Impact of Emojis on Automatic Analysis of Individual Emotion Categories

Ratchakrit Arreerard, Scott Piao
School of Computing and Communications
Lancaster University
UK
{r.arreerard,s.piao}@lancaster.ac.uk

Abstract

Automatic emotion analysis is a highly challenging task for Natural Language Processing, which has so far mainly relied on textual contents to determine the emotion of text. However, words are not the only media that carry emotional information. In social media, people also use emojis to convey their feelings. Recently, researchers have studied emotional aspects of emojis, and use emoji information to improve the emotion detection and classification, but many issues remain to be addressed. In this study, we examine the impact of emoji embedding on emotion classification and intensity prediction on four individual emotion categories, including *anger*, *fear*, *joy*, and *sadness*, in order to investigate how emojis affect the automatic analysis of individual emotion categories and intensity. We conducted a comparative study by testing five machine learning models with and without emoji embeddings involved. Our experiment demonstrates that emojis have varying impact on different emotion categories, and there is potential that emojis can be used to enhance emotion information processing.

1 Introduction

In this study, we investigate the issue of how emojis can impact on the automatic analysis of emotion in social media messages. This topic has been studied over past years, but further research is needed to fully understand the characteristics of the emojis and how they contribute to the conveyance of emotion. Automatic emotion analysis is a process of identifying emotions expressed by people. In social media, the emotions can be conveyed with various media including text, emojis, pictures, or other codes.

Because social media platforms impose little or no restriction on language usage in terms of grammar and formality, social media data contains a wide range of styles and forms, including informal, colloquial, slang, and ungrammatical expressions, mixed with emojis and other images. Such an unconstrained writing styles of social media messages present a tough challenge to the task of automatic emotion processing. As Hasan et al. (2019) pointed out, the casual style and semantic ambiguity of social media messages are the main two challenges in determining emotions in such data. To improve the automatic emotion analysis, researchers started to consider emojis as additional features. For example, word and emoji embedding are combined in the hope to generate better features for emotion classification. Emojis can contain emotion information that can help to identify emotions. However, as Barry et al. (2021) found, emojis are not always a good choice for representing emotion.

In this work, firstly we carried out experiments of emotion classification of four emotion categories and emotion intensity prediction using word embeddings as the sole features based on EmoInt dataset (Mohammad and Bravo-Marquez, 2017), and used the results as a benchmark. Then, we added emoji embeddings to the word embeddings to observe how the emoji information affects the performance of the emotion analysis. Our experiment results show that, overall, adding emoji embedding can marginally improve emotion analysis for some emotion categories. We foresee that emoji embedding can potentially improve the performance of emotion analysis further if we can design better methods of combining word and emoji embeddings.

2 Related Work

Recently, emojis have been used in automatic emotion analysis. For example, Wood and Ruder (2016) grouped commonly used emojis into six emotion categories, including *anger*, *disgust*, *fear*, *happi-*

Proceedings of Recent Advances in Natural Language Processing, pages 124–131
Varna, Sep 4–6, 2023
https://doi.org/10.26615/978-954-452-092-2_014

ness, *sadness*, and *surprise*. These emojis were used as emotion labels of messages for training emotion classification models. They also created a test data by manually annotating data. Their emotion classifiers trained on the emoji-labeled dataset produced a good performance on *joy* and *sadness*, but produced slightly lower performance on the other emotion categories.

Another application of emojis is to use them to train better word embeddings (Shoeb et al., 2019) to achieve a better emotion representation. The authors extracted a new word embedding by using Mikolov et al. (2013)'s *Word2vec* model as an intermediate representation. Firstly, they collected Twitter data to train a word2vec model. Then they created a new embedding model based on cosine similarity between words and emojis. They tested emotion intensity prediction by comparing EmoTag with well-known embedding models as benchmark, such as GloVe, and found EmoTag produced similar performances to that of the benchmark.

