YNU-HPCC at SemEval-2023 Task 9: Pretrained Language Model for Multilingual Tweet Intimacy Analysis

Qisheng Cai, Jin Wang and **Xuejie Zhang**
School of Information Science and Engineering
Yunnan University
Kunming, China
Contact: kujou@mail.ynu.edu.cn, {wangjin, xjzhang}@ynu.edu.cn

Abstract

This paper describes our fine-tuned pretrained language model for task 9 (Multilingual Tweet Intimacy Analysis, MTIA) of the SemEval 2023 competition. MTIA aims to quantitatively analyze tweets in 6 languages for intimacy, giving a score from 1 to 5. The challenge of MTIA is in semantically extracting information from code-mixed texts. To alleviate this difficulty, we suggested a solution that combines attention and memory mechanisms. The preprocessed tweets are input to the XLM-T layer to get sentence embeddings and subsequently to the bidirectional GRU layer to obtain intimacy ratings. Experimental results show an improvement in the overall performance of our model in both seen and unseen languages.

1 Introduction

Intimacy involves lots of social information (e.g., social norms). (Pei and Jurgens, 2020) By evaluating the intimacy score of tweets crawled on social media sites like Twitter quantitatively, it is possible to mine more meaningful social information. MTIA aims to build a sentiment analysis model (or system) to predict the intimacy scores, ranging from one to five, of tweets in a Twitter dataset containing ten languages, including English, Spanish, Portuguese, Italian, French, Chinese, Hindi, Dutch, Korean and Arabic. Among those ten languages, the first six are observable (i.e., appear during training), and the rest are unobservable (used for zero-shot learning).

Tweets abound with online slang, emojis and face characters, which usually contain extra intimate thoughts. Thus, it is critical to extract semantic information from those unconventional texts. However, many accessible pre-trained models (e.g., XLM) lack consideration of non-traditional text in the training process. The non-traditional text needs elimination to de-noise when we perform transfer learning on these models. Another feasible approach is to convert these non-traditional symbols to conventional words. However, the extraction of information from face characters remains a challenge.

Fortunately, we can perform transfer training using the XLM-T model (Barbieri et al., 2021) trained on the Twitter dataset, thus avoiding information loss in the symbolic text. Additionally, we suggest a better way to link bidirectional GRU (Cho et al., 2014) layers at the XLM-T. The approach, which combines attention and memory mechanisms, is inspired by the analogy to humans. We will demonstrate experimentally that this measure enhances the model's overall performance, both seen and unseen text. Our model achieved 13/45 places in terms of overall performance in the leaderboard, with both Dutch and Arabic ranking fourth. The code will be available at https://github.com/cskujou/tweet-intimacy.

The roadmap for this paper is as follows. Section 2 describes the work related to sentiment analysis. Section 3 describes the architecture of our model. Section 5 reports the experimental results. Section 6 concludes the paper.

2 Related Works

In the past, researchers have used lexicon-based methods and machine-learning approaches for sentiment analysis. (Wankhade et al., 2022) Since then, the performance of sentiment analysis tasks has significantly increased because of the development of deep learning algorithms. CNN (LeCun et al., 1998), LSTM (Hochreiter and Schmidhuber, 1997), and GRU (Cho et al., 2014) are examples of deep learning models. Wang et al. (2022) proposed a contextual sentiment embedding model that can distinguish the meaning of the same word in different contexts and improve the performance of the sentiment task. Recently, by resolving the issue of long-distance dependency, BERT-based large-scale pre-training models achieved a significant advancement in sentiment analysis (Zheng

Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023), pages 733–738

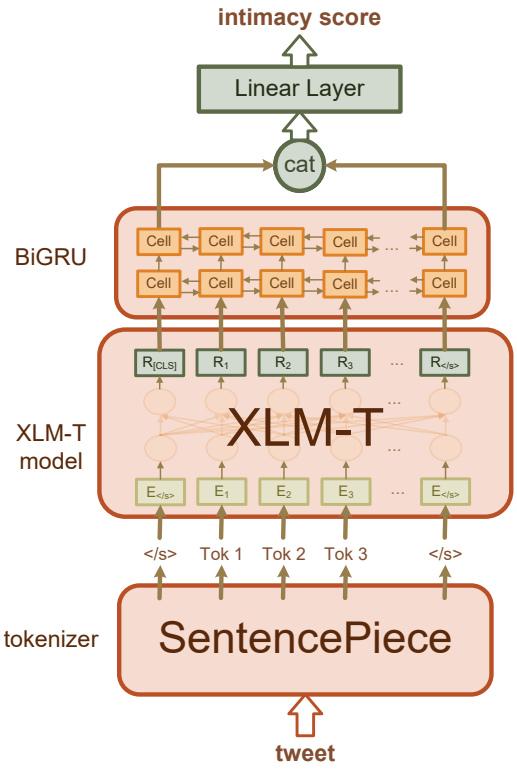


Figure 1: Model architecture of XLMT-GRU

et al., 2022; Bai et al., 2022). Early multilingual NLP models included mBERT (Devlin et al., 2018), XLM (Conneau and Lample, 2019), and mBART (Liu et al., 2020). Goyal et al. merged the XLM and the RoBERTa model to create the more potent XLM-R (Conneau et al., 2020). Barbieri et al. obtained the XLM-T using an identical architecture to the XLM-R, trained on the Twitter dataset. The XLM-T has a significant improvement in its ability to extract symbolic information.

3 Model Description

The XLM-T model (Barbieri et al., 2021) was applied as the baseline model. Moreover, we connect two more bidirectional GRU layers on top of XLM-T to get the improved model called XLMT-GRU. Figure 1 illustrates its architecture. We'll go into more detail about this model in the following paragraphs.

3.1 Preprocess

Preprocessing can help to reduce the amount of bias in the model, leading to fairer and more accurate results. Notice the presence of the username in the dataset, whose pattern is a token beginning with "@". We eliminate all usernames after preprocessing since we empirically think they are unre-lated to intimacy. Additionally, the presence of a username may interfere with predictions. For ex-ample, a username token "@love_jenny" may increase the intimacy of a sentence. We will maintain each individual "@" character and only remove those whose token length is more than one since the "@" character may also be used to represent the word "at".

3.2 Tokenizer

A tokenizer splits a text into tokens, which are then used by the language model to generate responses. SentencePiece (Kudo and Richardson, 2018), a language-independent tokenizer, is used by XLM-T. A sentence of any language can be broken up into smaller pieces efficiently by using it. It accomplishes this by combining the BPE and unigram algorithms. The unigram model assigns a probability to each word and presumes that each is independent. The probability is calculated using the maximum likelihood estimation method, which assumes that the probability of each word is equal to the number of times it appears in the corpus divided by the total number of words in the corpus. Mathematically, this can be expressed as:

$$P(w) = \frac{\text{Count}(w)}{N},$$

where $P(w)$ is the probability of a given word w, Count(w) is the number of times the word appears in the corpus, and N is the total number of words in the corpus. The BPE algorithm examines at the frequency of each subword in the corpus and assigns a single token to the most frequent subwords. This process is repeated until no byte pair appears more than once in the vocabulary. The Sentence-Piece algorithm is a modification of the BPE algorithm that adds a smoothing factor to the probability calculation, which makes the model more robust to unseen input. Also, SentencePiece has a parameter that allows users to specify the number of tokens they want to generate.

A statement like "Using SentencePiece for tokenization." would be split up into subword units like ['_U', 'sing', '_Sent', 'ence', 'Pie', 'ce', '_for', '_to', 'ken', 'ization', '.', '_']. Note that the character "_" is not an underscore, but a Unicode character with the code value "U+2581", indicating the beginning of a sentence or a space. XLM-T has a vocabulary size of 250K and can convert tokens into their corresponding IDs. Subword information

can be extracted efficiently using SentencePiece. As a result, the language model can comprehend the statement more clearly and make a more accurate sentiment analysis.

