learn a regression model $p(h|\mathbf{s}, \mathbf{m})$ and a sequence labeling model $p(\mathbf{y}|\mathbf{s}, \mathbf{m})$.

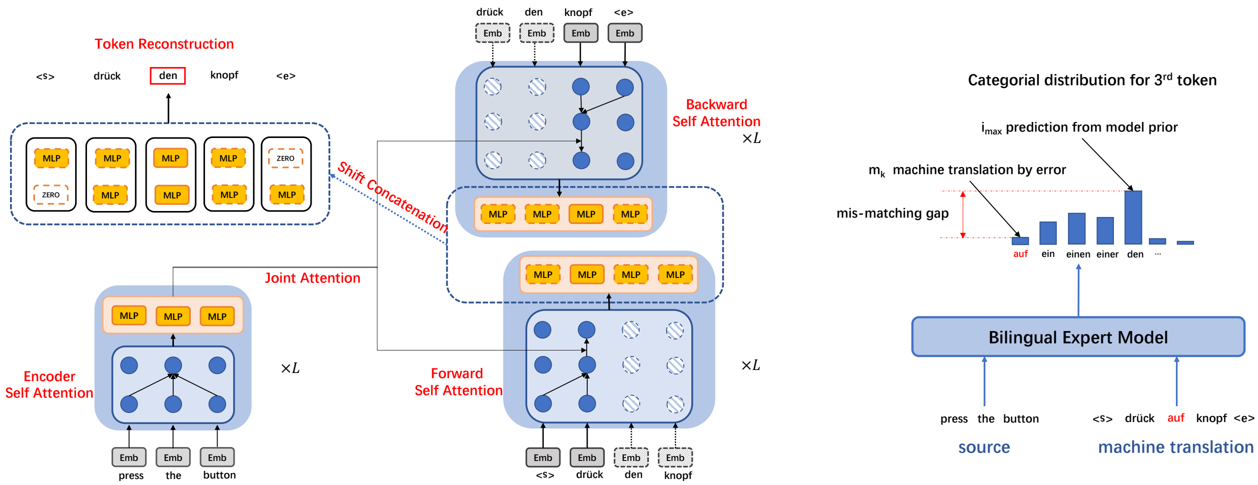
Bilingual Expert Model

In this section, we will first highlight how to train a neural bilingual expert model with a parallel corpus including $(\mathbf{s}, \mathbf{t})$ pairs. By default of QE task, the statistical/neural machine translation system $p(\mathbf{t}|\mathbf{z})p(\mathbf{z}|\mathbf{s})$ is unknown, but in representation learning we are usually interested in the latent variable $\mathbf{z}$, whose posterior may contain the deep semantic representation between source and target language and be beneficial to many downstream tasks (Hill, Cho, and Korhonen 2016). According to the Bayes rule, we can write the posterior distribution of the latent variable as,

$$p(\mathbf{z}|\mathbf{t}, \mathbf{s}) = \frac{p(\mathbf{t}|\mathbf{z})p(\mathbf{z}|\mathbf{s})}{p(\mathbf{t}|\mathbf{s})} \tag{1}$$

where the integral $p(\mathbf{t}|\mathbf{s}) = \int p(\mathbf{t}|\mathbf{z})p(\mathbf{z}|\mathbf{s})\mathrm{d}\mathbf{z}$ is usually intractable. Instead of exact inference, we propose a variational distribution $q(\mathbf{z}|\mathbf{t}, \mathbf{s})$ to approximate true posterior by

[3]settings: tokenized, case insensitive, exact matching only, disabling shifts by using the '-d 0' option



(a) Bidirectional Transformer (b) Mis-matching Feature

Figure 2: (a) The encoder with self-attention is basically identical to the transformer NMT. The forward and backward self-attentions mimic the structure of bidirectional RNN, implemented by the left to right and right to left masked softmax respectively. For example, in the left panel we can see that token "den" of target is desired to predict, but only the information of source and all the other tokens in the target will be propagated to the final layer for prediction. *Notice* that the detailed structures like positional encoding, skip-connection, layer normalization, are omitted for clarity, but we still implemented it in our experiments. (b) The mis-matching feature measures the difference between what the "bilingual expert" thinks the target token is and the actual token of translation output.

minimizing exclusive Kullback-Leibler (KL) divergence.

$$\min D_{KL}(q(\mathbf{z}|\mathbf{t},\mathbf{s})\|p(\mathbf{z}|\mathbf{t},\mathbf{s})) \tag{2}$$

Rather than optimizing the objective function above, we can equivalently maximize the following loss.

