Abstract

We consider the scenario of partially ranked crowd translations, where workers collaboratively post and peer edit prior translations. The iterative nature of the contributions, leads to interdependencies in the quality. We pose the problem as pair-wise comparisons of translations considering their history and model it using Siamese LSTM architecture. The LSTMs model translation dependencies and Siamese network model the preference function. We consider a supervised setting and predict the pair-wise comparisons for non-ranked translations. A sorting algorithm is used to get the complete set of rankings. The sequential nature of the problem is modeled well by LSTMs yielding 88.83% accuracy and 0.95 rank correlation, against a non-sequential Siamese DNN network providing an accuracy of 74.35% and 0.85 rank correlation, thus establishing the efficacy of the proposed approach.

Introduction

Post editing is an attractive alternative to completely manual translation offering time and cost efficiency, provided the quality of the translations is ensured (Zbib et al. 2013; Läubli et al. 2013; Callison-Burch 2009; Goto, Lin, and Ishida 2014; Potepa et al. 2011). Crowd ranking can serve the purpose (Callison-Burch 2009; Goto, Lin, and Ishida 2014; Bentivogli et al. 2011; Paul et al. 2012); but the quality of ranks still remain questionable. (Kilian et al. 2012; Zaidan and Callison-Burch 2011) use expert’s rank as one of the features to train classifiers for quality prediction, where all contributions are independent. Such independence cannot be guaranteed when the workers post-edit machine or peer-edit prior human contributions, as the quality of a translation is not only a function of its quality parameters; but also depends on the quality of prior refereed translations.

We consider a workflow (Figure 1), where a crowd worker inspects all the machine and prior crowd translations for a sentence; rates the prior contributions on a scale of 1 (worst) to 5 (best); selects one and submits a post-edited version, forming a trail of translation history. We enable an expert to rank all the available translations for a subset of source sentences. Lower rank denotes higher quality. All ranked contributions of a source sentence are compared with each other to generate pairs. If the rank of the first translation in a pair is less than the second; the pair is labeled as ‘1’, otherwise as ‘0’. We compare the paired translations along with their edit-history and train a Siamese recurrent neural architecture to identify better quality translation in a pair. Such networks are earlier used by (Mueller and Thyagarajan 2016) for identifying semantic similarity between paired sentences. They are designed to implement the symmetries naturally present in a preference function. We use this trained network to compare the remaining non-ranked translation sequences of a sentence to get complete set of paired comparisons and further solve the rank completion problem.

Prior approaches (Rigutini et al. 2011; Fürnkranz and Hüllermeier 2010) learn preference function using shared weight approach to compare an paired objects, by providing their feature representations. However, to the best of our knowledge, ours is the first attempt to apply the shared weight architectures for translation ranking problem. Moreover, considering the sequential nature of the post-edit problem, we use Siamese adaption of LSTMs, for pairwise translation comparisons along with their post-edit history. We benchmark against Siamese DNN which compares translations without consideration of the history.

Methodology

Let \( t_{ij} \) be a post-edit translation submitted for the \( i^{th} \) sentence in \( j^{th} \) post-edit round. The index of round, \( j \), is utilized as the time index for the LSTMs. \( x_{ij} \) be the input vector for the \( j^{th} \) post-edit translation of the \( i^{th} \) sentence. \( y_{ijk} \),
The last hidden states $h_{t}$ is the label for ordered translation pair $t_{ij}$ and $t_{ik}$, where $j \neq k$, $y_{ijk} \in \{0, 1\}$, 1 indicating the $1^{st}$ translation $t_{ij}$ is better than the $2^{nd}$ translation $t_{ik}$ and 0 otherwise. The problem we are trying to solve is: Given an ordered pair of translation along with their post-edit sequences for a source sentence, determine the better quality translation. There are two networks LSTM$_a$ and LSTM$_b$ (Figure 2). Each process one of the translation post-edit sequence in a given pair. We model siamese architectures with tied weights such that LSTM$_a$ = LSTM$_b$. A LSTM learns the mapping from the space of variable length post-edit sequences of n-dimensional translation feature vectors $x_{i1}, x_{i2}, \ldots, x_{ij-1}, x_{ij}$ into an an embedding $h_{ij}$, provided by the last hidden state. Following equations demonstrate the mapping defined by the LSTM model. The equations are valid for all the sentences $i$ and both LSTM$_a$ and LSTM$_b$.

