Word Level Translation Quality Estimation

Shravan Nayak
Aryansh Omray

Problem

- To predict which words in the machine translated sentence are correct and which are wrong.
- Both syntactic information and semantic information have to be considered before making predictions which makes the task difficult.
- More formally the task assigning one of the two label (**BAD**, **OK**) to every word in the machine translated sentence.

Related work

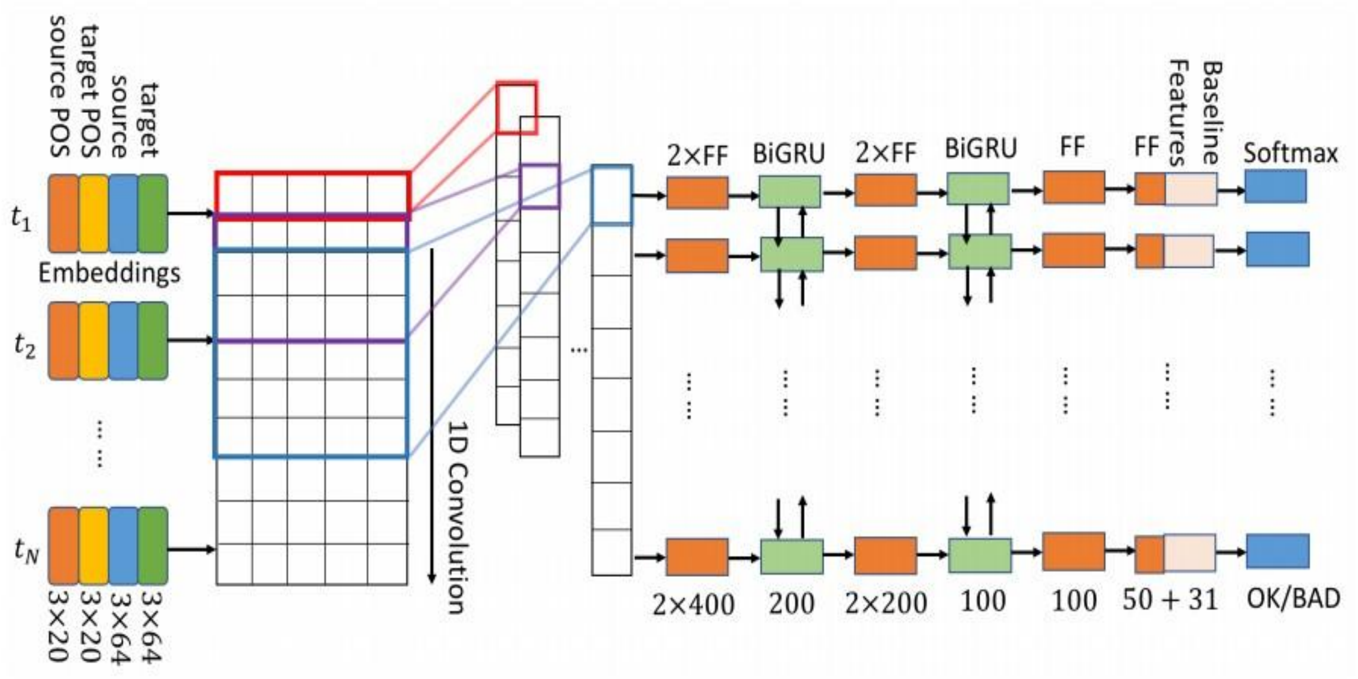
- Early work on the problem used hand crafted features with simple regression and classification models (*Ueffing and Ney, 2007; Bicici, 2013*)
- More recent works have used recurrent neural network to encode target and source words along with their contexts *(Martins et al., 2017).*
- Large performance gains can be obtained using deep neural networks (RNN)

Data

- The data set consists of source sentences along with their machine translated sentences and gold labels for each word.
- Moreover word alignments between each word in the source and the target are provided.
- Further we also have access to 31 baseline features for each sentence which we use in our models.

Model

- We base our model on the model by *Junjie Hu et.al* which achieved top scores in WMT 2018 word level quality estimation task.
- We use a three part neural network to encode global and local contextual information.
- The first layer uses an embedding layer to represent words and their parts of speech in both source and target language.
- The second layer used 1D CNN to integrate local context information.
- The third layer consists of stacked bi-GRU/bi-LSTM(in different experiments) layers to encode global context before making predictions.



target
source
target POS
source POS
t_1
Embeddings
t_2
t_N
3×64
3×64
3×20
3×20
1D Convolution
2×FF
BiGRU
2×FF
BiGRU
FF
FF
Baseline Features
Softmax
2×400
200
2×200
100
100
50 + 31
OK/BAD

Model: Embedding Layer

- Used to create a vector representation (R^D) of each word.
- Embedding of the word in target language is concatenated with the embeddings of the corresponding words in the source language.
- In order to provide local context the embeddings of the neighbouring words are also concatenated.
- Thus for each word we obtain a vector representation of dimension (R^{6*D})

Model: Convolutional Layer

- Provide a powerful way to extract local context features and create rich representations.
- Embeddings of all the words are concatenated column wise and 1D convolution is applied along the sequence.
- Embedding of different window sizes (1,3,5) are used to capture various granularities and all the embeddings are concatenated.

Model: RNN Layer

- A stack of RNN layer followed by linear layer is used to get the final representation of each word.
- This helps to encode global context in the representation of each word.
- Softmax is applied on the final representation of each word to get the prediction.

Training

- We use binary cross entropy loss as the loss function to train our model.
- We use an Adam optimizer with initial learning rate of 0.001 which decays linearly with every epoch.
- Dropout of 0.3 is used for RNN and the feed forward networks.

Results

The results of English to German are as follows:

Model	Accuracy	F-1 Score (Macro)	F-1 Score (Micro)	F-1 score (weighted)
GRU	82.68	55.38	82.68	80.49
LSTM	81.27	56.12	81.27	80.02

- The LSTM model overfits more than the GRU, hence the decrease in scores.
- The large difference between F1 macro and F1 micro is due to label imbalance.

Future Work

- The model overfits very easily due to the large imbalance. We plan to use regularization techniques and data upsampling to tackle this issue.
- Experiment more with using pre trained embeddings (BERT, glove, BPE) rather than the embedding layer.

THANK YOU