$$\max \mathbb{E}_{q(\mathbf{z}|\mathbf{t},\mathbf{s})}[p(\mathbf{t}|\mathbf{z})] - D_{KL}(q(\mathbf{z}|\mathbf{t},\mathbf{s})\|p(\mathbf{z}|\mathbf{s})) \tag{3}$$

A nice property of the new objective is that it is unnecessary to parameterize or estimate the implicit machine translation model $p(\mathbf{t}|\mathbf{s})$. The first expectation term in (3) can be readily considered as a conditional auto-encoder system if we use one sample Monte Carlo integration during optimization, and the second KL term can be analytically computed if we practically set the prior $p(\mathbf{z}|\mathbf{s})$ as standard Gaussian distribution, playing as a model regularization. Furthermore, if we omit the conditional information $\mathbf{s}$, the objective exactly reduces to amortized variational inference or variational auto-encoders (VAE) framework (Kingma and Welling 2013). In analogous to most VAE models, the expected log-likelihood is commonly approximated by a practical surrogated loss,

$$\mathbb{E}_{q(\mathbf{z}|\mathbf{t},\mathbf{s})}[p(\mathbf{t}|\mathbf{z})] \approx p(\mathbf{t}|\tilde{\mathbf{z}}), \;\; \tilde{\mathbf{z}} \sim q(\mathbf{z}|\mathbf{t},\mathbf{s}) \tag{4}$$

Next, we will show the details of constructing the other two probability distributions appeared in (3) with self-attention based transformer neural networks.

Bidirectional Transformer

Transformer (Vaswani et al. 2017), unlike RNN, is based solely on attention mechanisms, dispensing with recurrence and convolution, becoming the state-of-the-art NMT model in the recent WMT 2018 News translation competition, where almost all the first-place systems for different language pairs are based on transformer [4]. (Vaswani et al. 2017) claims that self-attention mechanism has several advantages: first, its gating or multiplication enables crisp error propagation; second, it can replace sequence-aligned recurrence entirely; third, from the implementation perspective, it is trivial to be parallelized. When we design the bidirectional transformer, we are trying to keep the three properties remained.

The overall model architecture of bidirectional transformer is illustrated in Fig. 2(a). There are three modules in total, self-attention encoder for the source sentence, forward and backward self-attention encoders for target sentence, and the reconstructor for the target sentence, where the first two modules represent the proposed posterior approximation $q(\mathbf{z}|\mathbf{s},\mathbf{t})$ and the third reconstruction process corresponds to $p(\mathbf{t}|\mathbf{z})$. To make the inference efficient, we explicitly assume the conditional independence with the following factorization,

$$p(\mathbf{t}|\mathbf{z}) = \prod_k p(t_k|\overrightarrow{\mathbf{z}_k}, \overleftarrow{\mathbf{z}_k}) \tag{5}$$

$$q(\mathbf{z}|\mathbf{s},\mathbf{t}) = \prod_k q(\overrightarrow{\mathbf{z}_k}|\mathbf{s},\mathbf{t}_{<k})q(\overleftarrow{\mathbf{z}_k}|\mathbf{s},\mathbf{t}_{>k}) \tag{6}$$

where the bidirectional latent variable $\mathbf{z}$ includes all $\{\overrightarrow{\mathbf{z}_k}, \overleftarrow{\mathbf{z}_k}\}$. Our factorization is slightly different from ELMO (Peters et al. 2018), where they use a finer grained form $\prod_k p(t_k|\overrightarrow{\mathbf{z}_k})p(t_k|\overleftarrow{\mathbf{z}_k})$ but with the shared parameters between forward and backward $p(t_k|\cdot)$.

Note that $\overrightarrow{\mathbf{z}_k}, \overleftarrow{\mathbf{z}_k}$ are sampled from $q(\overrightarrow{\mathbf{z}_k}|\mathbf{s},\mathbf{t}_{<k})$ and $q(\overleftarrow{\mathbf{z}_k}|\mathbf{s},\mathbf{t}_{>k})$ respectively, which are assumed to follow the

[4] http://matrix.statmt.org

Gaussian distribution. However, in order to reduce the number of parameters, we assume $q(\cdot|\cdot) \sim \mathcal{N}(\mu(\mathbf{s},\mathbf{t}), \sigma^2\mathbf{I})$, where the mean $\mu(\mathbf{s},\mathbf{t})$ is learned in amortized way. In other words, we can efficiently implement stochastic layer as the deterministic one via dropout training with additive fixed variance Gaussian noise (Srivastava et al. 2014). During the inference, it is still required to add the extra Gaussian noise, since we define a stochastic latent variable $\mathbf{z}$.