3.3 XLM-T Model

The main advantage of using pre-trained models is that they reduce the cost and effort required for deep learning since they eliminate the need to spend time and money gathering and cleaning data when training a model from scratch. Additionally, pre-trained models typically provide superior results because they already have optimized weights and can converge faster. The XLM-T is a pre-trained model that performs superiorly on multilingual natural language processing tasks. It consists of an embedding layer and twelve transformer-based encoding layers, each with 768 hidden units and 12 attentional heads. The most important among them is the encoding layer. Each encoding layer comprises two sublayers: a multi-head self-attention layer and a feed-forward layer.

Each token is converted into a word embedding, while the position of the tokenizer in the sen-tence is converted into a positional embedding. The first layer of encoding receives the sum of these two embeddings and feeds its output to the second layer, then the second layer feeds its output to the third layer, and so on. In the end, the token passes through 12 encoding layers, resulting in a vector of length 768. The main advantage of XLM-T is that it is trained on the Twitter dataset, in line with MINT. Thus, XLM-T is better suited to unconventional text, particularly the semantics of emojis.

3.4 Bidirectional GRU

Gate Recurrent Unit (GRU) can learn representations from both the past and future time sequences to capture long-term dependencies, and it can help to reduce the vanishing and exploding gradients issue. In comparison with the LSTM, the GRU has fewer parameters, resulting in faster training and less data. Two GRUs make up a bidirectional GRU, also known as a BiGRU. One processes the input moving ahead, while the other moves backwards. GRU comprises two gates: the update gate and the reset gate. The update gate instructs the model on which parts of the input are important or insignificant, whereas the reset gate instructs the model on how to keep track of the context and how much of the previous state to pass on to the next step. By doing this, the model can capture long-term dependencies in both directions. Finally, the output values of the two GRUs are concatenated and used as the bidirectional output. For instance, input a token of length N and then BiGRU will output a matrix of $2H \times N$, where H is the number of hidden cells in BiGRU.

3.5 Linear Layer

The goal of the linear layer is to transform the BiGRU result into a scalar. Due to the GRU output being a sequence, we concatenate the first and last outputs to obtain a vector of length $4H$. A linear layer with $4H$ hidden cells is then used to translate the vector into a value.

The loss function in our model is the MSE function. Assume that the actual value is y and the model's predicted value is $\hat{y}$, where i is the index. Then the MSE can be calculated by the following formula.

$$\frac{1}{n}\sum_{i=1}^{n}(y-\hat{y})^2,$$

where n is the size of training set.

4 Experimental Setup

In this section, we first introduce the dataset and the evaluation metric and then go over the model implementation detail. Next, we demonstrate the improvement of our model over the baseline model through experiments. Further, we will include two models for comparison purposes.

- XLM-R: XLM-RoBERTa-base model
- XLMR-GRU: Similar structure to the XLMT-GRU, except that the XLM-T layer has been replaced by the XLM-RoBERTa-base.

The experiments are divided into two parts. The first part reveals the performance of our model on the observable language, while the second part shows the performance on the unobservable language.

4.1 Datasets

The dataset used for the experiments is the MINT provided by Pei et al. (Pei et al., 2022) There are 13,384 tweets in the MINT, and there are tweets in ten different languages. Table 1 lists the number of tweets in each language. The model will be trained using the first six languages in the table, and the final four languages will be used as test data

Language	# Train	# Total	Avg Len	Vocab
English	1,587	1,984	19.64	8,056
Spanish	1,592	1,991	20.07	7,938
Portuguese	1,596	1,996	17.44	6,346
Italian	1,532	1,916	18.13	7,383
French	1,588	1,981	20.02	6,987
Chinese	1,596	1,996	27.46	9,928
Hindi	0	280	28.80	4,882
Korean	0	411	27.40	4,068
Dutch	0	413	19.82	5,183
Arabic	0	416	23.71	4,421
Total	9491	13,384	36,90	4,4971

Table 1: Sample size of the MINT dataset

to evaluate the performance of zero-shot learning. We split the training set in the ratio of 8:2 into a training set and a development set for the fine-tuning phase. Only the training set is used to train the model of different hyperparameters. And we choose the model that outperforms the others on the development set.

