大規模言語モデルを用いたマイクロブログに対する絵文字予測

花一傑 [1]　宇津呂武仁 [2]　陳嘉敏 [3]　鈴木良弥 [1]
[1] 山梨大学工学部メカトロニクス工学科　[2] 筑波大学システム情報系知能機能工学域
[3]IIJ 技術研究所

概要

昨今の SNS では，ユーザーが絵文字を自分の文章に付与することにより，文章の表す感情を増幅，補完することが多く，その絵文字付き日本語文章を利用した感情分析に関する研究も増えている．しかし，英語文章そのものの意味を理解し，その文章に相応しい絵文字を複数のクラスの中から予測し，正しいクラスに分類する絵文字予測タスク [1] [2] [3] は存在するが，日本語文章に対する絵文字予測を目的とした研究は存在しない．そこで本研究では，大規模言語モデル ChatGPT[1)] を利用し，マイクロブログ (X) の日本語ポスト文に対する絵文字予測を行い，ChatGPT の fine-tuning で高い精度を達成した．

1　はじめに

昨今の SNS では，ユーザーが絵文字を自分の文章に付与することにより，文章の表す感情を増幅，補完することが多い．絵文字はテキスト処理において無視できない存在になりつつある．多くの研究は絵文字を利用し，文章の意味を理解しようとしている．例えば，Golazizian ら [4] の研究では，絵文字予測モデルを転移学習として利用し，アイロニー検出の精度の向上を達成している．

一方，絵文字がない文章からその文章に最も相応しい絵文字を複数のクラスの中から予測し，正しいクラスに分類する絵文字予測タスクも存在する．絵文字予測タスクは，SNS 上のポスト文への理解や意味分析において重要なタスクである．特に ChatGPT のような大規模言語モデルの分類タスクにおいては，従来の分類モデルと異なり，文章の意味だけでなく，その絵文字の意味や使い方も理解し，文章と絵文字を対応させる必要がある．

しかし，これまでの絵文字予測は全て英語を対象としており，日本語を対象とした絵文字予測に関する研究は存在しない．また，これらの関連研究では，ラベルとしての絵文字の妥当性を考慮していない．絵文字には使い方や意味が似ているものが存在する．従って，絵文字予測はこれまでの分類タスクと違い，予測を行う前にデータセット内の絵文字を適切な絵文字グループに統合する必要がある．そこで本研究では，マイクロブログ (X) の日本語ポストを対象とし，グループ化した絵文字クラスを用い，大規模言語モデル ChatGPT 及び日本語 BERT [2)] で絵文字予測を行い，その結果についての分析を行う．

2　関連研究

絵文字予測は自然言語処理の様々なタスクに貢献できる．Felbo ら [5] は，絵文字予測モデルで感情分析，感情認識，皮肉検出のベンチマークにおいて，最先端の精度を達成している．Barbieri ら [1] の研究では，SemEval 2018 Task 2[2] のデータと改良した Felbo ら [5] のモデルを利用し，ラベルごとの attention 付き LSTM による絵文字予測を行い，絵文字予測の精度と絵文字・各単語間の関連の強さを調べている．この手法では，使用頻度の少ない絵文字の予測精度の向上，絵文字と各単語間 attention の可視化を達成している．また，Lee ら [3] の研究では，感情分類を補助タスクとした絵文字予測をマルチタスク学習で行っている．

絵文字予測タスクにおいては，適切にアノテーションされているデータセットが存在しないことが問題となっている．そのため，Ma ら [6] の研究では，マルチクラスとマルチラベルのアノテーションを機械学習で行い，絵文字予測のためのデータセットを作成している．

3　マイクロブログ・データセット

マイクロブログの前処理をする上で二つの技術情報 (絵文字ステータス[3)] と絵文字バージョン) がある．特に絵文字バージョンは ChatGPT による絵文

1）https://platform.openai.com/docs/models/gpt-3-5

2）https://github.com/cl-tohoku/bert-japanese/

3）https://www.unicode.org/reports/tr51/tr51-25.html

表 1 前処理の各手順後のポスト数

手順	データ数
1	5,568,951
2	1,234,431
3	1,192,752
4	592,546
5	315,746

表 2 M_{20} 内の絵文字及びその出現回数

絵文字	出現回数	絵文字	出現回数
😊	38,210	💦	12,896
✨	32,244	🥰	12,234
😭	23,972	🤔	11,631
☺️	23,423	👍	9,525
☀️	21,869	😇	9,336
😂	19,120	😃	8,210
😍	18,641	💕	7,841
🤣	17,552	🥲	7,760
😅	13,461	🎉	7,597
😆	13,180	😁	7,044