Translation Error Prediction

Once the bilingual expert model has been fully trained on large parallel corpora, we can reasonably assume the model will predict higher likelihood for the correct target token, given the source and other context of the target, if only very few tokens are incorrect. Therefore, we will use the prior knowledge learned by bilingual expert to extract the features for subsequent translation error prediction. Basically, we will derive three sequential (token-wise) features based upon the pre-trained model with the source sentence and translation output as input, i.e. $(\mathbf{s},\mathbf{m})$ pairs.

Naturally, the first feature, the latent representation $\mathbf{z}_k = \text{Concat}(\overrightarrow{\mathbf{z}_k}, \overleftarrow{\mathbf{z}_k})$, comes from the model. As we discussed previously, the entire latent variable $\mathbf{z}$ should generally summarize the information of the source and the target. Additionally, the subtle $\mathbf{z}_k$ only contains the information from the source and the context around the k-th token in the target, being greatly beneficial to our manually extracted mis-matching features.

In (Peters et al. 2018), the token embedding is also used as one linear component to compute the final ELMO feature. However, in our case that translation output is fed into the model, it is not guaranteed that every single token is correct. Therefore, we design a different token embedding feature following the rationale of information flow within latent variable $\mathbf{z}_k$. In fact, we use the embedding concatenation of two neighbor tokens $\text{Concat}(\mathbf{e}_{t_{k-1}}, \mathbf{e}_{t_{k+1}})$. Since the possibly erroneous translation may mislead the model in the downstream quality estimation task, we did not extract any information from current token t_k. More importantly, the correct syntax representation of the token which is supposed to be translated should come from the source sentence, which has been encoded into $\mathbf{z}$ via joint attention.

Mis-matching Feature

Besides the proposed model features that are just the nodes within the computational graph of bidirectional transformer, we experimentally found another crucial feature that can directly measure how the prior knowledge from the well-trained bilingual expert model is different from translation output. To make it concrete, $p(t_k|\cdot)$ follows the categorical distribution with the number of classes equal to the vocabulary size. Since we pre-train bilingual expert model on golden reference, the objective (3) is theoretically to maximize the likelihood of each $p(t_k|\cdot)$, which achieves its maximum when t_k is ground truth. Intuitively, we should have $p(m_k|\cdot) \leq p(t_k|\cdot)$ if $m_k \neq t_k$, illustrated in Fig 2(b). Following this intuition, we propose the mis-matching features.

Suppose $\mathbf{l}_k$ is the logits vector before applying the softmax operation, i.e. $p(t_k|\cdot) \sim \text{Categorical}(softmax(\mathbf{l}_k))$, thus we can define the 4-dimensional mis-matching features,

$$\mathbf{f}_k^{mm} = (\mathbf{l}_{k,m_k}, \mathbf{l}_{k,i_{\max}}, \mathbf{l}_{k,m_k} - \mathbf{l}_{k,i_{\max}}, \mathbb{I}_{m_k \neq i_{\max}}) \quad (7)$$

where we use m_k to represent the vocabulary id of the k-th token in translation output, $i_{\max} = \arg\max_k \mathbf{l}_k$ is the token id that bilingual expert predicts, and $\mathbb{I}$ is indicator function. Therefore, these four values will directly reflect the differences or errors. Apparently, if the machine translation coincides with the bilingual expert, the first 2 values should be the same and the last two values should both be 0, representing soft and hard differences respectively. In our experiments, we found the quality estimation model can achieve acceptable result even with the mis-matching features alone.

Algorithm 1 Translation Quality Estimation with Bi-Transformer and Bi-LSTM

Require: QE training data $(\mathbf{s}, \mathbf{m}, \mathbf{t}, h, \mathbf{y})_{1:M}$, QE inference data $(\mathbf{s}, \mathbf{m})$, and parallel corpus $(\mathbf{s}, \mathbf{t})_{1:N}$.

1: Combine the parallel corpus with 10 times QE training parallel corpus $C = (\mathbf{s}, \mathbf{t})_{1:N} \bigcup 10 \times (\mathbf{s}, \mathbf{t})_{1:M}$

2: Pre-train bilingual expert model via bidirectional transformer on the combined corpus C.

3: Extract features $\mathbf{f}_k = \text{Concat}(\overrightarrow{\mathbf{z}_k}, \overleftarrow{\mathbf{z}_k}, \mathbf{e}_{t_{k-1}}, \mathbf{e}_{t_{k+1}}, \mathbf{f}_k^{mm})$ given $(\mathbf{s}, \mathbf{m})$.

4: Train Bi-LSTM model via objectives (9)(10).