Eisner et al. (2016a) developed an emoji embedding model named Emoji2Vec, which was trained on emoji names and keyword phrases from the Unicode emoji list. They used Google News word2vec embeddings to formulate vectors and represent emojis from their describing phrases to train Emoji2Vec. Sentiment analysis task was used to evaluate the capability of Emoji2Vec, and the result showed that Emoji2Vec improves the overall performance of sentiment analysis.

Ahanin and Ismail (2020) proposed another pre-trained emoji embedding named FuzzyMoji2Vec. They compiled a list of commonly used emojis. Then these emojis were classified into one or more emotion classes based on the correlation between emojis and emotion labels. The embedding was trained on emojis and their emotion labels. Because the number of emojis in their dataset was limited, they extended the coverage of emojis using Fuzzy Clustering to classify unseen emojis collected from Twitter. The unseen emojis were clustered based on messages classified into 11 emotions. FuzzyMoji2Vec was reported to outperform Emoji2Vec in emotion classification.

More recently, Barry et al. (2021) developed the pre-trained emoji embedding Emojional. Emojional learned emoji embedding based on keywords representing emojis collected from the online emoji dictionaries of Emojipedia and EmojisWiki. They employed Google News Word2vec to create input vectors. Then they trained the embedding by predicting the corresponding emojis from the given inputs. They evaluated the Emojional in comparison with FuzzyMoji2Vec and Emoji2Vec. They showed Emojional was generally more accurate than state-of-the-art embeddings for the sentiment analysis task.

The past research shows that emoji embedding can improve the performance of emotion analysis. However, the most past works mainly reported on overall performances. It is necessary to gain a deeper understanding of the characteristics of emojis and about how emoji embedding can affect analysis of individual emotion categories. This paper examines the impact of emoji embedding on emotion classification and intensity prediction on four emotion categories *anger*, *fear*, *joy*, and *sadness*, which are included in the EmoInt annotation.

3 Experiment Setup

3.1 Dataset for Experiment

In this study, we used EmoInt as our experiment dataset. It is a collection of tweets in English, in which each tweet is tagged with an emotion label (*anger*, *fear*, *joy*, and *sadness*) and an emotion intensity value from the range of [0, 1]. These tweets are grouped into four sub-datasets of the aforementioned emotion categories.

Table 1 shows the structure of the dataset contents. As shown, there are slightly more *fear* messages than other categories. On the other hand, the average length (number of characters) of messages under different emotion categories are roughly the same, around 95 characters. Approximately 10% of tweets in the EmoInt contain at least one emoji. Also, the messages under each emotion category contain from 61 to 93 unique emojis.

We chose this dataset for our experiment, because it contains emojis, and its generally balanced emotion category structure and manual emotion annotation, which closely match the aim of this study. Particularly, the manual annotation of emotion intensity provides very useful information for our study.

3.2 Machine Learning Model

Because our focus of this study is to assess the impact of emoji embedding on emotion classification and intensity prediction, we selected five commonly used machine learning models, including Support Vector Machine (SVM), Support Vec-

Features	Train data				Test data			
	Anger	Fear	Joy	Sadness	Anger	Fear	Joy	Sadness
Total Sentences	857	1,147	823	786	760	995	714	673
Avg. Sent. Length	91.75	97.47	94.26	96.42	94.82	96.04	93.84	95.61
Sent. with emojis	100	127	91	79	108	122	115	77
Sent. without emojis	757	1020	732	707	652	873	599	596
Total emojis	234	204	190	216	216	220	263	128
Total unique emojis	64	78	78	74	64	80	93	61

Table 1: The statistics of EmoInt dataset contents.

tor Regression (SVR), Linear Regression, Logistic Regression, and Bi-directional Long Short Term Memory (Bi-LSTM).

In further detail, we chose SVM and SVR for emotion classification and intensity prediction respectively. Similarly, we chose Logistic Regression and Linear Regression for the classification and intensity prediction. We also selected Bi-LSTM to perform both tasks.

Figure 1 illustrates the workflow of Bi-LSTM. The left workflow is used when emojis are not considered, and the right workflow is used when emojis are considered. Emoji input will be concatenated with the output from the Bi-LSTM.

