Levenshtein Training for Word-level Quality Estimation

Shuoyang Ding Marcin Junczys-Dowmunt
Matt Post Philipp Koehn

Task: Word-level QE

Given the **source sentence** and the **MT output**, generate a binary quality tag for each **MT output word** as well as each **gap between every two output words**.

source	Impfungen	beenden	die	Pandemie						
target	Vaccinations	␣	put	␣	ends	␣	to	␣	the	␣	pandemic
labels	OK	OK	OK	BAD	BAD	OK	OK	OK	OK	OK	OK

Motivation

MT → **Word-Level QE**? Seem intuitive, but not very practical. **Challenges**:

- Unable to use bi-directional context
- Unable to predict gap tags
- Subword vs. Word

Hence, we use **Levenshtein Transformer**, which has an iterative decoding procedure that is very similar to what we need for word-level QE task.

prev. target	Vaccinations	␣	put	␣	ends	␣	to	␣	the	␣	pandemic
deletion	NO		NO		YES		NO		NO		NO
mask ins.		0		2				0		0	
word ins.	Vaccinations		put	an end			to		the		pandemic

- deletion head → word label prediction
- mask insertion head → gap label prediction

Like all non-autoregressive models, **knowledge distillation (KD)** from an autoregressive system is necessary to achieve the optimal performance.

To take advantage of **large-scale pre-training**, we also experimented with initializing the encoder & decoder of the model with those from the **M2M-100-small** multilingual translation model.

Finetuning

There are some differences in terms of the translation task vs. word-level QE task for a Levenshtein Transformer decoder:

- **Input**: Noised Human Translation vs. Machine Translation
- **Output**: Subword-level Labels vs. Word-level Labels

Hence, **finetuning** is still necessary after training for translation. For finetuning, we will need **translation triplets**, i.e. the triplet of source, MT output and (pseudo/human) post-edits.

Synthetic Triplet Finetuning

- Data Construction
  - **src-mt-ref** • **bt-rt-tgt** • **src-mt1-mt2** • **mvppe**
- Multiview Pseudo Post-Editing (MVPPE): two **views** of a multilingual translation model by passing different input and language code: **translation view** $p_t(\texttt{pe} \mid \texttt{src})$ and **paraphrase view** $p_p(\texttt{pe} \mid \texttt{tgt})$.
  We approximate the post-edited output distribution $p(\texttt{pe} \mid \texttt{src}, \texttt{tgt})$ by **interpolating the two views**, i.e. $p = \lambda_t p_t + \lambda_p p_p$.
- Subword: Levenshtein Transformer is trained to generate subwords, hence the labels used for finetuning need to be on subword-level as well.
  Naïve: run TER on subword-level segmented text – will cause 10 % errors when coverted back to word-level
  Heuristic: run TER on both subword and word-level text, then heuristically (see our paper appendix) interpolate two versions of labels

Human Post-Edit Triplet Finetuning

Same as above, but on human post-edited translation triplets.

Beyond non-autoregressive translation, **Levenshtein Transformer** is also an effective model for **word-level quality estimation**.

Take a picture to view the paper, data and code

Experimental Setup

- Evaluation Data: WMT 2020 QE shared task, en-de & en-zh
- Evaluation Metric: Matthews Correlation Coefficient (**MCC**, higher is better), **F1 score** on `OK` & `BAD` tags

We have two separate training data configurations:

- Constrained Setting: use the same resource as the **OpenKiwi baseline** (a widely-adopted baseline implementation) built by the shared task organizer
- Unconstrained Setting: use all resource we can access. The **extra resources/procedures** we adopt for this setting are:
  - a better autoregressive en-de model for KD
  - extra training data for en-de
  - synthetic data finetuning
  - M2M-100 initialization

Results

	MCC	F1-OK	F1-BAD
en-de			
OpenKiwi	0.358	0.879	0.468
LevT w/o KD	0.441	0.926	0.498
LevT	**0.477**	**0.929**	**0.535**
en-zh			
OpenKiwi	0.509	0.849	0.658
LevT	**0.629**	**0.885**	**0.741**

Table 1: Results for Constrained Setting

Init.	LevT	Data Synth.	MCC	F1-OK	F1-BAD
en-de					
N	base	N	0.539	0.925	0.613
N	base	src-mt-ref	0.542	0.925	0.616
N	base	bt-rt-tgt	0.535	0.925	0.609
N	base	mvppe	0.548	0.926	0.620
N	base	src-mt1-mt2	0.549	0.926	0.622
N	big	N	0.551	0.927	0.623
N	big	src-mt1-mt2	0.562	**0.939**	0.617
M2M	418M	N	0.583	0.932	0.650
M2M	418M	src-mt1-mt2	**0.589**	0.934	**0.654**
en-zh					
N	base	N	0.629	0.885	0.741
N	big	N	0.625	0.885	0.738
M2M	418M	N	0.633	0.884	0.744
M2M	418M	mvppe	**0.646**	**0.892**	**0.752**
WMT20					
en-de best			0.597	0.935	0.662
en-zh best			0.610	0.887	0.723

Table 2: Results for Unconstrained Setting

Ablation Configuration	MCC	F1-OK	F1-BAD
best	0.589	0.934	0.654
-LevT	0.555	0.938	0.610
-LevT +lang-adapt	0.565	0.930	0.635
-LevT -synth.	0.321	0.915	0.380
-LevT -synth. +lang-adapt	0.451	0.946	0.498
-m2m	0.562	0.939	0.617
-m2m -KD	0.526	0.933	0.589
-m2m -heuristic tag	0.551	0.936	0.610
-m2m -synth.	0.551	0.927	0.623
-m2m -synth. -heuristic tag	0.539	0.925	0.613

Table 3: Ablation Study

JOHNS HOPKINS UNIVERSITY Microsoft