5: **return** $h, \mathbf{y}$ for QE inference data

Bi-LSTM Quality Estimation

To this end, we have the model derived and manual designed sequential features, each time stamp of which is corresponding to a fixed size vector. Our quality estimation task is built upon the bidirectional LSTM (Graves and Schmidhuber 2005) model, which has been widely used for sequence classification or sequence tagging problems. In sequence tagging, (Huang, Xu, and Yu 2015) proposed a variant of Bi-LSTM with one Conditional Random Field (CRF) layer (BI-LSTM-CRF). In our experiments, we found that the extra CRF layer did not show any significant improvement over vanilla Bi-LSTM, which we simply adopted. Another natural question is whether the traditional encoder self-attention or our proposed forward/backward self-attention can be an alternative to the Bi-LSTM. We empirically found the results with self-attention module become worse, and we attribute this to the scarcity of labelled quality estimation data, which is incomparable to the sufficient parallel corpus.

We concatenate the two sequential features along the depth direction to obtain a single one, denoted as $\{\mathbf{f}_k\}_{k=1}^T$, where T is the number of tokens in $\mathbf{m}$. Therefore, the sentence level score HTER prediction can be formulated as a regression problem (9), and the word error prediction is a sequence labeling problem (10),

$$\overrightarrow{\mathbf{h}_{1:T}}, \overleftarrow{\mathbf{h}_{1:T}} = \text{Bi-LSTM}(\{\mathbf{f}_k\}_{k=1}^T) \quad (8)$$

$$\arg\min \left\| h - \text{sigmoid}\left(\mathbf{w}^\top [\overrightarrow{\mathbf{h}_T}, \overleftarrow{\mathbf{h}_T}]\right) \right\|_2^2 \quad (9)$$

$$\arg\min \sum_{k=1}^{T} \text{XENT}(y_k, \mathbf{W}[\overrightarrow{\mathbf{h}_k}, \overleftarrow{\mathbf{h}_k}]) \quad (10)$$

Method	test 2017 en-de					test 2017 de-en				
	Pearson's $r\uparrow$	MAE $\downarrow$	RMSE $\downarrow$	**Spearman's** $\rho\uparrow$	DeltaAvg $\uparrow$	**Pearson's** $r\uparrow$	MAE $\downarrow$	RMSE $\downarrow$	**Spearman's** $\rho\uparrow$	DeltaAvg $\uparrow$
Baseline	0.397	0.136	0.175	0.425	0.0745	0.441	0.128	0.175	0.45	0.0681
Unbabel	0.641	0.128	0.169	0.652	0.1136	0.626	0.121	0.179	0.61	0.974
POSTECH Single	0.6599	0.1057	0.1450	0.6914	0.1188	0.6985	0.0952	0.1461	0.6408	0.1039
POSTECH Ensemble	0.6731	0.1067	0.1412	0.7029	0.1198	**0.7146**	0.0942	0.1359	0.6327	0.1044
Ours (MD+MM)	**0.6837**	0.1001	0.1441	**0.7091**	0.12	0.7099	0.0927	0.1394	**0.6424**	0.1018
– MM	0.6763	0.1015	0.1466	0.7009	0.1182	0.7063	0.0947	0.141	0.6212	0.1005
– MD	0.6408	0.1074	0.1478	0.663	0.1101	0.6726	0.1089	0.1545	0.6334	0.0961

Table 1: Results of sent level QE on WMT2017. MD: model derived features. MM: mis-matching features. Notice our ensemble results of Pearson's r for en-de and de-en are **0.7159** and **0.7338** respectively.

Method	test 2018 en-de SMT			test 2018 en-de NMT			test 2018 en-de SMT		
	F1-BAD	F1-OK	**F1-Multi**	F1-BAD	F1-OK	**F1-Multi**	F1-BAD	F1-OK	**F1-Multi**
Baseline	0.4115	0.8821	0.3630	0.1973	0.9184	0.1812	0.4850	0.9015	0.4373
Conv64	0.4768	0.8166	0.3894	0.3573	0.8520	0.3044	0.4948	0.8474	0.4193
SHEF-PT	0.5080	0.8460	0.4298	0.3353	0.8691	0.2914	0.4853	0.8741	0.4242
Ours	0.6616	0.9168	**0.6066**	0.4750	0.9152	**0.4347**	0.6475	0.9162	**0.5932**

Table 2: Results of word level QE on WMT2018. SMT or NMT means the translation outputs of QE data are from the unknown SMT or NMT system.

where $\mathbf{w}$ is a vector, $\mathbf{W}$ is a matrix, y_k is the error label for the k-th token of translation output, and XENT is the cross entropy loss (with logits). Notice HTER h is a real value within interval $[0, 1]$, we apply a squash function before the regression. Additionally, since the HTER is a global score for the entire sentence, we use the last hidden states of forward and backward LSTM as the regression signals. Besides, we can train the two losses together in a multi-task setting. In summary, we describe the outline of our proposed approach in Algorithm 1.

