

EPUTION at SemEval-2018 Task 2: Emoji Prediction with User Adaption

Liyuan Zhou^{1,2} Qiongkai Xu^{1,2} Hanna Suominen^{1,2,3,4} Tom Gedeon¹

¹ The Australian National University, Canberra, ACT, Australia

² Data61, CSIRO, Canberra, ACT, Australia

³ University of Canberra, Canberra, ACT, Australia

⁴ University of Turku, Turku, Finland

{Liyuan.Zhou, Qiongkai.Xu, Hanna.Suominen, Tom.Gedeon}@anu.edu.au

Abstract

This paper describes our approach, called *EPUTION*, for the open trial of the SemEval-2018 Task 2, Multilingual Emoji Prediction. The task relates to using social media — more precisely, Twitter — with its aim to predict the most likely associated emoji of a tweet. Our solution for this text classification problem explores the idea of transfer learning for adapting the classifier based on users’ tweeting history. Our experiments show that our user-adaption method improves classification results by more than 6 per cent on the macro-averaged F1. Thus, our paper provides evidence for the rationality of enriching the original corpus longitudinally with user behaviors and transferring the lessons learned from corresponding users to specific instances.

1 Introduction

Twitter sentiment analysis is an essential problem for companies and organizations to computationally measure customers’ perceptions which attracts attention from fields of both social media analytics and natural language processing (Rosenthal et al., 2017; Felbo et al., 2017; Mac Kim et al., 2017). A Twitter message, called a *tweet*, is generally composed of text, *emojis*, links, and mentioned users, known as *tweeters*. An emoji is a small picture or symbol of a standardized set to represent a feeling or another concept (Dictionary.com, 2018), contributing to the sentiment of its sender (Barbieri et al., 2017). Consequently, techniques for *emoji classification* are relevant and can be used to transfer information to subsequent tasks of sentiment, emotion, and sarcasm analysis (Felbo et al., 2017).

The SemEval-2018 Task 2 challenges its participants to perform multilingual *emoji prediction* in *English* and *Spanish*. The top-20 most frequent emojis of each language are annotated as tweets’

class labels. To encourage systems with better performance on less frequent emojis, the *macro-averaged F₁ score* (Macro-F) (Suominen et al., 2008) is used as the official evaluation measure.

Emoji prediction is widely formalized as a text classification problem in which the state-of-the-art systems fail to perform satisfactory (Barbieri et al., 2018). Because individual users enjoy diverse preferences in their emoji usage, it is hard to train a generalized classifier to tackle emoji prediction. As demonstrated in Table 1, with two examples of simple tweets with various annotations from the training set, even with exactly the same tweet texts, different tweeters have various choices of emojis, such as **i**) a user can select one of the emojis express the same emotion and **ii**) a user can have different attitudes towards the same objects or topics.

| Tweet | Emoji |
|----------------------|---|
| @user happy birthday |    |
| @ new york, new york |    |

Table 1: The diversity emojis by different users.

In light of such observations, we propose to utilize a user adaption method to capture the specific preference for each individual user. We propose *Emoji Prediction with User Adaption* (*EPUTION*). It trains user-adapted classification models by applying tweeters’ tweeting history to personalize a basic model trained by the benchmark training data. We implement the method on SemEval-2018 Task2 in English, where the basic model is competitive to the state-of-the-art systems, while the user-adaptation model further improves the classification results.

2 System Description

In this section, we describe the text classification method and user adaption approach.

2.1 Text Classification

The text classification component of our system is based on *FastText*¹ (Joulin et al., 2017), which can achieve results comparable to those by the state-of-the-art deep learning methods but with many orders of magnitude less running time. FastText feeds a linear classifier with averaged word representations as follows:

$$P(y|x_n) = \text{Softmax}(BAx_n) \quad (1)$$

where y refers to the *class label* of a given document, x_n is the respective *normalized bag of features vector* of the document, and A and B are the *weight matrices*. The *cross entropy loss* is updated to optimized for parameter learning. The model is trained using the *stochastic gradient descent* algorithm with a linearly decaying learning rate.

