Supplementary Material: A High-Quality Multilingual Dataset for Structured Documentation Translation

Kazuma Hashimoto **Raffaella Buschiazzo** **James Bradbury***
Teresa Marshall **Richard Socher** **Caiming Xiong**
Salesforce
{k.hashimoto,rbuschiazzo,james.bradbury,
teresa.marshall,rsocher,cxiong}@salesforce.com

A Dataset Construction

A.1 XML Tag Categorization

The three manually-categorized XML tags are as follows:
– translatable {title, p, li, shortdesc, indexterm, note, section, entry, dt, dd, fn, cmd, xref, info, stepresult, stepxmp, example, context, term, choice, stentry, result, navtitle, linktext, postreq, prereq, cite, chentry, sli, choption, chdesc, choptionhd, chdeschd, sectiondiv, pd, pt, stepsection, index-see, conbody, fig, body, ul},
– transparent {ph, uicontrol, b, parmname, i, u, menucascade, image, userinput, codeph, systemoutput, filepath, varname, apiname},
– untranslatable {sup, codeblock}.
Among them, our pre-processed dataset has {ph, xref, uicontrol, b, codeph, parmname, i, title, menucascade, varname, userinput, filepath, term, systemoutput, cite, li, ul, p, note, indexterm, u, fn} embedded in the text as the actual XML tags.

A.2 URL Normalization

We have noticed that URLs are frequently mentioned in our dataset, and they are copied from one language to another. For simplicity, we replaced URL-like strings with placeholders. For example, the following sentence

> "http://aclweb.org/anthology/ has been moved to https://aclanthology.coli.uni-saarland.de/."

is changed to

> "#URL1# has been moved to #URL2#."

by keeping the correspondence between the same URLs in both sides of the paired languages. The evaluation is performed with the URL-anonymized form of the text.

*Now at Google Brain.

B XML-Constrained Beam Search

Algorithm 1 shows a comprehensive pseudo code of our XML-constrained beam search. T is the set of possible XML tag types, B is a beam size, and L is a maximum length of the generated sequences. Following Oda et al. (2017), we use a length penalty α. The proposed beam search ensures a valid XML structure conditioned by its source information, unless the generated sequence does not violate the maximum length constraint. It should be noted that this does not always lead to exactly the same structure as the structure of its reference text.

C Detailed Experimental Settings

This section describes more detailed experimental settings, corresponding to Section 4.

C.1 Tokenization by Sentencepiece

We used the SentencePiece toolkit to learn a joint sub-word tokenizer for each language pair, and we set the shared vocabulary size to 8,000 for all the experiments. In the experiments without the XML tags, the URL placeholders (#URL1#, #URL2#, ..., #URL9#) are registered as user-defined special symbols when training the tokenizers. For each of the English-to-{Japanese, Simplified Chinese} and Finnish-to-Japanese experiments, we over-sampled English or Finnish text for training the joint sub-word tokenizer, because Japanese and Simplified Chinese have much more unique characters than other alphabetic languages.

In the experiments with XML, we further added all the XML tags (e.g. ,) to the list of the user-defined special symbols. We also set the three tokens &, <, and > as the special tokens. When computing BLEU scores, &, <, and > are replaced with &, <, and >, respectively.

C.2 Model Training

We implemented the transformer model with $K = 6$ and $d = 256$ as a competitive baseline model. The number of the multi-head attention layer in the transformer model is 8, and the dimensionality of its internal hidden states is 1024. For more details about the multi-head attention layer and the internal hidden states, please refer to Vaswani et al. (2017).

For optimization, we used *Adam* (Kingma and Ba, 2015) with a modified weight decay and a cosine learning rate annealing (Loshchilov and Hutter, 2017). The mini-batch size was set to 128, and the weight decay coefficient was set to 1.0×10^{-4}. A gradient-norm clipping method was used to stabilize the model training, with the clipping size of 1.0. The initial learning rate is 5.0×10^{-4}, and it is linearly increased to 1.0×10^{-3} according to the number of iterations in the first 10 epochs of the model training. Then, the learning rate and the weight decay coefficient are multiplied by the following annealing factor:

$$\eta_i = 0.5 + 0.5 \cos\left(\frac{i - 10}{50 - 10}\pi\right), \qquad (1)$$

where η_i is for the i-th ($i \geq 10$) epoch of the model training, and "50" is the maximum number of the training epochs. During the model training, a greedy-generation BLEU score without XML is evaluated at every half epoch by using the development set, and the best-performing checkpoint is used for evaluation.

Algorithm 1 XML-constrained beam search

function ConstrainedBeamSearch(x, T, B, L, α)
  $C = []$ ▷ Candidates in the beam search
  for i in $1 \ldots B$ **do**
    $y = [\texttt{BOS}]$ ▷ Output token sequence
    $s = 0.0$ ▷ Score
    $t = [t_1, \ldots, t_T]$ ▷ Possible XML tag types in x
    $t' = []$ ▷ History of opened tags
    C.append($\{y, s, t, t'\}$)
  end for

  while max length $< L$ and $C[0].y[-1]$ is not EOS **do**
    for i in $1 \ldots B$ **do**
      if $C[i].y[-1]$ is EOS **then**
        $\ell_i = C[i].s$
      else
        $\ell_i = \log p(w|x, C[i].y) \in \mathbb{R}^{|\mathbb{V}|}$
        $\ell_i \mathrel{+}= C[i].s + \alpha$
        for τ in T **do**
          if τ is not in $C[i].t$ **then**
            $\ell_i(w : <\tau>) = -\inf$
          end if

          if $C[i].t'$ is [] or $\tau \neq C[i].t'[-1]$ **then**
            $\ell_i(w : </\tau>) = -\inf$
          end if
        end for

        if $C[i].t$ is not [] or $C[i].t'$ is not [] **then**
          $\ell_i(w : \texttt{EOS}) = -\inf$
        end if
      end if
    end for

    $C' = []$ ▷ Updated candidates
    for i in $1 \ldots B$ **do**
      $w_i, j_i = \arg\max_{w,j}(\ell_1, \ldots, \ell_j, \ldots, \ell_B)$
      if $C[j_i].y[-1]$ is EOS **then**
        C'.append($C[j_i]$)
        $\ell_{j_i} = 0$
        continue
      end if

      $y = C[j_i].y + [w_i]$
      $s = \ell_{j_i}(w : w_i)$
      C'.append($\{y, s, C[j_i].t, C[j_i].t'\}$)

      if w_i is an XML open tag **then**
        $C'[-1].t$.remove(type of w_i)
        $C'[-1].t'$.append(type of w_i)
      end if

      if w_i is an XML close tag **then**
        $C'[-1].t'$.pop()
      end if

      if the first token **then**
        $\ell_{\text{all}}(w : w_i) = -\inf$
      else
        $\ell_{j_i}(w : w_i) = -\inf$
      end if
    end for
    $C = C'$
  end while

  return $C[0].y$
end function

References

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. 2017. Fixing Weight Decay Regularization in Adam. *arXiv preprint arXiv:1711.05101*.

Yusuke Oda, Katsuhito Sudoh, Satoshi Nakamura, Masao Utiyama, and Eiichiro Sumita. 2017. A Simple and Strong Baseline: NAIST-NICT Neural Machine Translation System for WAT2017 English-Japanese Translation Task. In *Proceedings of the 4th Workshop on Asian Translation*, pages 135–139.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.