ROSETTA

Ludovico Mitchener
lm2917@ic.ac.uk

Sacha Hu
sh2719@ic.ac.uk

Alvaro Prat Balasch
ap5915@ic.ac.uk

Abstract

The importance of robust quality estimation (QE) has been rising in accordance with the ever-growing use and impact of wide-spread machine translations. Despite this international necessity, most attempts at QE have yet to deliver satisfactory results. We propose, a RObust Stratified EvoluTionary Trained Architecture (ROSETTA). We build on previous methods incorporating shallow and linguistic features, by adding the full representational power of Language Agnostic SEntence Representations (LASER). The results on this small dataset are outstanding, achieving first place in the Codalab Sentence-level QE task 2020 leaderboard with a pearson score of 0.263. We esteem that the method holds promise and would truly benefit from a larger dataset as well as a more diverse feature subset.

1 Introduction

Attempts at solving the QE riddle have come in many forms and varieties, gaining increasing interest over the past decade (Specia et al.; Unbabel et al.; Felice and Specia). Unfortunately, most solutions still fall short of providing meaningful results. Herein we describe a first attempt at using LASER embeddings alongside shallow and linguistic features, all under a stratified-convolutional-encoder model approach. Our results, although preliminary, indicate a promising alley for future work in QE.

2 Methods

2.1 Feature Extraction

Base Features

A model is only as strong as the data it has access to. Natural language being intrinsically sparse, finding appropriate representations of texts is an essential precursor of natural language processing (NLP). As such, much effort has been dedicated to extracting useful linguistic features (Specia et al.; Felice and Specia; Shah et al.; Felice, 2012) that can be used for various NLP tasks. QE in particular, is highly reliant on the features extracted from the source and target corpi to properly approximate a quality score (Specia et al.). Our approach was no different and as such, much effort was dedicated to extracting appropriate features for the QE task.

Table 1: Summary of language features used.

Features	Source	Target	Difference	Type
Number of Tokens	x	x		Shallow
Number of Non-Alphanumeric Token	x	x		Shallow
Sentiment Analysis	x	x		Linguistic
Proofreading Errors	x	x		Linguistic
Part of Speech Tags			x	Linguistic
Entity Tags			x	Linguistic
LASER Embeddings	x	x		Representations

Our approach combined a mix of both linguistic and shallow baseline features as listed in Table 1. Much pre-processing and feature extraction was conducted using third party libraries such as `TextBlob` (tex, b) and `Spacy` (spa). Third party libraries were chosen on the basis of them working for both English and German, their ongoing development trace, and their ease of integration with Python 3.6.

Sentiment analysis was performed using TextBlob (tex, b) and its German equivalent TextBlobDE (tex, a). For both target and source, tokens were lemmatised before running sentiment analysis to ensure all inflections of polarised words were accounted for. Lemmatisation was particularly important for the German text as we found results varied significantly pre and post lemmatisation.

Grammatical, spelling and styling errors were extracted using a python wrapper for 'Language Tool' (lan), an open-source proofreading software known for its wide-range of languages and exten-

sive error checks. Part of speech tags and entity tags were extracted via `Spacy` (spa). A layer of processing was then added on top of the tag extraction to map the difference in occurrences between source and target sentences for each part of speach (POS) and entity. German entities were more limited than the English ones and so both were constrained to named people, politically or geographically defined locations, organisations and miscellaneous entities such as events and nationalities.

LASER Embeddings

The AI team at Facebook recently open-sourced LASER, a library capable of encoding multi-lingual sentence embeddings (Artetxe and Schwenk). LASER uses an attention-driven transformer trained on all input-languages, where the encoder is a five-layer bidirectional long short-term memory neural network, generating a fixed-sized vector representation $Z_{LSR} \in \mathcal{R}^{1024}$ which allegedly groups close sentence meanings in the same neighborhood, regardless of language. LASER embeddings have been successfully used on a variety of classic NLP tasks such as machine translation and cross-lingual classification, consistently within the state-of-the-art models (Schwenk, 2018; Schwenk et al., 2019; Conneau et al., 2018; Schwenk and Li).

Our motivation for using LASER embeddings for the QE paradigm stems from our focus on meaning. Shallow or even linguistic features fail to encapsulate meaning: sentences may have divergent linguistic, shallow and even semantic features, yet convey the exact same concept. This incongruity constitutes one of the principal challenges of evaluating machine translations as well as a leading explanation for the shortcomings of quality estimation thus far in producing satisfactory correlation scores. Multilingual word and sentence embeddings may be viewed as a first step in the direction of cross-lingual meaning embedding whereby we can represent meaning in a language-agnostic N-dimensional space.

