-: Project Report :-
-: Quality Estimation of Machine Translated Text :-

Professor: Vasudeva Verma
Mentor: Nisarg Jhaveri

Neeraj Battan
Harsha Vardhan
Kartavya Gupta

Problem Statement

Applications

Challenges

Baseline Features

Approaches
- Word Vectors Based Approach
  - Motivation
  - Workflow
  - Procedure
  - Tools
  - Results
- A Recurrent Neural Networks Approach for Estimating the Quality of Machine Translation Output
  - Workflow for the approach
  - Motivation
  - Description of the Neural network
  - How to get the quality vectors?
  - How to get the Quality estimation score using the Quality Vectors?
  - Model for the Neural Network
  - Tools
  - Results
  - Possible Improvements

Problem Statement

The aim of our project is to determine the quality of a machine translated text by predicting the HTER(Human-targeted Translation Error Rate) scores. We are given 23000 English sentences, their machine translated text to German, their human translated text for German and score for machine translation. Similarly we are provided with 25000 German sentences with same features as of English sentences. We need to build a system which will determine the quality of the machine translated text without any human feedback.

Applications

Some of the applications of Machine translated quality estimation are:

- Decide whether a given translation is good enough of publishing as it is
- Inform readers of the target language only whether or not they can rely on a translation
- Filter out the sentences which are not good enough for posting and need post-editing by human
- Select the best translation among options from multiple Machine Translation and/or translation memory systems
- Highlight the words that need post-editing task
- Inform readers the portion of the sentence that are not reliable

Challenges

Challenges related to (A Recurrent Neural Networks Approach for Estimating the Quality of Machine Translation Output):

- We don’t have enough parallel corpus to train the sequence to sequence model
- Modifying the sigmoid function of the decoder to generate quality estimation vectors
- Time for training the network

Challenges related to (Improving Machine Translation Quality Estimation with Neural Networks Features):

- Choosing the dimension of the word vectors
- Dealing with words which are out of vocabulary
- Choosing the right regression model

Baseline Features

- number of tokens in the source sentence
- number of tokens in the target sentence
- average source token length
- LM probability of source sentence
- LM probability of target sentence
- number of occurrences of the target word within the target hypothesis (averaged for all words in the hypothesis - type/token ratio)
- average number of translations per source word in the sentence (as given by IBM 1 table thresholded such that prob(t|s) > 0.2)
- average number of translations per source word in the sentence (as given by IBM 1 table thresholded such that prob(t|s) > 0.01) weighted by the inverse frequency of each word in the source corpus

- percentage of unigrams in quartile 1 of frequency (lower frequency words) in a corpus of the source language (SMT training corpus)
- percentage of unigrams in quartile 4 of frequency (higher frequency words) in a corpus of the source language
- percentage of bigrams in quartile 1 of frequency of source words in a corpus of the source language
- percentage of bigrams in quartile 4 of frequency of source words in a corpus of the source language
- percentage of trigrams in quartile 1 of frequency of source words in a corpus of the source language
- percentage of trigrams in quartile 4 of frequency of source words in a corpus of the source language
- percentage of unigrams in the source sentence seen in a corpus (SMT training corpus)
- number of punctuation marks in the source sentence
- number of punctuation marks in the target sentence

Approaches

We are using two approaches:

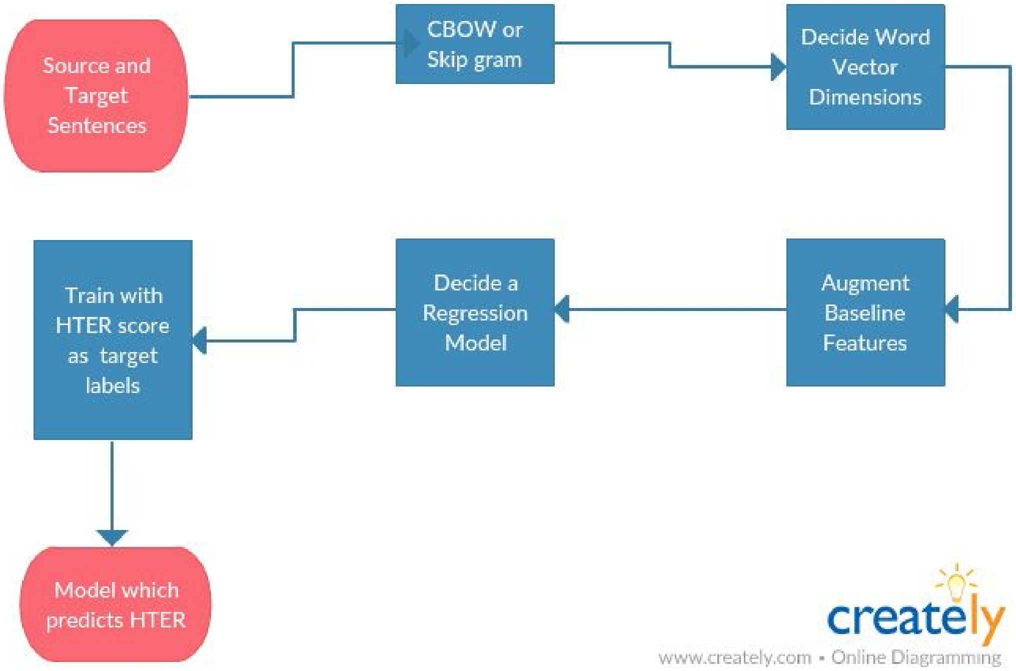
Word Vectors Based Approach

In this approach, neural network features are utilized, including the word embedding features of both the source and the target sentences along with baseline features that are provided. In the data both English to Spanish machine translated and Spanish to English machine translated sentences are used.

Motivation

Traditional methods for quality estimation use linguistic features such as part-of-speech analysis, syntactic analysis, sentiment analysis etc. These approaches are constrained by the availability of linguistic resources and statistical tools available in those particular languages. Therefore here only neural network features are used. The word2vec model is used in order to capture the context between words.

Workflow


Source and Target Sentences
CBOW or Skip gram
Decide Word Vector Dimensions
Augment Baseline Features
Decide a Regression Model
Train with HTER score as target labels
Model which predicts HTER
creately
www.creately.com • Online Diagramming

Procedure

The Embedding Features are first generated from the source and target languages using the CBOW method with the following parameters

- Size - optimised for en-de and de-en. Dimensions from 256 to 2048 were tested
    - En - De 1024 for the source language sentences and 2048 for the target language sentences gave the best results.
    - De - EN 2048 for the source language sentences and 2048 for the target language sentences gave the best results.
- Sampling threshold of high frequency word - 1e-5
- Negative samples - 10

The Continuous Bag of Words Approach was used as it was faster in generating the word embeddings when compared to the skip gram model.

The obtained word embeddings were then augmented to the baseline features that were provided along with the shared task .

These features were then trained in a regression model.

The regression models that were tried were SVR (Support Vector Regression), RFR (Random Forest Regression), an ensemble of Decision Tree Regression.

Although the approach in the paper was to use a SVR method, it is extremely slow when used on a big dataset like the one provided. So the Random Forest Regression was used which gave comparable results but is way faster than a Support Vector Regressor.

Tools

- Word2Vec for training word embedings
- Sklearn for regression models

Results

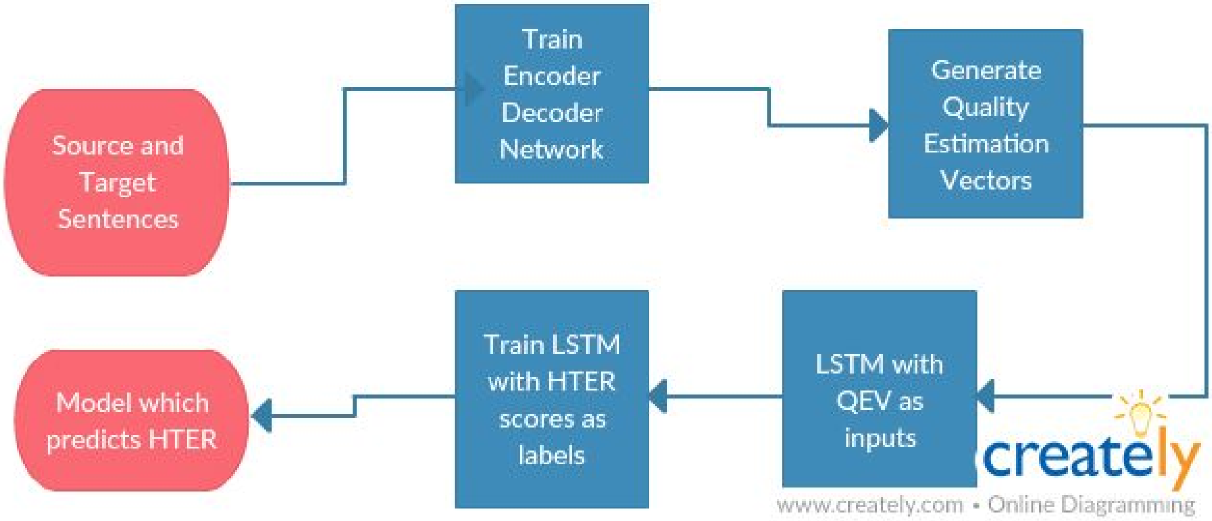
- For German-English: 0.45
- For English-German: 0.43

