SemEval 2023 Task 9: Multilingual Tweet Intimacy Analysis

Ghazal ZamaniNejad - 401722244

Mohammadmostafa Rostamkhani - 401722235

Abstract:

In this paper we build a model using large multilingual pre-trained language model XLM-RoBERTa (XLM-T) for regression task and fine-tune it on the MINT (Multilingual Intimacy Analysis) dataset which covers 6 languages for training and 4 for testing zero-shot performance of the model. This dataset includes tweets annotated for 6 languages. annotations are intimacy scores. We add some dense and dropout layers for the task of regression.

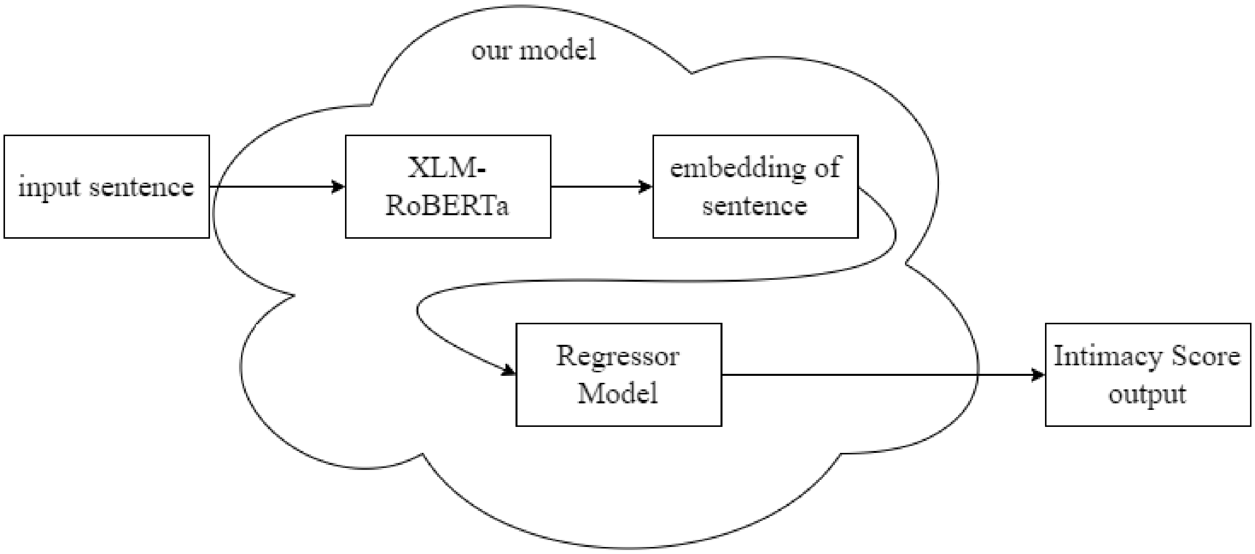
Introduction:

Our task is to predict intimacy scores of tweets for 10 languages including 6 languages seen during training and 4 languages for zero-shot performance of our model.
The task of predicting the intimacy score of tweets is important because we can understand and extract some social information. Our dataset covers 6 languages for training including English, Spanish, Portuguese, Italian, French, and Chinese. Our dataset also covers 4 languages for zero shot learning of our model. These languages include Hindi, Korean, Dutch, and Arabic.
The main strategy of our team was to use XLM-RoBERTa which is a multilingual pre-trained large language model, pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages (it is a multilingual version of RoBERTa) and it was pre-trained with the Masked Language Modeling (MLM) objective. Our model contains XLM-RoBERTa for extracting embeddings of sentences and also some dense and dropout layers for predicting (approximate) the intimacy score of a sentence. We fine-tune the last few layers of XLM-T and train the regressor part of our model using the training dataset. Since our dataset was almost small, we used some methods for augmenting our data.

Background:

The inputs to our model were some tweets collected from twitter and we expect our model to produce a number between 1 to 5 as an intimacy score (in fact our model produces a number between -1 and 1 and then we scale it to 1 to 5 for ~~our~~ final prediction).

A schematic of our model is shown below:



Output of XLM-RoBERTA which is an embedding of the input sentence is a tensor of shape (768*1) and is fed to the regressor model to predict a number as an intimacy score. Furthermore, some augmentation methods were added for zero-shot languages in order to improve the model performance.

You can find more details about the dataset in its paper.

System Overview:

Preprocessing Data:

We use the MINT dataset as our main dataset. Since we have no training data for zero shot languages, we decided to translate tweets from one language to zero-shot languages using Google Translate API to train our model on them to improve its performance.

There we faced some issues. For example, we don’t want @user which are mentions of not verified users to be translated. So we decided to remove all the mentions in the translated sentences. We choose english tweets for translating to zero-shot languages for preparing our data. We suppose that the translated sentence got the same intimacy score. After augmentation, we store all data in a csv file and feed it as input to our model.

Creating Dataset:

For creating the dataset we store tokenized sentences and intimacy scores. We use XLM-RoBERTa Tokenizer for tokenizing sentences. We save tokenized sentences for feeding to the XLM-T. It also has special tokens for emojis which is what we need because emojis are used a lot in tweets. We also normalize scores using MinMaxScaler and save them as desired output of our model.

Model:

XLM-RoBERTa is chosen because it was pre-trained on multilingual tweets which contain emojis which are desired. The structure of the model is as follows: some dense layers with the activation of ReLU as the activation function with a dropout layer and again a dense layer for generating one number as output of the regression task. MSE (mean square error) loss function is chosen as loss function to minimize during the training phase. AdamW is chosen as the optimizer. We fine-tune the model with three different settings. First we freeze all the XLM-T layers and just train the regressor part. In the second setting, just fine-tune the last dense layer of the model and freeze all the others. In the third setting, layers 0 to 10 were freezed and fine-tuned layer 11 and the last dense layer.

Post-processing Data:

The outputs of the model are between -1 and 1. So we scale them to get a score between 1 to 5.

Experimental setup:

The data is splitted into three groups with the portion of 0.9, 0.05, 0.05 for training, testing and validation.

We try some experiments with different setups. Learning rate of 0.0001 is chosen after some trial and error. Choosing a large learning rate causes overshoot from optimum point and choosing small one causes stucking in local minima.

For the number of epochs for training, 6 is chosen with trial and error. Choosing a small number of epochs is because of the large language model that has been used, and was pre-trained on tweets which are the same data as our dataset.

Batch size of 32 is chosen.

Results:

We run 5 experiments with different setups:

	Dataset size	Number of training epochs	Learning rate	Freezed & Fine-tuned layers	Train loss	Validation loss
exp1	original	5	1-e3	All layers freezed	0.0344	0.0322
exp2	original	5	1-e3	1-11 freezed	0.0321	0.0305
exp3	augmented	5	1e-5	1-11 freezed	0.0302	0.0281
exp4	augmented	6	1e-5	1-10 freezed	0.0262	0.0235
exp5	augmented	6	1e-4	1-10 freezed	0.0254	0.0231

Conclusion:

In conclusion, augmenting the dataset affects the results. Choosing the right learning rate also has effects on converging faster and the results. Training the last few layers of XLMT improves results.

Future works:

For future works, some ideas are listed below:

- Try different types of regressor like XGB as Regressor
- Use different translation pairs for data augmentation: use all pairs of languages existing in dataset for augmenting dataset
- Use GPT to generate more data from embeddings for augmenting dataset
- Use T5 model as base model for regression task
- Use sentiment analysis of sentences and feed them as input to our model
- Detect some sensitive and immoral words and feed them to our model as a feature because most of intimate tweets are immoral
- We can use DeepL API for more accurate translations