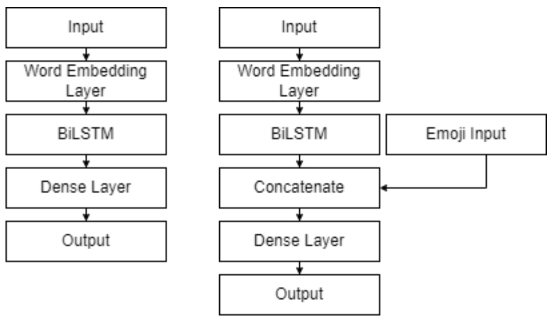

Figure 1: Bi-LSTM model for word embedding only (left) and Bi-LSTM model for word and emoji embedding (right).

We select a linear kernel for SVM and SVR. As for Bi-LSTM, we freeze the embedding layer to prevent it from adjusting weights. The loss functions for emotion classification and emotion intensity prediction are Binary Cross-Entropy and Mean Square Error respectively. As for activation functions, SoftMax and Sigmoid are used in emotion classification and emotion intensity prediction respectively.

For SVM, SVR, and Logistic Regression, *scikit-learn* (Pedregosa et al., 2011) software library was used; for Linear Regression, *statsmodels* library (Seabold and Perktold, 2010) was used; for Bi-LSTM, it was implemented using TensorFlow (Abadi et al., 2015) library.

3.3 Feature Selection

With regrades to embedding, we selected three pre-trained word embeddings: fastText(Mikolov et al., 2018a,b), GloVe(Pennington et al., 2014b,a), and BERT(Devlin et al., 2018b,a). For embedding, we selected Emoji2vec(Eisner et al., 2016b) and Emojional(Barry et al.).

As mentioned in the previous section, we selected five machine learning models with different types of inputs. For SVM, SVR, Logistic Regression, and Linear Regression, we use averaged word embedding vectors as input. Firstly, we sum the embedding vectors of the words in each tweet. Then each element value of the summed vector is divided by the number of words to generate a new vector to represent the whole tweet.

Regarding Bi-LSTM, we create word index vector and use it as input. The word index vector is created by transforming each word in the text of the tweet into an index number according to the embedding model used. This index is mapped to the embedding vector in the second layer of Bi-LSTM as shown in Figure 1.

As for emoji embedding, we use the averaging of emoji embeddings as input. Again, we sum the embedding vectors of individual emojis appear in a tweet. Then each element value of the summed vector is divided by the number of emojis present in the tweet to generate a new emoji embedding vector for the tweet.

When we combine word and emoji embeddings of a tweet for SVM, SVR, Logistic Regression, and Linear Regression, the averaged emoji embedding vector is concatenated to the counterpart averaged word embedding. For Bi-LSTM, the averaged emoji embedding vector is concatenated to the output of the third layer of Bi-LSTM, as illustrated in Figure 1.

3.4 Evaluation

We used Pearson correlation coefficient as the measurement for emotion intensity prediction, while the performance of emotion classifiers was evaluated using precision, recall, and F-measure. In our case, emotion classification is a multi-class classification task with four emotion categories. Therefore, we measured the performance for each individual class as well as the overall performance of the classifiers. However, the numbers of tweets under different emotion categories in EmoInt are not exactly the same. Thus, when calculating the overall performance metrics, we considered the ratios of numbers of the tweets under each emotion category, as shown below:

$$Precision = \sum_e Precision_e \times ratio_e \quad (1)$$

$$Recall = \sum_e Recall_e \times ratio_e \quad (2)$$

$$F1 = \frac{2(Precision \times Recall)}{(Precision + Recall)} \quad (3)$$

where,

$\sum_e ratio_e = 1$

$Precision_e$ = precision of emotion e

$Recall_e$ = recall of emotion e

4 Experiment

4.1 Emotion Classification

4.1.1 Word Embeddings as Sole Features

In the first phase of this experiment, we used only word embeddings as features for emotion classification, including BERT, fastText and GloVe. With regards to classifiers, we tested SVM, Logistic Regression and Bi-LSTM. This part of experiment aims to test the efficiency of word embeddings for emotion classification and to create a benchmark for comparing the performance of emotion classification when emoji embeddings are added as additional features.