Possible Improvements

- Using neural networks based regression to improve the results

Recurrent Neural Networks Approach

Workflow for the approach


Source and Target Sentences
Train Encoder Decoder Network
Generate Quality Estimation Vectors
LSTM with QEV as inputs
Train LSTM with HTER scores as labels
Model which predicts HTER
creately
www.creately.com • Online Diagramming

This approach is comparable to currently state of the art approach. They are using recurrent neural networks for estimating the quality of machine translation output. A sequence of vectors made by the prediction method is used as the input of the final recurrent neural network. The prediction method uses bidirectional recurrent neural network architecture both on source and target sentence to fully utilize the bidirectional quality information from source and target sentence. Their experiments show that the proposed recurrent neural networks approach achieves a performance comparable to the existence state of the art models for estimating the sentence level quality of English-to-Spanish translation. We are using their model for estimating the quality of the German-English Translated and English-German translated text.

Motivation

The recurrent neural network possesses sequentiality and memorability, and it performs well in sequential data modeling. Therefore, the Recurrent Neural Network Language Model (RNNLM) (Mikolov et al. 2010) was proposed and first used in reordering of machine translations. The experimental results in- dicate that the RNNLM is superior to the back-off language model Since RNNLM accounts for the word order, they will capture more information than just using the word embeddings like we did in the previous method .

Description of the Neural network

The input of the final RNN is a sequence of vectors that have quality information about whether target words in a target words in sentence are properly translated from a source sentence. We will refer to this sequence of vectors as quality vectors ($q_{y1,}$... q_{yt}). Each quality vector has the quality information about how well a target word y_j in a target sentence y = (y_1...y_t) is translated from a source sentence x = (x_1...x_t). Quality vectors are generated from the prediction method.

How to get the quality vectors?

The training data for Quality Estimation is not enough to use a neural networks approach for making quality vectors, so they are using an alternative based on large-scale parallel corpora such as Europarl. They modify word prediction method of RNN ENcoder-Decoder using parallel corpora to make the quality vectors.

RNN Encoder-Decoder proposed by Cho et al. is able to predict the target word y_j given a source sentence x and all preceding target words by using a softmax function. And it is extended by Bahdanau et al. to use information of relevant source words for predicting the target word y_j such that

$$p(y_j / \{y_1,....,y_{j-1}\}, x) = g(y_{j-1},s_{j-1},c_j)$$

g is a nonlinear function predicting the probability of y_j. S_{j-1} is the hidden state of the forward RNN on target sentence and contains information of preceding target words $\{y_1,...y_{j-1}\}$. C_j is the context vector which means relevant parts of source sentence associated with the target word y_j. S_{j-1} and y_{j-1} are related to all preceding target words $\{y_1,...y_{j-1}\}$, and c_j is related to x in the word prediction function of the above equation.

In their proposed QE model, bidirectional RNN architecture is used both on source and target sentence. By applying bidirectional RNN architecture both on source and target sentence, we can fully and bidirectionally utilize source and target sentence for predicting target words, such that

$$p(y_j \,|\mathbf{y}_{!}\ni = y_j , \mathbf{x}) = g([y_{j-1} ; y_{j+1}], [s_{j-1} ; s_{j+1}], c_j)$$

To make quality vectors, we regard that the probability of the target word y_j involves the quality information about whether the target word y_j in target sentence is properly translated from source sentence. Thus, by decomposing the softman function of the above equation. The quality vector q_{yj} for the target word y_j is computed by