To optimize the computing time, *Hierarchical Softmax* based on the *Huffman tree* (Mikolov et al., 2013) is used to estimate label distribution. The probability of the label node n_y in the Huffman tree, with parents n_1, \dots, n_p , is calculated as

$$P(n_y|x) = \prod_{i=1}^p P(n_i|x). \quad (2)$$

In order to capture the word order information in the text, bag of n -grams are used as features.

2.2 User Adaption Framework

Our *User Adaption* (UA) framework is composed of the following two main components (Figure 1): a *pre-training process* and an *adaption process*.

During the pre-training process, we train a basic classification model M_b using the training set of the benchmark corpus C_b through FastText. During the adaption process, for each user u_i , we adapt the basic classification model M_b to a user-adapted model M_i . Namely, we initialize the parameters B_i and A_i of M_i with pre-trained parameters from M_b , and train M_i for 5 epochs using the retrospective tweet collection C_i of u_i . Out-of-vocabulary words in C_i are randomly initialized in our experiments.

3 Experimental Setup

In this section, we will describe our supplementary data collection process, model, and test settings.

¹<https://fasttext.cc/>

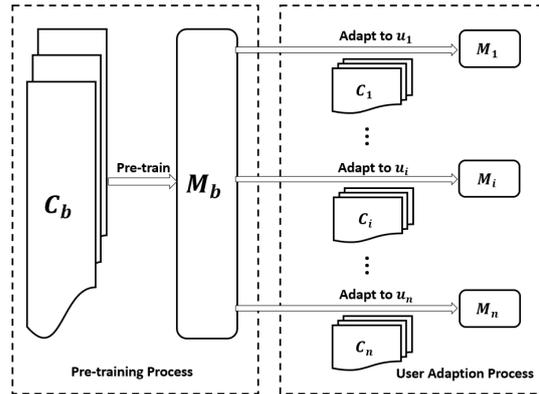


Figure 1: An overview of the user adaption framework.

3.1 Supplementary Data Collection

To implement our user adapted classification system, supplementary tweets are collected for each user. First, for each tweet of the emoji prediction task, the original tweeter who posted the tweet is retrieved by using the *Twitter Application Programming Interfaces*² (API). In order to map a tweet to its tweeter, we use the content of the tweet as a query to search for a match in Twitter. If precisely one result is retrieved and its content is precisely the same as the query text, the user who posted the retrieved tweet is assigned as the tweeter. Otherwise, no specific tweeter is assigned. Second, for each retrieved user, a retrospective collection of tweets is crawled from the most recent to the maximum number of 3,200 tweets³. The current tweet used to retrieve the additional tweeter data is excluded from the user’s tweet collection⁴.

After enriching the task corpus by these user-specific collections, text content of the tweets is extracted using official scripts (Barbieri et al., 2017), removing hyperlinks while keeping texts and emojis. The tweets with only one emoji are selected, where the emoji is considered as the class label of the tweet. To ensure no overlapping instances exist between the test set and additional data that is collected for the user adaptation model (i.e., the collection of users’ historical tweets), we remove the instances in the retrieved dataset that match the test tweets. More over, drawing from the use of the inverse document frequency in information retrieval as a way to scale down words

²<https://developer.twitter.com/>

³This number is determined by the Twitter API limitation.

⁴The crawling was performed for our SemEval-2018 Task 2 submission on 2nd February, 2018.

that only appear in few documents as too specific, all tweets that occur only in a single user’s tweet collection are filtered out. This post-processing eliminates accidentally collected test cases where a user name cannot be retrieved, but keeps general cases that commonly appears in a tweet message such as *Happy Birthday* and *Good Morning*.

To summarize the dataset setting, there are 487,088⁵ and 50,000 samples in the benchmark training and test sets, respectively. From all the tweets in the test set, 22,642 of them matched a tweeter, from 20,594 unique tweeters⁶. The final supplementary tweet collection contains 2,565,459 tweets, with user IDs. This is about five times the size of the benchmark training set.

3.2 Model and Test Settings

Because the number of retrospective tweets from a single user is limited⁷, the performance of training one model for each user is unsatisfactory in our preliminary experiments. Therefore, we apply a pre-training model to the benchmark training data of the task as a way to achieve properly initialized model parameters.

We implement the following three models:

- **FastText** is the baseline text classification model trained on the benchmark training set.
- **Data Augment** (DA) is the adapted model that used all tweeters’ tweets grouped as a whole.
- **Individual User Adaption** (IUA) model is the adapted model that tailored the model to each individual tweeter’s tweets.