$$I_j = \tanh(W_{x}x_{j} + W_{h}h_{j-1} + b_{I})$$  \hspace{1cm} (1)

$$c_j = \sigma(W_{x}x_{j} + W_{h}h_{j-1} + b_{c})$$  \hspace{1cm} (2)

$$F_j = \sigma(W_{x}x_{j} + W_{h}h_{j-1} + b_{F})$$  \hspace{1cm} (3)

$$O_j = \tanh(W_{x}x_{j} + W_{h}h_{j-1} + b_{O})$$  \hspace{1cm} (4)

$$c_j = c_{j-1} \odot F_j + I_j \odot c_{j-1}$$ \hspace{1cm} (5)

$$h_{j} = O_j \odot \tanh c_{j}$$ \hspace{1cm} (6)

The last hidden states $h_{ij}$ and $h_{ik}$ of the LSTM$_a$ and LSTM$_b$ respectively are concatenated and provided as an input to one dense sigmoid layer to map it to the output $y_{ijk}$.

$$y_{ijk} = \sigma(W_{yh}[h_{ij}, h_{ik}])$$ \hspace{1cm} (7)

The siamese LSTM network solves the problem by computing the conditional probability $P[y_{ijk} | x_{i1}, x_{i2}, \ldots, x_{ij-1}, x_{ij}, x_{i1}, x_{i2}, \ldots, x_{ik-1}, x_{ik}] = P[y_{ijk} | h_{ij}, h_{ik}]$. Where $[.]$ denotes a post-edit sequence. Translation input feature vector $x_{ij}$ consists of: (i) Edit distance of the translation with the parent translation selected for post-edit. (ii) Word length of the source sentence (iii) Time taken in seconds by a worker to perform the task (iv) Average peer rating of a translation (v) Worker one-hot vector encoding.

For validating the results we split our ranked translation data (i) Sentence-wise (ii) Translation-wise and (iii) Pairwise. For (ii) and (iii) the predicted labels along with the training labels provide us with a complete set of comparisons for a sentence. We rank all the translations of a sentence by sorting them based on the number of wins. In case of a tie, a translation submitted in the later round is considered to be better quality. Spearman’s coefficient is used to determine rank correlation between actual and predicted rankings.

### Data and Experimental Results

60 non-professional crowd workers provided us 6016 Hindi post-edit translations for 1758 English sentences from a book about an autobiography. The maximum rounds of post-edits for any sentence were limited to 4. Thus, along with 3338 machine translations, we have total of 9354 translations. Translations of 474 sentences (2693 translations) were ranked between 1 to 6 by an expert worker (in-house linguist). We formed total of 12,824 ordered pairs of post-edit translation sequences and the corresponding labels by following the process discussed in the introduction section.

We modeled siamese LSTM, where each LSTMs have 10 dimensional hidden layer, time step of 6 and with 67 dimensional feature vector. Table 1 represents the validation accuracies and rank correlations for distinct splits. Siamese LSTMs results are better as compared to the results of non-recurrent Siamese DNNs and approximately 8-10% better than purely deterministic rank completion algorithms.

### Conclusion

Siamese LSTMs are able to effectively model the pairwise comparisons for translations along with capturing post-editing interdependencies. In future, we plan to apply the technique for other crowd annotation NLP tasks with free-flowing labels, such as text simplification (Coster and Kauchak 2011), content moderation (Bernstein et al. 2015), image or speech transcription (Callison-Burch and Dredze 2010), question-answering (Bian et al. 2008), etc.

### Table 1: Results

<table>
<thead>
<tr>
<th>% Split</th>
<th>Validation Accuracy</th>
<th>Rank Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Siamese</td>
<td>DNN</td>
<td>LSTM</td>
</tr>
<tr>
<td>Sentence-Wise Random Split</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90-10</td>
<td>73.67</td>
<td>74.32</td>
</tr>
<tr>
<td>Translation-Wise Split</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(n-1)-1*</td>
<td>68.88</td>
<td>71.05</td>
</tr>
<tr>
<td>Pair-Wise Random Splits</td>
<td></td>
<td></td>
</tr>
<tr>
<td>90-10</td>
<td>74.05</td>
<td>92.67</td>
</tr>
<tr>
<td>80-20</td>
<td>74.04</td>
<td>90.84</td>
</tr>
<tr>
<td>70-30</td>
<td>74.09</td>
<td>89.94</td>
</tr>
<tr>
<td>60-40</td>
<td>74.35</td>
<td>88.83</td>
</tr>
<tr>
<td>50-50</td>
<td>73.39</td>
<td>85.64</td>
</tr>
<tr>
<td>40-60</td>
<td>73.00</td>
<td>82.95</td>
</tr>
<tr>
<td>30-70</td>
<td>72.79</td>
<td>70.75</td>
</tr>
</tbody>
</table>

*1** n-1 sentence translations for training, last for testing
References