Tables 2, 3 and 4 show precision, recall and F-measure of emotion classification for each emotion obtained with BERT, fastText, and GloVe separately.

Table 2 shows that Bi-LSTM outperformed SVM and Logistic Regression when using BERT as a feature. It achieved 65.23% precision, 64.42% recall, and 0.648 F-measure. The classifiers effectively identified tweets under *joy* with F-measure ranging from 0.597 to 0.746, but struggled to identify tweets under *sadness* with F-measure ranging from 0.477 to 0.557.

Table 3 shows, when fastText was used, Bi-LSTM also outperformed SVM and Logistic Regression. It produced 68.99% precision, 68.91% recall, and 0.69 F-measure. The classifiers performed best when identifying tweets related to *joy*, with F-measures of 0.608-0.745. However, it performed poorly for identifying tweets related to *sadness*, with F-measure ranging 0.464-0.623.

Table 4 shows the results obtained using GloVe as feature. Again, Bi-LSTM outperformed the other two classifiers. It achieved 80.44% precision, 80.30% recall, and 0.804 F-measure. The classifiers were effective in detecting tweets under *joy* category, with F-measure ranging 0.722-0.866. However, it performed poorly when detecting tweets under *sadness* category, with F-measure ranging 0.612-0.768.

The above results reveal that Bi-LSTM is the most effective classifier, and GloVe provides the most effective features. All classifiers performed well in identifying *joy* tweets, but they struggled in recognising *sadness* tweets.

4.1.2 Combining Emoji and Word Embeddings

In the second phase of the experiment, we combined word and emoji embeddings for emotion classification. With respect of emoji embedding, we tested Emoji2Vec and Emojional. Regarding classifiers, we tested three classifiers of SVM, Logistic Regression and Bi-LSTM. This part of experiment aims to test the impact of emoji embeddings on emotion classification by using the results obtained with word embeddings (see Tables 2, 3 and 4) as the benchmark.

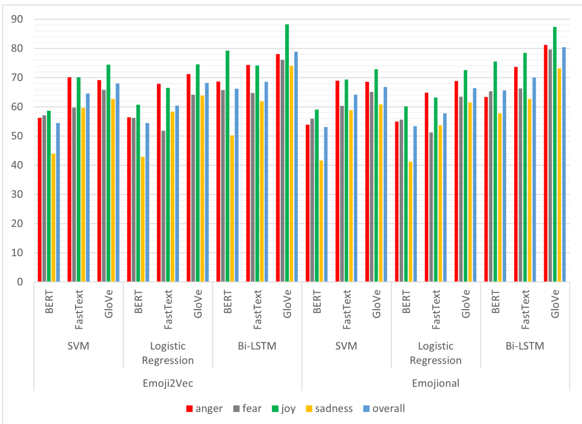

Figure 2: Precision (%) of emotion classification using word and emoji embeddings as features.

In detail, we first created word embedding features and emoji embedding features for each tweet, using the method discussed in Section 3.3. Then we concatenated each of three word embeddings (BERT, fastText and GloVe) with each of two emoji embeddings (Emoji2Vec and Emojional), obtain-

	Classifier	Metric	Anger	Fear	Joy	Sadness	Overall
BERT	SVM	Pre. (%)	58.04	56.85	58.96	44.03	54.87
		Rec. (%)	53.68	52.16	60.36	52.01	54.36
		F1	0.558	0.544	0.597	0.477	0.546
	Logistic Regression	Pre. (%)	56.23	56.09	60.78	43.56	54.51
		Rec. (%)	49.87	50.45	61.20	55.27	53.79
		F1	0.529	0.531	0.610	0.487	0.541
	Bi-LSTM	Pre. (%)	**70.09**	**63.60**	**75.57**	**51.18**	**65.23**
		Rec. (%)	**60.13**	**63.22**	**73.67**	**61.22**	**64.42**
		F1	**0.647**	**0.634**	**0.746**	**0.557**	**0.648**

Table 2: Performance of emotion classification using BERT as feature.