After grid searching for the parameters on the benchmark development set, the *initial learning rate* α is set to be 0.01. The baseline model uses 100 dimensions of word vectors and 5 words in the context. It is trained over 50 epochs. The UA model has $\alpha = 0.05$ and is trained over 5 epochs. As the key point of this paper is the user adaption model, we explore the basic text classification features of *unigrams*, *bigrams*, and *trigrams* in our primary experiments. Trigram features achieve the best results. Thus, we follow such settings in all UA models while leaving room for further

⁵We retrieved 487,088 samples among the 500,000 tweet IDs provided by the task organizers.

⁶Some users have more than one instance in the test set.

⁷to approximately 23 tweets in the case of this paper

| Model | Test-F | Test-R | Test-N |
|----------|--------------|--------------|--------------------|
| FastText | 31.45 | 30.98 | 31.24 |
| DA | 33.17 | 34.94 | 30.81 |
| IUA | 37.54 | 43.25 | 31.24 [†] |

†: We reuse M_b for tweets without the retrieved retrospective tweets for a given user.

Table 2: Macro-F [%] for the models on the test sets improvements of our system performance by introducing the features implemented in other leading systems.

To demonstrate the influence of retrieved user information, we compare our approaches on the test sets with the following settings:

- **Test-F** (*Full* set of 50,000 tweets) is the whole test set provided by the organizers of the SemEval2018 Task 2.
- **Test-R** (*Retrieved* set) is the subset of Test-F where the tweets are used to retrieve users’ retrospective tweets, containing 22,642 tweets.
- **Test-N** (*Non-retrieved* set) is the subtraction of Test-F and Test-R, where a user was not retrieved with this tweet or the retrieved user’s retrospective tweets were not available, containing 27,358 tweets.

4 Experimental Results

With additional retrospective data from the users, our model achieves more than 6 per cent better Macro-F than FastText. Consequently, it outperforms leading results from this competition.

Both DA and IUA achieve higher performance on the retrieved part of test set, Test-R, and thus improve the Macro-F on the full test set, Test-F (Table 4). This demonstrates the effectiveness of introducing the users’ retrospective tweets.

IUA outperforms DA, with a margin of more than 8 per cent on Test-R, indicating the necessity of training individual user adaptive models for the emoji prediction task. Compared with the best results in the task, on Test-F, — namely, 35.99 percent for *cagri* and 35.36 per cent for *cbaziotis* (Barbieri et al., 2018) — IUA achieves better results, even without an intensive feature engineering process.

5 Discussion

This section discusses and analyzes the success of our method in terms of its advantage on easily con-

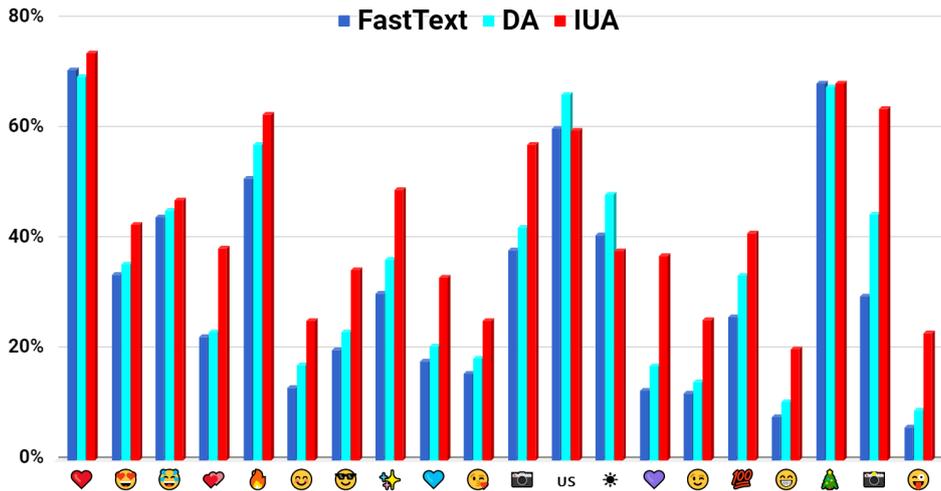


Figure 2: Class-specific results of the FastText, DA, and IUA models in emoji prediction on the Test-R data. We provide emoji labels on the x -axis and their respective label-specific F_1 score on the y -axis.

fused labels, and on users with certain amounts of historical tweets, over FastText.