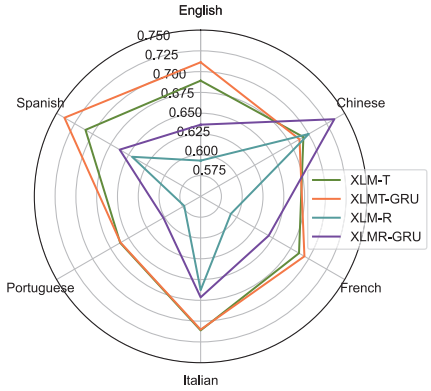
4.2 Evaluation Metrics

MTIA requires the model to output a number between one and five as a predictor of the intimacy score, where one means not intimate and five means most intimate. We will use Pearson's r as an evaluation metric of model performance. The closer this indicator is to 1, the more closely the model's prediction resembles the actual number, indicating a higher degree of accuracy. The formula of Pearson's r is as follows.

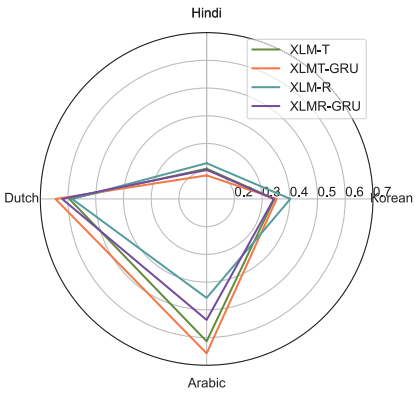
$$r = \frac{\sum_{i=1}^{m}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{m}(\hat{y}_i - \bar{\hat{y}})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}.$$

4.3 Implementation Details

To implement the entire model, we used the transformer library from the Hugging Face community and the PyTorch deep learning framework. The pre-training parameters of the model are twitter-xlm-roberta-base developed by cardiffnlp. Regarding the hyperparameters, we keep the parameters of XLM-T as default (i.e., twelve encoding layers, 768 hidden units and 12 attention heads). There are two bidirectional GRU layers containing 256 hidden units. The dropout ratio between the XLM-T, BiGRU and linear layers is 0.1. We use the Adam optimizer to train the model. Adam can often converge faster than other algorithms, and it helps to mitigate the weight decay problem. The learning rate is 5e-6, the number of epochs is 3, the size of the batch at training is 8, and the size of the batch at evaluation is 16.



(a) Seen Language



(b) Unseen Language

Figure 2: Performance of XLMT-GRU

4.4 Comparative Results

Table 2 and Figure 2 (a) show the performance of models on seen languages. Note that XLMT-GRU greatly enhances the model's overall performance in the seen languages by significantly increasing prediction accuracy in both English and Spanish. On the other hand, the XLMR-GRU also significantly outperforms the XLM-R, confirming the GRU layer's capability to boost the performance of the XLM model. We observe that XLM-T outperforms XLM-R in general. It motivates us to believe that pre-trained models and tasks should correspond as closely as possible, particularly for the dataset. To our disappointment, the performance of XLM-T on Chinese is somewhat weakened. This is because the Chinese text in the Twitter dataset is relatively small, roughly one thousandth the size of the English text. Moreover, the Pearson's r be-

Model	English	Spanish	Portuguese	Italian	French	Chinese	Seen
XLM-R	0.593	0.645	0.572	0.662	0.591	0.700	0.638
XLMR-GRU	0.636	0.662	0.601	0.671	0.644	0.735	0.670
XLM-T	0.689	0.710	0.661	0.711	0.687	0.687	0.701
XLMT-GRU	0.711	0.739	0.661	0.710	0.694	0.687	0.710

Table 2: Performance in seen language

Model	Hindi	Dutch	Korean	Arabic	Unseen
XLM-R	0.228	0.585	0.402	0.456	0.353
XLMR-GRU	0.205	0.621	0.342	0.536	0.375
XLM-T	0.209	0.595	0.344	0.613	0.400
XLMT-GRU	0.185	0.645	0.352	0.657	0.435

Table 3: Performance in unseen language

tween our model's predictions and the actual values is 0.7109 on the test set and 0.7064 on the development set. The similarity between the two numbers suggests that our model is not overfitting.