Experiments

In this section, we will demonstrate the performance of our approach with quantitative results on WMT 2017/2018 English↔German Translation Quality Estimation Tasks.

Setting Description

The English↔German parallel dataset we used is the same as the corpora for WMT 2017/2018 News/Biomedical Translation Track. To ensure the quality of the corpora, we filter the source and target sentence with length ≤70 and the length ratio between 1/3 to 3, thus resulting roughly 9 million (2017) and 25 million (2018) parallel sentences pairs for both English↔German direction. We tried word and byte-pair-encoding (BPE) (Sennrich, Haddow, and Birch 2016) tokenization, where the work tokenization fits the word level QE task. However, there exists the discrepancy between word level task and BPE tokenization, and we will present how to bridge the gap in the next section.

The number of layers in the bidirectional transformer for each module is 2, and the number of hidden units for feed-forward sub-layer is 512. Notice we prefer to use the 8-head self-attention in practice, since the single one is just a weighted average of previous layers. However, to mimic the convolution operation, where a different linear transformation for each relative position can allow the model to distinguish what information came from where, the multi-head attention layer allows each head uses different linear transformations, where in turn different heads can learn different relationships. The bilingual expert model is trained on 8 Nvidia P-100 GPUs for about 3 days until convergence. For translation QE model, we use only one layer Bi-LSTM, and it is trained on single GPU.

We evaluate our algorithm on the testing data of WMT 2017 and 2018. Notice for the QE task of WMT 2017, it is prohibitively to use any data from 2018, since the training data of 2018 includes some testing data of 2017. The same setting applies to the following experiments. We tuned all the hyper-parameters of our model on the development dataset to obtain the best single model, and reported the corresponding results for testing data.

Sentence Level Scoring

Method	**Pearson's** $r\uparrow$	MAE $\downarrow$	RMSE $\downarrow$	**Spearman's** $\rho\uparrow$
	test 2018 en-de			
Baseline	0.3653	0.1402	0.1772	0.3809
UNQE	0.7000	0.0962	0.1382	0.7244
Ours (MM + MD)	**0.7308**	0.0953	0.1383	**0.7470**
Method	test 2018 de-en			
Baseline	0.3323	0.1508	0.1928	0.3247
UNQE	**0.7667**	0.0945	0.1315	0.7261
Ours (MM + MD)	0.7631	0.0962	0.1328	**0.7318**

Table 3: Results of sent level QE on WMT2018

The sentence level results of WMT 2017 are listed in Table 1. We mainly compared our single model with the two algorithms (Kim et al. 2017; Martins, Kepler, and Monteiro 2017), ranking top 3 in the finalist. The primary metrics of sentence level are Pearson's correlation and Spearman's rank correlation of the entire testing dataset. Meanwhile, mean average error (MAE), root mean squared error

	F1-BAD	F1-OK	**F1-Multi**
Method	test 2017 en-de		
Baseline	0.407	0.886	0.361
DCU	0.614	0.910	0.559
Unbabel	0.625	0.906	0.566
POSTECH Ensemble	0.628	0.904	0.568
Ours (MM + MD)	0.641	0.9083	**0.5826**
Method	test 2017 de-en		
Baseline	0.365	0.939	0.342
POSTECH Single	0.552	0.936	0.516
Unbabel	0.562	0.941	0.529
POSTECH Ensemble	0.569	0.940	0.535
Ours (MM + MD)	0.5816	0.9470	**0.5507**

Table 4: Results of word level QE on WMT2017

Method	F1-BAD	F1-OK	**F1-Multi**
UAlacante SBI	0.1997	0.9444	0.1886
SHEF-bRNN	0.2710	0.9552	0.2589
SHEF-PT	0.2937	0.9618	0.2824
Ours	0.5109	0.9783	**0.4999**

Table 5: Result of gap level QE on WMT2018 En-De

(RMSE), and the average of delta values (DeltaAvg) can also measure the performance of overall predictions. For single model comparison, our algorithm can outperform the other two, and even be better than the ensemble model of best system for en-de testing data. In addition, we also analyze the importance of two proposed features by the ablation study. With 4-dimensional mis-matching (MM) features alone, the model can still achieve comparable or better performance than the second place system last year.

