Supplementary Material for Long-Short Term Masking Transformer: A Simple but Effective Baseline for Document-level Neural Machine Translation

Pei Zhang, Boxing Chen, Niyu Ge, Kai Fan[*]
Alibaba Group Inc.
{xiaoyi.zp,boxing.cbx,niyu.ge,k.fan}@alibaba-inc.com

1 Evaluation Metrics of Consistency

BLEU is a commonly used metric to evaluate the precision-based quality of the translation in terms of n-gram, but it is not fit to evaluate discourse phenomena, because n-gram precision does not specifically reflect the cohesion and consistency in the long-range dependencies. **Deixis** addresses the error related to personal pronouns, specifically gender marks and informal/formal distinction. **Lexical cohesion** is refers to the consistency of a word or phrase when it occurs multiple times. **Ellipsis** is the omission of words that are understood from the context and it sometimes involves replacement of generic term for a specific term (such as 'did' for 'saw' in English). Since the target language is Russian, we care about both the verb and inflection.

2 Code in TensorFlow

We present the code snippet for generating local masking matrix for transformer encoder. The matrix for transformer decoder is simply add the above encoder matrix to the regular decoder self-attention masking matrix.

```
def generate_masking(inputs, sentence_sep_id):
    """Generate Long Short Term Masking
    Args:
        inputs: a dense vector [batch, length] of source or target word ids
        sentence_sep_id: the id of the sentence separation token
    """
    shape = tf.shape(inputs)
    length = shape[1]
    sentence_sep_id_matrix = sentence_sep_id * tf.ones(shape, dtype=inputs.dtype)
    sentence_end = tf.cast(tf.equal(inputs, sentence_sep_id), tf.float32)
    sentence_end_mask = tf.cumsum(sentence_end, axis = -1)
    sentence_end_mask_expand_row = tf.expand_dims(sentence_end_mask, -1)
    sentence_end_mask_expand_row = tf.tile(sentence_end_mask_expand_row, [1, 1, length])
    sentence_end_mask_expand_column = tf.expand_dims(sentence_end_mask, -2)
    sentence_end_mask_expand_column = tf.tile(sentence_end_mask_expand_column, [1, length, 1])
    mask = tf.cast(tf.equal(sentence_end_mask_expand_row, sentence_end_mask_expand_column), tf.float32)
    mask = -1e9 * (1.0 - mask)
    mask = tf.reshape(mask, [-1, 1, length, length])

    return mask
```

3 More Examples

We randomly selected three translation examples and illustrated in Table 1. For Example 1, the proposed system learnt "And" at the beginning of the translation, which is a side effect of document-level training. For Example 2, whether using "love" or "love to" is consistency in the proposed system and 1-to-1 baseline transformer. It seems that 1-to-1 baseline can approximately translate "极" to "radical", which does not even appear in the reference. I personally think "extremely" is a better translation. For Example 3, the reference seems not consistency in "how are we" and "how do we", but our proposed system prefers to keep in consistency using "how do we".

[*] corresponding author.

Src	养殖金枪鱼的饲料转换率是15比1。这个意思是说，每生产1磅金枪鱼肉耗费15磅用其他野生鱼类做的饲料。这可不是很具有可持续发展性。
Ref	It's got a feed conversion ratio of 15 to one. That means it takes fifteen pounds of wild fish to get you one pound of farm tuna. Not very sustainable.
Sys0	Feeding tuna is 15 to one. That means that every pound of tunas costs 15 pounds to feed feed on other wild fish. It's not sustainable.
Sys1	It's 15 to 1. What that means is that every pound-pound tuna produces 15 pounds of feed on every other wild fish. It's not sustainable.
Sys2	And the shift rate of breeding tuna is 15 to one. That means, for every one pound of tuna, it takes 15 pounds of feeding on other wild fish. It's not very sustainable.
Src	我们爱极了革新 我们爱技术，我们爱创造 我们爱娱乐
Ref	We love innovation. We love technology. We love creativity. We love entertainment.
Sys0	We love radical innovation. We love technology. We love creation. We love entertainment.
Sys1	We love to be innovative. We love technology. We love to create. We love entertainment.
Sys2	We love innovation. We love technology. We love creating. We love entertainment.
Src	想要喂饱这个世界？让我们开始问：我们怎么去喂养我们自己？或者更好的，我们怎么去建立一种环境它可以让每一个团体去养活自己？
Ref	Want to feed the world? Let's start by asking: how are we going to feed ourselves? Or better: how can we create conditions that enable every community to feed itself?
Sys0	Do you want to feed the world? So let's start asking: how do we feed ourselves? Or better, how can we build an environment that allows every group to feed themselves?
Sys1	How do we feed the world? So let's start asking: how do we feed ourselves? Or even better, how do we build an environment that will feed itself?
Sys2	Want to feed the world? Let's start asking: how do we feed ourselves? Or better, how do we build an environment that allows every single group to feed itself?

Table 1: Examples of translation results. Sys0: 1-to-1 transformer. Sys1: 3-to-3 transformer. Sys2: 3-to-3 long-short term masking transformer.