	Classifier	Metric	Anger	Fear	Joy	Sadness	Overall
fastText	SVM	Pre. (%)	**72.28**	60.82	71.86	60.72	66.08
		Rec. (%)	62.11	75.68	67.23	52.60	65.53
		F1	0.668	0.674	0.695	0.564	0.658
	Logistic Regression	Pre. (%)	68.24	51.43	66.56	59.22	60.60
		Rec. (%)	51.45	**79.30**	56.02	38.19	58.47
		F1	0.587	0.624	0.608	0.464	0.595
	Bi-LSTM	Pre. (%)	67.35	**66.25**	**74.58**	**68.95**	**68.99**
		Rec. (%)	**73.55**	69.65	**74.37**	**56.76**	**68.91**
		F1	**0.703**	**0.679**	**0.745**	**0.623**	**0.690**

Table 3: Performance of emotion classification using fastText as feature.

ing six new embedding vectors for each tweet. In this way, for the tweets of EmoInt, we created six sets of feature vectors, which were passed to the classifiers for emotion classification. Figures 2, 3, and 4 show the precision, recall, and F-measure of the emotion classification respectively.

As shown in Figure 2, Bi-LSTM with GloVe+Emojional achieved the best overall precision of 80.43%. In terms of individual emotions, all classifiers except Logistic Regression (with fastText+Emojional and fastText+Emoji2Vec) yielded the best precision for detecting *joy* tweets compared to other emotion categories, and the best precision (88.29%) was produced by Bi-LSTM with GloVe+Emoji2Vec.

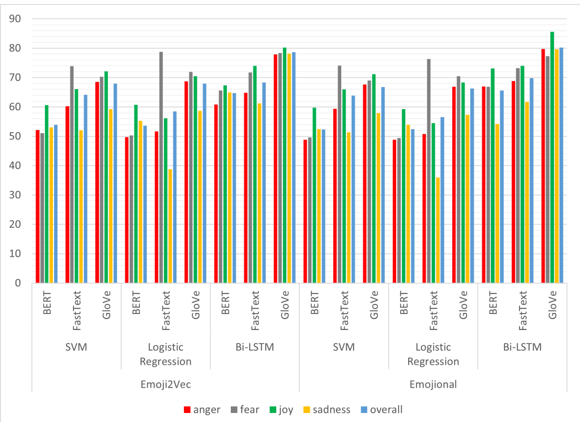

Figure 3: Recall (%) of emotion classification using word and emoji embeddings as features.

Figure 3 reveals that Bi-LSTM with GloVe+Emojional yielded the best overall recall of 80.27%. Regarding individual emotions, twelve and six classifiers produced the best recalls for the *joy* and *fear* categories respectively. Again, Bi-LSTM with GloVe+Emojional achieved the best recall of 85.57% for classifying *joy* tweets.

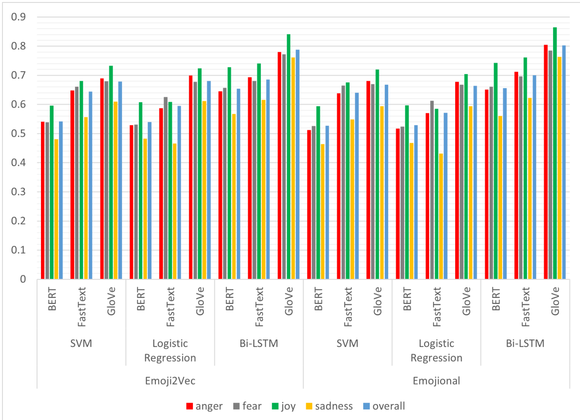

Figure 4: F-measure of emotion classification using word and emoji embeddings as features.