Word Level Tagging

The metric of word level is evaluated in terms of classification performance via the multiplication of F1-scores for the 'OK' and 'BAD' classes against the true labels. For the binary classification problem, we tuned the best threshold on the development dataset and applied to the test dataset. The overall results are shown at Table 2 and 4 . The baseline method is officially provided by WMT organizers, and the system is trained with CRFSuite toolkit with passive-aggressive algorithm (Okazaki 2007). We also compared the top 3 algorithms of WMT 2017 QE tasks, POSTECH (Kim et al. 2017), Unbabel (Martins, Kepler, and Monteiro 2017), and DCU (Martins et al. 2017). In the primary metric F1-Multi, our algorithm of the single model outperforms all other models, including the best ensemble system in WMT 2017. In WMT 2018 word level QE task, our approach exceeds the other algorithms with significant better numbers.

The higher value of single F1-OK or F1-BAD cannot reflect the robustness of the algorithm, since it may result in lower F1 of another metric. Though we presented the F1-OK and F1-BAD, it is not a valid official metric to QE task. However, by comparing them, we can conclude that all algorithms tend to classify the tag as OK in general, since the true labels are very imbalanced. This is the reason why we use the threshold tuning strategy to finalize our classifier.

Gap Level Tagging

Besides the sentence and word level quality estimation, the gap level error prediction is important to machine translation system as well. Missing tokens in the machine translations, as indicated by the TER tool are annotated as follows: after each token in the sentence and at the sentence start, a gap tag is placed. Note that number of gap tags for each target sentence is $T+1$, including the predictions before the first token and after last one. Theoretically, we can directly build the gap prediction model by modifying (10) as,

$$\arg\min \sum_{k=0}^{T} \mathrm{XENT}(g_k, \mathbf{W}[\overrightarrow{\mathbf{h}_k}, \overleftarrow{\mathbf{h}_k}, \overrightarrow{\mathbf{h}_{k+1}}, \overleftarrow{\mathbf{h}_{k+1}}]) \qquad (11)$$

where g_k is the gap tag between the kth and k+1st tokens. However, we can train the neural bilingual expert model with the expertise for gap prediction, to extract more representative features for the downstream task. Basically, we have the following factorization model $p(\mathbf{t}, \mathbf{t}^g|\mathbf{z}) = p(\mathbf{t}|\mathbf{z})p(\mathbf{t}^g|\mathbf{z})$ and $q(\mathbf{z}|\mathbf{s}, \mathbf{m})$, where $p(\mathbf{t}|\mathbf{z})$ is identical as previously discussed model, gap token prediction distribution $p(\mathbf{t}^g|\mathbf{z}) = \prod_k p(t_k^g|\overrightarrow{\mathbf{z}_k}, \overleftarrow{\mathbf{z}_k}, \overrightarrow{\mathbf{z}_{k+1}}, \overleftarrow{\mathbf{z}_{k+1}})$ and q becomes conditional on $\mathbf{m}$. Note that we need to define a "<blank>" token for gap prediction. Therefore, it also results in a side product – automatic post-editing.

As we discussed in the introduction, most computer assisted translation scenarios use the quality estimation model as the an activator of APE, a guidance to APE corrections, or a selector of final translation output (Chatterjee et al. 2018). Though QE can play the role of a helper function for APE, they are fundamentally considered as two separated tasks. In our proposed model, after we pre-trained the neural bilingual model for gap prediction, we can subsequently feed the model derived and mis-matching features to the Bi-LSTM model for gap quality estimation. We propose a direction to unify the quality estimation and automatic post-editing. First, we demonstrate the performance of our result for gap quality estimation in Table 5. As a side product, we will also show serval examples of APE result by our pre-trained model in Table 6.