字予測タスクにおいて，重要なフィルタとなる．

絵文字は，fully-qualified，minimally-qualified，unqualified の三つのステータスに分けられている．fully-qualified の絵文字が minimally-qualified と unqualified に含まれる場合，これらは同等の絵文字であるため，minimally-qualified と unqualified の絵文字を fully-qualified に置き換える．

絵文字には複数の絵文字バージョンが存在し，OpenAI 社のモデル gpt-3.5-turbo-1106 は v13.1 の絵文字までしか正しく認識できず，gpt-4-1106-preview は v14.0 の絵文字までしか正しく認識できない．したがって，本研究では，gpt-3.5-turbo-1106 に準じ，v13.1 を認識の上限とする．

上記の絵文字の技術情報を考慮し，データセットの前処理を行う．本研究では，2023 年 1 月までのマイクロブログ (X) の日本語ポストを収集し，以下の手順で前処理を行う．

1. url およびユーザーへの@を削除し，unqualified と minimally-qualified の絵文字を fully-qualified に置き換える．
2. 本研究では，絵文字予測タスクをマルチクラス分類タスクとして扱うため，ポスト文に 1 種類の絵文字しか使われていないものを抽出する．
3. 前述した ChatGPT モデルの絵文字の認識上限 v13.1 以下の絵文字付きポストだけ抽出する．
4. 本研究では，全文の意味を理解し，絵文字を予測するため，絵文字が文の末尾にあるデータだけ抽出する．
5. 上記の手順で抽出したデータセットの中で出現回数上位 20 位の絵文字つきポストを抽出する．

表 3 絵文字グループと置き換え後のクラス

絵文字グループ	置き換え後クラス
😊 ☺️ 😃 😆 😄 👍 ✨ ☀️	😊
🤣 😂 😅	🤣
😭 🥲 🙄	😭
🎉	🎉
🥰 💕	🥰
😇	😇
💦	💦
🤔	🤔

前処理の各手順により抽出されたデータ数を表 1 に示す．前処理で得られたデータ (約 31 万の絵文字つきポスト) を実験で使うマイクロブログ (X) データセット M_{20} とする．

4 予測対象の絵文字

4.1 予測対象の絵文字の選定

3 節で作成された M_{20} 内の絵文字及びその出現回数を表 2 に示す．従来のマルチクラス分類タスクにおいては，分類の対象となる文とクラスは一意に定まる．しかし，😊と☺️のように，絵文字には意味や使い方が似ているものが存在するため，一意に定まらない場合が多い．そこで，似た絵文字に対し，文とクラスが一意に定まるように絵文字のグループ化を行う．表 2 の絵文字に対し，表 3 のように絵文字をグループ化し，代表の絵文字で置き換える．本研究では，絵文字予測の精度を向上させるために，できるだけ多くの絵文字を第一著者の主観でグループ化している．置き換え前のクラス数は 20 であったのに対し，置き換え後のクラス数は 8 である．置き換え後のマイクロブログデータセットを M_8[4] とする．また，6 節の結果から，😇，💦は予測が難しい絵文字と判断し，😇，💦付きポストを M_8 から削除したマイクロブログデータセット M_6[5] を作成する．

4) M_8 のデータ数は M_{20} と同様で 315,746 件．
5) M_6 のデータ数は 293,514 件．

表 4　ChatGPT 入力プロンプト

「絵文字の使用法解説なし」プロンプト	「絵文字の使用法解説あり」プロンプト
[実際予測するポストはここに入る] というツイートに最も相応しい絵文字を選択肢の中から一つ選んでください。なお、選んだ一つの絵文字だけ回答してください。選択肢：😊、🤣、😭、🎉、🥰、😇、💦、🤔	[実際予測するポストはここに入る] というツイートに最も相応しい絵文字を選択肢の中から一つ選んでください。なお、選んだ一つの絵文字だけ回答してください。ただし、🎉は「おめでとう」のようなお祝いの言葉の後にしか付けない、🥰は「好き」、「幸せ」、「素敵」を表現する、😇は「終わった」、「もうダメ」、「天使」に関連して使う、💦は「ストレス」、「緊張」、「汗」、「暑さ」に関連して使うという条件がある。選択肢：😊、🤣、😭、🎉、🥰、😇、💦、🤔

4.2　訓練・評価用データセット

人間にとって予測できるものを大規模言語モデルはどれほど予測できるかを検証するために，人手により絵文字予測可能なポストを抽出する．手順を以下に示す．

1. データセット M_8，もしくは M_6 の中から無作為に一つのポスト p を取り出し，そのテキストを t_p，絵文字を e_p とする．
2. 取り出したポスト p のテキスト t_p だけを見て，第一著者が t_p に最も相応しい絵文字 $\hat{e}$ を予測する．
3. $\hat{e} = e_p$ の場合，ポスト p を人手により絵文字予測可能なデータセット M_{8h}，もしくは M_{6h} に追加する．