Figure 4 shows that Bi-LSTM with GloVe+Emojional produced the best F-measure of 0.803. As for individual emotions, all classifiers except Logistic Regression (with fastText+Emojional and fastText+Emoji2Vec) achieved the best F-measures for detecting *joy* compared to other emotion categories, and Bi-LSTM with GloVe+Emojional yielded the best F-measure of 0.865.

The experiment results reveal that Bi-LSTM with the combination of Emojional with either BERT or fastText can improve overall F-measure by up to 0.010. The best performance of emotion classification was obtained by using Bi-LSTM with GloVe+Emojional embedding vectors. But emoji embeddings do not always improve emo-

	Classifier	Metric	Anger	Fear	Joy	Sadness	Overall
GloVe	SVM	Pre. (%)	69.36	65.88	74.46	63.03	68.06
		Rec. (%)	68.82	70.05	71.85	60.03	68.01
		F1	0.691	0.679	0.731	0.615	0.680
	Logistic Regression	Pre. (%)	71.16	64.15	75.08	63.77	68.24
		Rec. (%)	68.82	72.46	69.61	58.84	68.01
		F1	0.700	0.680	0.722	0.612	0.681
	Bi-LSTM	Pre. (%)	**79.92**	**76.92**	**88.69**	**77.49**	**80.44**
		Rec. (%)	**78.03**	**81.71**	**84.59**	**76.23**	**80.30**
		F1	**0.790**	**0.792**	**0.866**	**0.768**	**0.804**

Table 4: Performance of emotion classification using GloVe as feature.

tion classification. For example, in our experiment, Emojional slightly degraded the classification result when it was added to GloVe for Bi-LSTM classifier.

4.2 Emotion Intensity Prediction

4.2.1 Intensity Prediction with Word Embedding

As mentioned earlier, one of our main aims of this study is to test how emoji embeddings can impact on emotion intensity prediction. For this purpose, we needed to create a benchmark for comparison, by involving only word embeddings. We followed similar process as that of emotion classification mentioned in section 4.1.1, only using word embeddings for emotion intensity prediction, including BERT, fastText and GloVe. For each of the three embeddings, we tested three prediction models of SVR, Linear Regression and Bi-LSTM. We used Pearson correlation coefficient to compare the automatic emotion intensity prediction results against the manual annotation in the EmoInt as gold standard.

Table 5 presents the evaluation results for BERT, fastText and GloVe. In the table, the codes *SVR*, *LR*, *BI* refer to Support Vector Regression, Linear Regression, and Bi-LSTM respectively. In addition, the codes *A*, *J*, *F* and *S* refer to *anger*, *joy*, *fear*, and *sadness*. An additional code *M* is used to refer to *mean coefficient score*. (Same codes are used for Tables 6 and 7)

As shown in the table, Bi-LSTM with GloVe achieved the highest overall performance, with 0.47 coefficient. In terms of individual emotions, all prediction models were relatively effective in predicting intensity value for *fear* and *sadness*, with coefficients ranging 0.38-0.54 and 0.36-0.58 respectively. On the other hand, all prediction models yielded the lowest performance in predicting *joy* intensity, with coefficients ranging from 0.13 to 0.35.

4.2.2 Intensity Prediction by Combining Emoji and Word Embeddings

Based on the experiment discussed in the previous section, we combined word and emoji embeddings (Emoji2Vec and Emojional) for extended features, following the same twitter embedding vector creation process mentioned in section 4.1.2. Then we applied three prediction models, SVR, Linear Regression, and Bi-LSTM on the feature vectors of tweets in the EmoInt.

This part of experiment aims to test the efficacy of emoji embedding on emotion intensity prediction, with the results obtained with only word embedding (see Table 5) as the benchmark. Tables 6 and 7 show the evaluation results for Emoji2Vec and Emojional respectively.

As shown in Table 6, Bi-LSTM with Emoji2Vec+GloVe yielded the highest overall Pearson correlation coefficient of 0.47. When it comes to individual emotions, all prediction models are relatively effective in predicting *fear* and *sadness* intensity values compared to *anger* and *joy*, with coefficients ranging between 0.38-0.55 and 0.37-0.57 respectively. On the other hand, all prediction models produced the lowest coefficients in predicting *joy* intensity compared to other categories, ranging from 0.10 to 0.36.