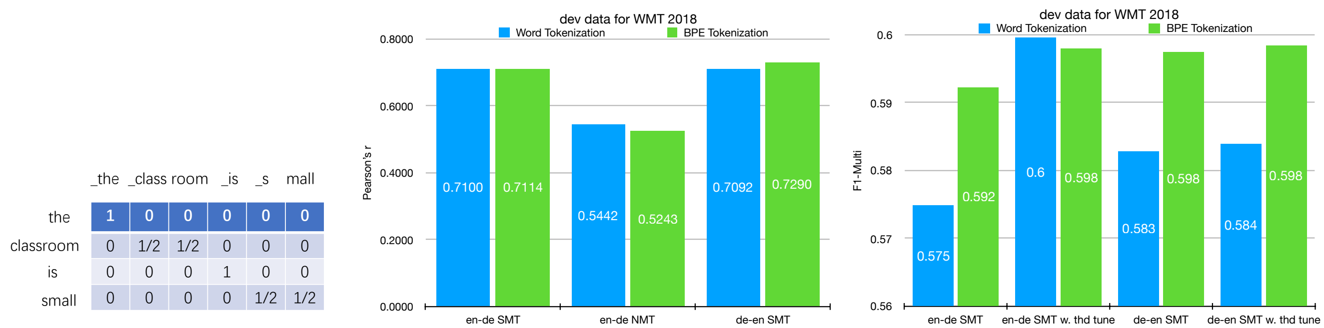
Extending to BPE Tokenization

In many NMT systems, using BPE or subword units gives an effective way to deal with rare words. Especially in German, there are a bunch of compound words, which are simply a combination of two or more words that function as a single unit of meaning, e.g. "handschuh" means glove in german, which is literally the "hand shoe". BPE tokenization gives a good balance between the flexibility of single characters and the efficiency of full words for decoding, and also sidesteps the need for special treatment of unknown words.

For sentence level HTER prediction, there is no harm or conflict to use BPE, since the regression signals only care about the hidden state of last time stamp. However, for word level labeling, the length of sequential features L_b with BPE tokenization is different from the number of word tokens L_w. We propose to average the features of all subword units belonging to one single word token, similar to average pooling along the time axis with dynamic sizes. To make the

MT	whlen sie im bedienfeld " profile " des dialogfelds " preflight " auf die schaltflche " lngsschnitte auswhlen . "
APE	klicken sie im bedienfeld " profile " des dialogfelds " preflight " auf die schaltflche " profile auswhlen . "
PE	klicken sie im bedienfeld " profile " des dialogfelds " preflight " auf die schaltflche " profile auswhlen . "
MT	das teilen von komplexen symbolen und groe textblcke kann viel zeit in anspruch nehmen .
APE	das trennen von komplexen symbolen und groen textblcke kann viel zeit in anspruch nehmen .
PE	das aufteilen von komplexen symbolen und groen textblcke kann viel zeit in anspruch nehmen .
MT	sie mssen nicht auf den ersten punkt , um das polygon zu schlieen .
APE	sie mssen nicht auf den ersten punkt klicken , um das polygon zu schlieen .
PE	sie mssen nicht auf den ersten punkt klicken , um das polygon zu schlieen .
MT	sie knnen bis zu vier zeichen .
APE	sie knnen bis zu vier zeichen eingeben .
PE	sie knnen bis zu vier zeichen eingeben .
MT	die standardmaeinheit in illustrator betrgt punkte (ein punkt entspricht .3528 millimeter) .
APE	die standardmaeinheit in illustrator ist punkt (ein punkt entspricht .3528 millimeter) .
PE	die standardmaeinheit in illustrator ist punkt (ein punkt entspricht .3528 millimetern) .

Table 6: Neural bilingual model with gap prediction expertise. In the shown examples, orange word means error translation, and yellow word means missing word. MT: machine translation; APE: automatic post-editing; PE: human post-editing.



(a) Segmentation matrix (b) Sent Level (c) Word Level

Figure 3: BPE experiments.

computational graph differentiable, the BPE segmentation information needs to be stored into a $L_w \times L_b$ sparse matrix S, where $S_{ij} \neq 0$ if j-th subword unit belongs to i-th word (see Fig 3(a) for an example). The averaged features can be computed by matrix multiplication.

We compared the performance of the word and BPE tokenization on both sentence and word levels, and results are plotted as histograms in Fig 3(b,c). Similar to NMT systems, the finer grained BPE tokenization can improve the QE performance in most tasks. In the sentence level, BPE model got a lower Pearson's r for en-de NMT dataset, which is very likely due to the small data size ($<$14000). In the word level, if we did not tune the threshold by using the default 0.5, the BPE model can always be better. After threshold tuning, the BPE model may have less improvement (we tune the threshold on development dataset and evaluate on it as well, since we did not have the ground truth of the testing data).

Conclusion and Future Work

In this paper, we present a novel approach to solve the quality estimation problem for machine translation systems. First, we introduce the "neural bilingual expert" model as the prior knowledge model. Then, we extract the model derived features and manually designed mis-matching features from the pre-trained "neural bilingual expert" model. In the end, we use a simple Bi-LSTM as the quality estimation model with the extracted features, and achieve the state-of-the-art performance of single model in WMT 2017/2018 campaigns.

Notice the "neural bilingual expert" model and the quality estimation model can be jointly trained during the stage of quality estimation, no matter the preprocessing is word or BPE tokenization. Even for BPE tokenization, we can do back-propagation to update "bilingual expert" model when we are training Bi-LSTM, if appropriate column and row paddings are added to the segmentation matrix. We leave this as the future work. Another possible research direction to explore automatic post-editing, by using the bilingual transformer with gap prediction. If we label the human post-edited translations by the insertion or deletion operations to machine translations (which could be done by using TER tool), we can train the model to predict such operations on the target side, achieving a better APE system eventually.