Table 3 and Figure 2 (b) show the performance of models on unseen languages. Note that the XLMT-GRU's performance in Hindi has regressed. However, there is a significant improvement in performance in Dutch and Arabic. Thus, our model outperformed the baseline model on zero-sample learning. The same results can also be obtained by comparing the XLMR-GRU with the XLM-R. To explain why the performances of zero-shot learning results are so different, we analyze them in terms of linguistic similarity. First, Dutch and Portuguese are both Romance languages and share several similarities in terms of grammar, syntax, and vocabulary. Second, the typology and script of Arabic differ from that of English, but the representations are quite similar. However, Hindi and the six seen languages have very different grammatical rules and writing systems. Finally, the Korean language has been influenced by Chinese and therefore performs slightly better than Hindi. However, Korean is an agglutinative language and Chinese is an analytic language, and they have different grammar and morphology. Those facts support our experimental results: all models performed much better in Arabic and German than in Hindi and Korean. Figure 3 illustrates the prediction errors of the XLMT-GRU model in various languages. The horizontal coordinate represents the difference between the predicted and actual values, while the vertical coordinate represents the probability density. Notice that the expectation of the error on

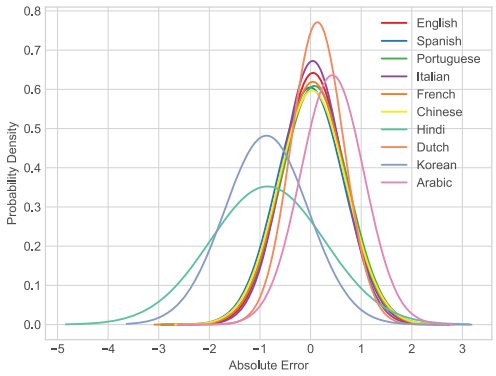


Figure 3: Probability Density of Prediction Error

the seen languages is close to 0. The maximum deviation value is around 1.8. On the unseen languages, the expectation deviates significantly from 0. The larger the offset value, the worse the prediction. Our model underestimates the intimacy of Hindi and Korean and slightly overestimates the intimacy of German and Arabic. Comparing the two experiments together, we find that integrating the attention and memory mechanisms results in an improvement in model performance. This is because GRU converges more quickly on small data sets and tends to outperform a simple linear layer.

5 Conclusion

In this paper, we study the multilingual intimacy analysis task. We attempted to combine XLM-T and BiGRU and found that the overall performance of our model outperformed the baseline model. Our model ranked 13^{th} on the leaderboard and 11^{th} on unobservable languages. Combining at-

tentional and memory mechanisms can boost the performance of our model for sentiment analysis on small datasets. In the future, we will also try to integrate other models (e.g., TreeLSTM), mainly to complement the shortcomings of our current model on Hindi.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61966038 and 62266051. The authors would like to thank the anonymous reviewers for their constructive comments.

References

Wenqiang Bai, Jin Wang, and Xuejie Zhang. 2022. YNU-HPCC at SemEval-2022 Task 4: Finetuning Pretrained Language Models for Patronizing and Condescending Language Detection. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 454–458.

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2021. XLM-T: Multilingual language models in twitter for sentiment analysis and beyond. *arXiv preprint arXiv:2104.12250*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8.

Jiaxin Pei and David Jurgens. 2020. Quantifying intimacy in language. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5307–5326.

Jiaxin Pei, Vítor Silva, Maarten Bos, Yozon Liu, Leonardo Neves, David Jurgens, and Francesco Barbieri. 2022. Semeval 2023 task 9: Multilingual tweet intimacy analysis. *arXiv preprint arXiv:2210.01108*.

Jin Wang, You Zhang, Liang-Chih Yu, and Xuejie Zhang. 2022. Contextual sentiment embeddings via bi-directional gru language model. *Knowledge-Based Systems*, 235:107663.

Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. 2022. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55.

Guangmin Zheng, Jin Wang, and Xuejie Zhang. 2022. YNU-HPCC at SemEval-2022 Task 6: Transformer-based Model for Intended Sarcasm Detection in English and Arabic. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 956–961.