As such, our hope was to augment existing linguistic/shallow feature approaches using LASER embeddings as an adjunct to encapsulate meaning. Together, we hope these three attributes; linguistic relatedness, statistical similarity and meaning coherence represent the holy trinity in evaluating machine translations. To our knowledge, the LASER embeddings have yet to be applied to any QE tasks.

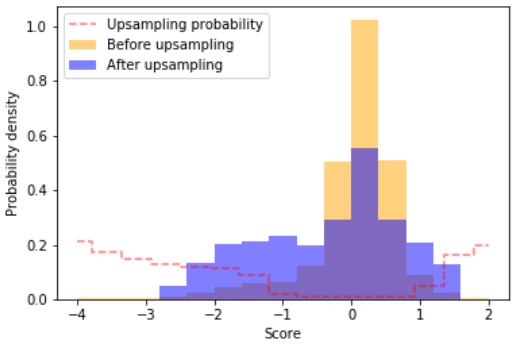

Figure 1: Upsampling distribution. The up-sampling probability at each scoring bin is inversely proportional to the number of samples in the training set. Distribution after up-sampling becomes more uniform.

2.2 Preprocessing

Two key aspects define our data: (1) the labels are unevenly distributed and (2) we have a small sample size (8000) with respect to a large feature space (2058). In light of this, we upsampled our features based on the score distribution. The pipeline is as follows and may be viewed in the `upsample()` method of the `Rosetta` class in `rosetta.py`: (1) Retrieve probability density distribution of scores; (2) Define the upsampling distribution for each bin according to α and β parameters; (3) Standardise up-sampling distribution to sum to one; (4) For each score, assign an up-sampling probability and normalise; (5) Randomly choose N scores and corresponding features according to the scaled up-sampling probability distribution, adding random noise scaled appropriately to the scores, ensuring we do not overfit to the training data; (6) Finally, concatenate the newly sampled feature score rows to full initial training set.

The outputs at various stages of the process are depicted in 1. As shown, up-sampling successfully augments our data to not only contain more samples, but also approximate more closely a uniform score distribution. The up-sampling parameters involved (N, alpha, beta, gamma) were found via manual tuning and were fixed throughout the evolutionary hyper-parameter search. Various other measures were taken to ensure our features were appropriately scaled and evenly distributed. The general pre-processing pipeline undertaken is outlined in Figure 2. Note we normalised all our features between -1 and 1 to weigh all features equally and thus smooth convergence (Xing et al.).

In preparation of the LASER embeddings, we encoded both source and target sentences into their

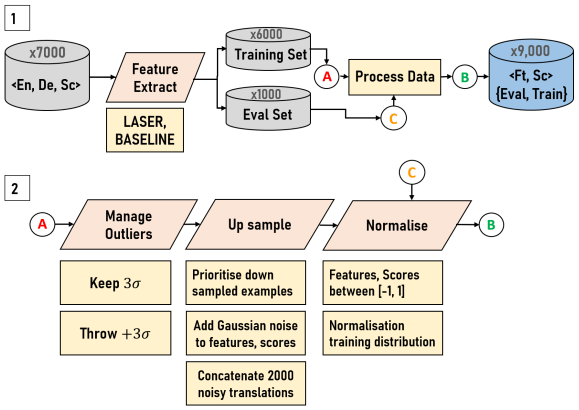

Figure 2: Pre-processing pipeline flow-chart. **1**: Pre-processing outline, **2**: Detailed "Process Data" block of the pipeline (**1**) involving the training and evaluation sets (A, C).

LASER 1024-dimensional vectors, which yielded an array $\in \mathcal{R}^{N\times 2\times 1024}$. We initially considered using a distance measure between two LASER vectors to represent meaning similarity. However, it has been argued that using various norms (e.g. euclidian or cosine) is not particularly useful in such a large dimensional space (Kusner et al.). Indeed, we found no correlation (-0.02) between LASER cosine distance and QE scores. As such, we opted for preserving the entire representational power of the full vector space and feeding these directly into our encoder model.

2.3 Neural Machine Model

Many successful feature extractors such as BERT or LASER rely on sophisticated transformers incorporating attention mechanisms and bi-directional recurrent networks (Artetxe and Schwenk; Devlin et al., 2018). We have limited sentences in the dataset and therefore attempt to exploit existing feature extractors such as LASER. As observed in (Specia et al.), baseline features also provide information which may aid in QE.