References

[Barrachina et al. 2009] Barrachina, S.; Bender, O.; Casacuberta, F.; Civera, J.; Cubel, E.; Khadivi, S.; Lagarda, A.; Ney, H.; Tomás, J.; Vidal, E.; et al. 2009. Statistical approaches to computer-assisted translation. *Computational Linguistics* 35(1):3–28.

[Bojar et al. 2017] Bojar, O.; Chatterjee, R.; Federmann, C.; Graham, Y.; Haddow, B.; Huang, S.; Huck, M.; Koehn, P.; Liu, Q.; Logacheva, V.; et al. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Proceedings of the Second Conference on Machine Translation*, 169–214.

[Chatterjee et al. 2018] Chatterjee, R.; Negri, M.; Turchi, M.; Frederic, B.; and Lucia, S. 2018. Combining quality estimation and automatic post-editing to enhance machine translation output. In *13th Conference of the Association for Machine Translation in the Americas (AMTA 2018)*, 26–38.

[Graves and Schmidhuber 2005] Graves, A., and Schmidhuber, J. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks* 18(5-6):602–610.

[Hassan et al. 2018] Hassan, H.; Aue, A.; Chen, C.; Chowdhary, V.; Clark, J.; Federmann, C.; Huang, X.; Junczys-Dowmunt, M.; Lewis, W.; Li, M.; et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

[Hill, Cho, and Korhonen 2016] Hill, F.; Cho, K.; and Korhonen, A. 2016. Learning distributed representations of sentences from unlabelled data. In *15th Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016*. Association for Computational Linguistics (ACL).

[Hokamp 2017] Hokamp, C. 2017. Ensembling factored neural machine translation models for automatic post-editing and quality estimation. In *Proceedings of the Second Conference on Machine Translation*, 647–654.

[Huang, Xu, and Yu 2015] Huang, Z.; Xu, W.; and Yu, K. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

[Kim et al. 2017] Kim, H.; Jung, H.-Y.; Kwon, H.; Lee, J.-H.; and Na, S.-H. 2017. Predictor-estimator: Neural quality estimation based on target word prediction for machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 17(1):3.

[Kingma and Welling 2013] Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[Martins et al. 2017] Martins, A. F.; Junczys-Dowmunt, M.; Kepler, F. N.; Astudillo, R.; Hokamp, C.; and Grundkiewicz, R. 2017. Pushing the limits of translation quality estimation. *Transactions of the Association for Computational Linguistics* 5:205–218.

[Martins, Kepler, and Monteiro 2017] Martins, A. F.; Kepler, F.; and Monteiro, J. 2017. Unbabel's participation in the wmt17 translation quality estimation shared task. In *Proceedings of the Second Conference on Machine Translation*, 569–574.

[Okazaki 2007] Okazaki, N. 2007. Crfsuite: a fast implementation of conditional random fields (crfs).

[Peters et al. 2018] Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

[Radford et al. 2018] Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training.

[Sennrich, Haddow, and Birch 2016] Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, 1715–1725.

[Shen et al. 2018] Shen, T.; Zhou, T.; Long, G.; Jiang, J.; and Zhang, C. 2018. Bi-directional block self-attention for fast and memory-efficient sequence modeling. *arXiv preprint arXiv:1804.00857*.

[Snover et al. 2006] Snover, M.; Dorr, B.; Schwartz, R.; Micciulla, L.; and Makhoul, J. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Biennial Conference of the Association for Machine Translation in the Americas, Cambridge, Massachusetts*.

[Specia, Paetzold, and Scarton 2015] Specia, L.; Paetzold, G.; and Scarton, C. 2015. Multi-level translation quality prediction with quest++. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, 115–120. Beijing, China: Association for Computational Linguistics and The Asian Federation of Natural Language Processing.

[Specia 2011] Specia, L. 2011. Exploiting objective annotations for measuring translation post-editing effort. In *Proceedings of the 15th Conference of the European Association for Machine Translation*, 73–80.

[Srivastava et al. 2014] Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929–1958.

[Tan et al. 2017] Tan, Y.; Chen, Z.; Huang, L.; Zhang, L.; Li, M.; and Wang, M. 2017. Neural post-editing based on quality estimation. In *Proceedings of the Second Conference on Machine Translation*, 655–660.

[Vaswani et al. 2017] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

[Wu et al. 2016] Wu, Y.; Schuster, M.; Chen, Z.; Le, Q. V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.