We analyze the performance of IUA on different classes, by illustrating the Macro-F results of each emoji on Test-R in Figure 2. For the emojis, “Two Hearts”, “Blue Heart”, and “Purple Heart”, they carry similar meanings but different users have diverse preference when expressing their emotions. Both “Camera With Flash” and “Camera” without flash can be chosen under the same circumstances. Compared with DA, IUA achieves a marginal improvement on distinguishing the user preferences of those emojis. For other emojis such as “United States”, “Sun”, and “Christmas Tree”, IUA is competitive, as these emojis are aligned with single entities. These result show that our adaptive model is capable of learning user preferences in emojis with similar meanings, that is, the Case 1 of Section 1.

To demonstrate that IUA is also able to tackle the Case 2 of Section 1, we demonstrate some sample tweets that provide different emoji predictions using different user adapted models. For example, when the test tweet is *University life*, users have different attitudes towards “Red Heart”, “Two Hearts”, “Smiling Face With Smiling Eyes”, “Face With Tears of Joy”, “Hundred Points”, and other emojis Meanwhile, FastText is only able to predict “Two Hearts” for all users. IUA manages to capture the attitudes of individual users towards the same tweets, while FastText and DA tend to provide common attitudes of the tweets.

Both IUA and DA outperform FastText under

different scale settings of retrieved tweets, as illustrated in Figure 3. With more retrieved samples, the performance of DA increases. IUA reaches its peak performance on tweets with 64 retrieved historical tweets. More retrieved tweets do not further improve the results in our experiments. We have not observed much improvement for FastText for users with more retrieved tweets.

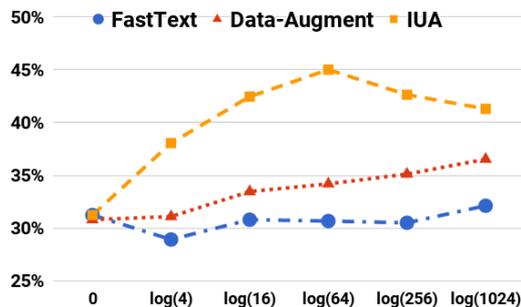


Figure 3: Macro-F of IUA, DA, and FastText on Test-R as numbers of tweets in each user collection increases.

6 Conclusion

This paper provides evidence for the rationality of enriching the original corpus longitudinally with user behaviors and transferring the lessons learned as user-adapted models to supervised machine learning tasks, such as the SemEval-2018 Task 2 on English emoji prediction. Our system achieves better performance than systems, which use all training data as a whole, even without much feature engineering. We believe this model can provide insight for introducing user-specific information for subsequent tasks of emoji prediction.

References

- Francesco Barbieri, Miguel Ballesteros, and Horacio Saggion. 2017. Are Emojis Predictable? In *Proceedings of the 15th European Chapter of the Association for Computational Linguistics (EACL)*.
- Francesco Barbieri, Jose Camacho-Collados, Francesco Ronzano, Luis Espinosa-Anke, Miguel Ballesteros, Valerio Basile, Viviana Patti, and Horacio Saggion. 2018. SemEval-2018 Task 2: Multilingual Emoji Prediction. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval)*.
- Dictionary.com. 2018. Define Emoji at Dictionary. <http://www.dictionary.com/browse/emoji>, Accessed: 2018-03-05.
- Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using Millions of Emoji Occurrences to Learn Any-domain Representations for Detecting Sentiment, Emotion and Sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th European Chapter of the Association for Computational Linguistics (EACL)*.
- Sunghwan Mac Kim, Qionikai Xu, Lizhen Qu, Stephen Wan, and Cécile Paris. 2017. Demographic Inference on Twitter using Recursive Neural Networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, volume 2, pages 471–477.
- Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *Computing Research Repository (CoRR)*, abs/1301.3781.
- Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 Task 4: Sentiment Analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518.
- Hanna Suominen, Tapio Pahikkala, and Tapio Salakoski. 2008. Critical Points in Assessing Learning Performance via Cross-validation. In *Proceedings of the 2nd International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR)*, pages 9–22.