LASER embeddings are in $\mathcal{R}^{N\times 1024}$ whilst we use 10 baseline features from the English-German database. In order to encompass both solutions adequately we propose ROSETTA, a stratified encoder approach: LASER embeddings are encoded into a lower dimensional space $\mathbf{z}$ using several passes of a 1-D convolutional network followed by a feed-forward neural network. Here the baseline features are concatenated to the latent space $\mathbf{z}$ and a multi-layer perceptron (MLP) is used to directly regress the set of features $\{z_{LSR}$, Baseline$\}$ into the true QE scores observed in the training set. A schematic of the overall model is show in Figure 3. Note that PyTorch was the selected API for the

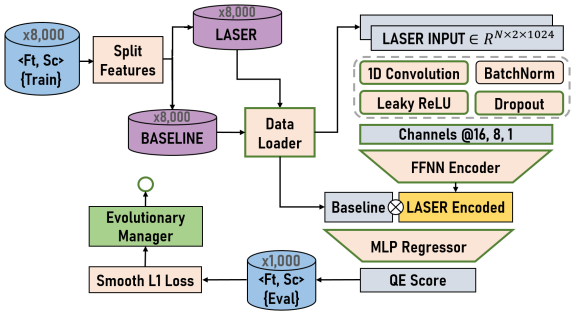

Figure 3: Stratified convolutional-encoder model for machine translation quality estimation. Includes a data-loader devised to control and process data coming from the LASER and BASELINE features; An evolutionary manager used to control the hyper-parameters - note we outline in green the segments of the model which are tuned through evolution; Concatenated baseline and LASER encoded features passed through an MLP regressor.

construction of the neural architecture.

There are several heuristics which critically improved the generalisation of our model during test time. The first one includes the addition of severe dropout regularisation in the MLP regressor. In contrast, less dropout in the FFNN layer allows feature extraction to be fully defined. As well as incorporating batch normalization on the CNN outputs, adding Leaky ReLU units instead of conventional ReLU activation functions improved convergence (smoother back-propagation of gradients) without the need for massive hidden layers to account for vanishing gradients in ReLU. Additionally, shifting from a standard mean squared error (MSE) loss function towards a smoothed L1 loss was pivotal: it is less sensitive to outliers and in some cases prevents exploding gradients. As outliers were present in the training dataset, it is sensible that this loss be, in practice, beneficial for the overall convergence of ROSETTA.

A large batch size was also found to be crucial: monte-carlo estimates of small batch-sized sentence translations is not a precise estimator of the true QE, considering the vast combination of words which may translate to similar QE scores.

2.4 Evolutionary Strategy

As highlighted in Figure 3, we use an evolutionary strategy (ES) to optimise: (i) information-based hyper-parameters (batch size, epochs, learning rates, etc.); (ii) model-based hyper-parameters (convolution channels, channel sizes, FFNN hidden layer sizes, etc.). The evolutionary manager (EM) deals with the QE correlation scores obtained during cross-validation on the evaluation set for a population of 25 randomly initialised agents.

Table 2: Metric scores produced by our model. During testing the model is re-trained under the same hyper-parameters, using both validation and training datasets.

Metric	Training	Validation	Test
Person ρ	0.37	0.44	0.26
MAE	0.48	0.50	0.53
RMSE	0.59	0.62	0.91

Using the mean cross-validated score as a fitness score, the best 10 agents are selected for crossover and mutation of the genotype. The new population is generated from the new set of hyper-parameters. This process is repeated until convergence. The EM also performs k-fold cross validation for each agent when evaluating its fitness value. As such, convergence on the test scores is assured to be robust as each agent evaluates its genotype on various training/validation splits. This is a necessary process undertaken in order to validate the robustness of the model, especially considering the small sample size of the dataset.

3 Results

After 14h of training managed by the EM on an Azure VM (Tesla k80 GPU) we find an optimal set of hyper-parameters (given the features and training scores used in QE). These correspond to: (a) Info-Based - {Batch Size: 500, learning rate: 4e-4, epochs: 30}; (b) Model-Based - (i) Convolutional {Channels: (2, 17), Kernel Size: (2, 4), Leaky ReLU Slope: 0.3, Dropout: 0}; (ii) FFNN {Hidden Layers: (40, **27**), Dropout: 0.1}; (iii) MLP {Hidden Layers (26, 6, 1), Dropout: 0.4}. Although most converged agents had different hyper-parameter configurations, we found through evolution that there were some agreements i.e. all agents had a LASER encoding dimension (second hidden layer dimension of the FFNN) of around **27**. The cross-validated results for the latter model are summarised in Table 2.