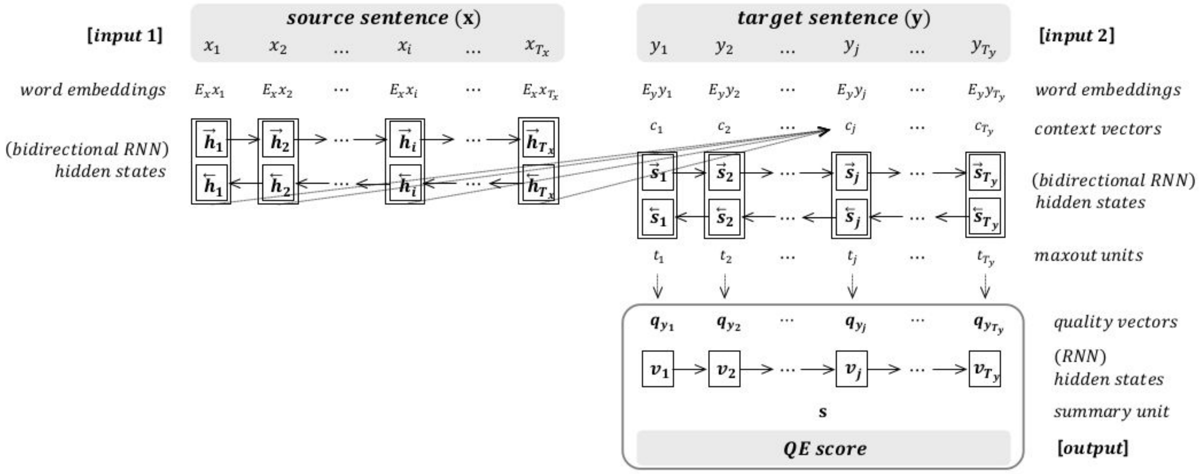
$$q_{yj} = [row_{yj}(W_{o1}) \,.\, [W_{o2}t_j]^T]^T$$

Where . is an element-wise multiplication. All of quality information about possible K_y target words at position j of target sentence is encoded in t_j. Thus, by decoding t_j, we are able to get quality vector q_{yj} for the target word y_j.

How to get the Quality estimation score using the Quality Vectors?

To predict a Quality Estimation score as an HTER score in [0,1] for each target sentence, we are using a LSTM model instead of a logistic sigmoid function used by the authors. The input of the LSTM model is concatenated quality vectors. The number of quality vectors differs for every sentence. We are taking upto 10 quality vectors and neglecting the next ones. If the number of quality vectors are less than 10 then we are padding them by zeros. The size of every quality vector is 620. So the size of the input for every sentence for LSTM is 6200. We are using a fully connected layer after the first input layer with 128 outputs and tanh as activation function. One more hidden layer is used after that with input dimension as 128 and output as 1, with sigmoid as activation function. So our output ranges b/w (0,1) which is the range of our quality estimated score. We are training this neural network and using this for predicting the final quality estimation score.

How to get the Quality estimation score using the Quality Vectors?



[input 1]
source sentence (x)
x_1 x_2 … x_i … x_{T_x}
word embeddings
$E_x x_1$ $E_x x_2$ … $E_x x_i$ … $E_x x_{T_x}$
(bidirectional RNN) hidden states
$\vec{h}_1$ $\vec{h}_2$ $\vec{h}_i$ $\vec{h}_{T_x}$
$\overleftarrow{h}_1$ $\overleftarrow{h}_2$ $\overleftarrow{h}_i$ $\overleftarrow{h}_{T_x}$
target sentence (y)
[input 2]
y_1 y_2 … y_j … y_{T_y}
word embeddings
$E_y y_1$ $E_y y_2$ … $E_y y_j$ … $E_y y_{T_y}$
context vectors
c_1 c_2 … c_j … c_{T_y}
(bidirectional RNN) hidden states
$\vec{s}_1$ $\vec{s}_2$ $\vec{s}_j$ $\vec{s}_{T_y}$
$\overleftarrow{s}_1$ $\overleftarrow{s}_2$ $\overleftarrow{s}_j$ $\overleftarrow{s}_{T_y}$
maxout units
t_1 t_2 … t_j … t_{T_y}
quality vectors
q_{y_1} q_{y_2} … q_{y_j} … $q_{y_{T_y}}$
(RNN) hidden states
v_1 v_2 … v_j … v_{T_y}
summary unit
s
QE score
[output]

Figure 1: An illustration of the proposed recurrent neural networks model for quality estimation

Tools

- Tensorflow

Results

- For German-English: 0.63
- For English-German: 0.55

Possible Improvements

- Train the RNN model with a large parallel corpus giving more accurate quality estimation vectors which will give us a huge boost
- Incorporating more features along with the generated QEV

External Links

https://talent404.github.io/IRE-MTQE/index.html
https://github.com/talent404/IRE-MTQE

References

http://www.statmt.org/wmt17/quality-estimation-task.html

http://www.statmt.org/wmt17/pdf/WMT61.pdf

http://www.aclweb.org/anthology/N16-1059