Table 7 shows that Bi-LSTM with Emojional+GloVe produced the highest overall coefficient of 0.48. Regarding individual emotions, all prediction models effectively predicted intensity values of *fear* and *sadness* compared to other emotion categories, with coefficients ranging from 0.38-0.56 and 0.38-0.58 respectively. On the other hand, all prediction models produced the lowest performance for *joy*, with coefficients ranging from 0.10 to 0.31.

As shown in our evaluation results, adding emoji embedding has improved the ability to predict intensity level of *anger*, *fear* and *sadness*. Before emoji embeddings are added, the coefficients of emotion intensity prediction range from 0.30-0.47 for *anger*, 0.38-0.54 for *fear*, and 0.36-0.58 for

		A	F	J	S	M
BERT	SVR	0.34	0.45	0.29	0.46	0.38
	LR	0.34	0.45	0.29	0.46	0.38
	Bi	0.40	**0.54**	**0.35**	0.49	0.45
fastText	SVR	0.36	0.44	0.17	0.50	0.37
	LR	0.31	*0.38*	*0.13*	0.40	*0.31*
	Bi	0.36	0.45	0.18	*0.36*	0.34
GloVe	SVR	0.34	0.44	0.28	0.52	0.40
	LR	*0.30*	0.42	0.29	0.47	0.38
	Bi	**0.47**	0.51	0.33	**0.58**	**0.47**

Table 5: Evaluation statistics of emotion intensity prediction with only word embeddings.

		Emoji2Vec				
		A	F	J	S	M
BERT	SVR	0.34	0.46	0.26	0.48	0.39
	LR	0.27	0.44	0.26	0.40	0.34
	Bi	0.40	**0.55**	**0.36**	0.51	0.46
fastText	SVR	0.35	0.45	*0.10*	0.51	0.36
	LR	*0.26*	*0.38*	*0.10*	*0.37*	*0.28*
	Bi	0.40	0.43	0.11	0.38	0.33
GloVe	SVR	0.34	0.45	0.22	0.53	0.39
	LR	0.28	0.42	0.25	0.42	0.35
	Bi	**0.49**	0.52	0.31	**0.57**	**0.47**

Table 6: Evaluation statistics of emotion intensity prediction with combination of word and Emoji2Vec embeddings.

sadness. After adding emoji embedding, the coefficients for these categories are marginally increased by up to 0.03. On the other hand, emoji embedding slightly degraded performance in predicting intensity of *joy*. Such a result indicates that emojis can generally be helpful in conveying intensity level of *anger*, *fear* and *sadness*, but they may be less relevant to intensity level of *joy*.

Our experiment showed that classifiers are less effective for *sadness* compared to other categories. We checked emotion words in each sub-dataset of EmoInt by looking up the NRC Emotion lexicon (Mohammad and Turney, 2013). We found that *anger* words are more likely to appear in *anger* messages, and similar case for *fear* and *joy* words. On the other hand, We found similar numbers of *anger*, *fear*, *joy*, and *sadness* words appear in the *sadness* sub-dataset. We speculate such an even distribution of emotion words in the *sadness* sub-dataset can be the cause of the difficulty of detecting *sadness* messages.

Regarding emotion intensity prediction, we found that the intensity prediction models performed poorly for *joy* compared to other categories. We checked some samples from the *joy* sub-dataset and observed that some emojis with opposite emotions co-occurred within same tweets, such as ”U+1F602” (face with tears of joy) and ”U+1F62D” (loudly crying face). In addition, emojis of *joy* appeared in messages classified under other categories. This may have caused the difficulty of predicting emotion intensity for *joy*.