We also compare the distribution of the predicted scores and the real scores in the evaluation set (Figure 4). The heavy tailed Gaussian is properly bounded by the model predictions, an attribute which was not observed without upsampling. Moreover, using the smooth L1 loss was beneficial to this remark in comparison to the MSE loss.

4 Discussion

After extensive hyper-parameter search, the results obtained are highly satisfactory. All our best models converge to similar losses and correlations suggesting that the model has reached the extent to which it has exploited its ability to draw patterns from the data given both the sample size and the features we have provided it with. Notice that during testing (Table 2), the pearson score computed using the training scores reduces significantly. This points towards the explanation that there are not enough data points to fully encode the link between the LASER and baseline embeddings to the QE score, or that the features extracted are not representative enough.

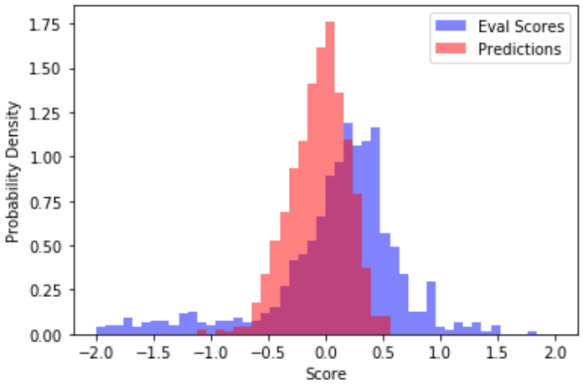

Figure 4: Distribution of the predicted scores and the test scores. There are evident similarities which showcase that the model properly embodies the test domain.

The competition leader board suggests that we have developed a robust and competitive model, achieving the best scores so far, both in terms of pearson's score (0.263) and RMSE (0.909). This revamps the stratified approach: similar to OpenKiwi's (Unbabel et al.) modular approach, ROSETTA is flexible to incorporate any other features necessary ad-hoc, the feature can just be concatenated via encoding in a similar fashion to LASER. As such, it would be most interesting to incorporate more features into this model and train on a larger data-set to evaluate the global QE capability of our model.

5 Conclusion

We have introduced a new framework for solving the QE problem and have demonstrated its validity on the Sentence-level QE task 2020. Much work still remains to be conducted, in particular refining the feature extraction pipeline and using larger datasets. Overall, we present a solution which is worth exploring in future works. Please find the code in our open-source Git repo [1].

[1] https://github.com/ludomitch/rosetta

References

LanguageTool - Spell and Grammar Checker.

spaCy · Industrial-strength Natural Language Processing in Python.

a. textblob-de — textblob-de 0.4.4a1 documentation.

b. TextBlob: Simplified Text Processing — TextBlob 0.15.2 documentation.

Mikel Artetxe and Holger Schwenk. Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond. Technical report.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating Cross-lingual Sentence Representations. pages 2475–2485.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Mariano Felice. 2012. Linguistic Indicators for Quality Estimation of Machine Translations. Technical report.

Mariano Felice and Lucia Specia. Linguistic Features for Quality Estimation. Technical report.

Matt J Kusner, Yu Sun, Nicholas I Kolkin, and Kilian Q Weinberger. From Word Embeddings To Document Distances. Technical report.

Holger Schwenk. 2018. Filtering and Mining Parallel Data in a Joint Multilingual Space. pages 228–234.

Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. 2019. WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia.

Holger Schwenk and Xian Li. A Corpus for Multilingual Document Classification in Eight Languages. Technical report.

Kashif Shah, Trevor Cohn, and Lucia Specia. An Investigation on the Effectiveness of Features for Translation Quality Estimation. Technical report.

Lucia Specia, Kashif Shah, Jose G C De Souza, Trevor Cohn, and Bruno Kessler. QuEst-A translation quality estimation framework. Technical report.

Fábio Kepler Unbabel, Jonay Trénous, Marcos Treviso, Miguel Vera, Unbabel André, and F T Martins. OpenKiwi: An Open Source Framework for Quality Estimation. Technical report.

Chao Xing, Chao Liu, and Dong Wang. Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation. Technical report.