		Emojional				
		A	F	J	S	M
BERT	SVR	0.32	0.45	0.24	0.47	0.37
	LR	0.27	0.44	0.25	0.40	0.34
	Bi	0.41	**0.56**	**0.31**	0.54	0.46
fastText	SVR	0.26	0.42	0.12	0.45	0.32
	LR	*0.21*	*0.38*	*0.10*	*0.38*	*0.27*
	Bi	0.40	0.46	0.12	*0.38*	0.34
GloVe	SVR	0.29	0.43	0.24	0.51	0.37
	LR	0.23	0.41	0.25	0.45	0.34
	Bi	**0.50**	0.53	**0.31**	**0.58**	**0.48**

Table 7: Evaluation statistics of emotion intensity prediction with combination of word and Emojional embeddings.

5 Conclusion

In this paper, we reported our study which aims to study the impact of emoji embeddings on emotion classification and intensity prediction in social media messages, using the EmoInt as our training and test dataset. We examined the performance of five machine learning models with all possible combinations between a set of three word embeddings (fastText, GloVe, BERT) and two emoji embeddings (Emoji2Vec and Emojional). We compared the results obtained with and without emoji embeddings to assess the impact of emoji embedding on analysing individual emotion categories. Because the EmoInt dataset only contains annotation of four emotion categories (*joy*, *anger*, *fear* and *sadness*), our study focused on these categories.

In our experiment, we tested 18 different combinations of {classifier + word_embedding + emoji_embedding}. We observed improvement on emotion classification for *fear* in six cases, for *joy* in five cases, and *anger* and *sadness* in four cases. As for emotion intensity prediction, the improvements was observed for *fear* in eight cases, *sadness* in seven cases, *anger* in four cases, and *joy* in one case. Therefore, it is a mixed picture how emojis can improve the automatic emotion analysis.

We acknowledge our results are not conclusive, as we used simple embedding combination methods, and only a small portion of tweets in EmoInt contain emojis, making it difficult to examine the impact of emoji embeddings in further details. For future work, we aim to explore larger emoji embedding datasets and more embedding combination techniques.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Zahra Ahanin and Maizatul Akmar Ismail. 2020. Feature extraction based on fuzzy clustering and emoji embeddings for emotion classification. *International Journal of Technology Management and Information System*, 2(1):102–112.

Elena Barry, Shoaib Jameel, and Haider Raza. Emojional embedding model. Downloadable at: `https://github.com/elenabarry/emojional`.

Elena Barry, Shoaib Jameel, and Haider Raza. 2021. Emojional: Emoji embeddings. In *UK Workshop on Computational Intelligence*, pages 312–324. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018a. Bert-base-uncased mbedding model. Downloadable at: `https://huggingface.co/bert-base-uncased`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018b. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bosnjak, and Sebastian Riedel. 2016a. emoji2vec: Learning emoji representations from their description. *CoRR*, abs/1609.08359.

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. 2016b. Emoji2vec embedding model. Downloadable at: `https://github.com/uclnlp/emoji2vec`.

Maryam Hasan, Elke Rundensteiner, and Emmanuel Agu. 2019. Automatic emotion detection in text streams by analyzing twitter data. *International Journal of Data Science and Analytics*, 7(1):35–51.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018a. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018b. fastText embedding model. Downloadable at: `https://fasttext.cc/`.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Saif M. Mohammad and Felipe Bravo-Marquez. 2017. Emotion intensities in tweets. In *Proceedings of the sixth joint conference on lexical and computational semantics (*Sem)*, Vancouver, Canada.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014a. Glove embedding model. Downloadable at: `https://nlp.stanford.edu/projects/glove/`.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014b. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Skipper Seabold and Josef Perktold. 2010. statsmodels: Econometric and statistical modeling with python. In *9th Python in Science Conference*.

Abu Awal Md Shoeb, Shahab Raji, and Gerard de Melo. 2019. EmoTag – towards an emotion-based analysis of emojis. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1094–1103, Varna, Bulgaria. INCOMA Ltd.

Ian Wood and Sebastian Ruder. 2016. Emoji as emotion tags for tweets. In *Proceedings of the Emotion and Sentiment Analysis Workshop LREC2016, Portorož, Slovenia*, pages 76–79. sn.