具体的な訓練データ，評価データ等の作成手順を以下に示す．

開発・評価データ　M_{8h} および M_{6h} から無作為に 400 件 (開発データ) もしくは 360 件 (評価データ) を選定した集合を，「人手により絵文字予測可能なポストから構成される開発データセットおよび評価データセット」とする．開発データセットとしては，V_{8h} および V_{6h} を作成し，評価データセットとしては，T_{8h} および T_{6h} を作成する．また，人手により絵文字予測可能か否かに関わらず，M_8 および M_6 から無作為に 360 件のポストを収集したデータセットを，それぞれ，評価データセット T_{8M} および T_{6M} とする．さらに，8 クラスもしくは 6 クラスの絵文字の一様分布に従って 360 件のポストを収集したデータセットを，それぞれ，評価データセット T_{8u} および T_{6u} とする．

ChatGPT の訓練データ　M_{8h} から開発・評価データセットを削除した集合から，8 クラスの絵文字の一様分布に従い，無作為に 40 件，80 件，120 件，160 件を選定した集合を，それぞれ，R_8^{40}，R_8^{80}，R_8^{120}，R_8^{160} とする．

BERT の訓練データ　M_8 から開発・評価データセットを削除した集合 R_{8b} (314,928 件) を，BERT の fine-tuning 用訓練データとする．

モデルの訓練は 8 クラスのデータセットのみで行い，評価時のみ，8 クラス，6 クラスの両方を用いて行った．また，ChatGPT の訓練データ数の最適化および BERT 訓練時に用いる開発データとして，開発データ V_{8h}，V_{6h} を使用した．

5　絵文字予測手法

本研究では，モデルに OpenAI 社の gpt-3.5-turbo-1106，gpt-4-1106-preview，東北大学の cl-tohoku/bert-base-japanese-v3 を用いる．予測結果は正解率 (Acc) と F1 値で評価する．

5.1　ChatGPT

5.1.1　概要

4.2 節で作成された ChatGPT の各種データセットで zero-shot，few-shot，fine-tuning の精度を調べる．

Zero-shot では，評価データのテキストのみを gpt-3.5-turbo-1106，gpt-4-1106-preview に入力し，予測結果の Acc と F1 値を計算する．Few-shot では，R_8^{160} からクラスごとに 1，2 件のポストを抽出し，8-shot，16-shot の shot とする．shot と評価データを gpt-3.5-turbo-1106，gpt-4-1106-preview に入力し，予測結果の Acc と F1 値を計算する．fine-tuning では，

表 5　人手により絵文字予測可能な評価データ (T_{8h}，T_{6h}) に対する評価結果 (「絵文字の使用法解説あり・なしモデル」・「8 / 6 クラス」のクラスごとの最大 Acc / F1 を**太字**で示す．)

モデル	絵文字の使用法解説なし		絵文字の使用法解説あり	
	T_{8h}	T_{6h}	T_{8h}	T_{6h}
	Acc / F1	Acc / F1	Acc / F1	Acc / F1
gpt-3.5-turbo-1106 (zero-shot)	0.44 / 0.39	0.46 / 0.47	0.71 / 0.44	0.72 / 0.61
gpt-3.5-turbo-1106 (8-shot)	0.65 / 0.46	0.67 / 0.61	0.62 / 0.43	0.72 / 0.62
gpt-3.5-turbo-1106 (16-shot)	0.59 / 0.41	0.68 / 0.59	0.61 / 0.50	0.72 / 0.60
gpt-3.5-turbo-1106 (fine-tuning)	**0.78 / 0.62**	**0.83 / 0.76**	**0.80 / 0.63**	**0.83 / 0.75**
gpt-4-1106-preview (zero-shot)	0.53 / 0.46	0.63 / 0.57	0.64 / 0.47	0.74 / 0.67
gpt-4-1106-preview (8-shot)	0.65 / 0.49	0.72 / 0.63	0.64 / 0.49	0.73 / 0.65
gpt-4-1106-preview (16-shot)	0.68 / 0.51	0.74 / 0.65	0.66 / 0.50	0.76 / 0.68
cl-tohoku/bert-base-japanese-v3	0.76 / 0.48	0.80 / 0.64	—	

4.2 節で作成した訓練データで gpt-3.5-turbo-1106 の fine-tuning を行う．開発データで訓練データ数とエポック数の最適設定を探索する．fine-tuning 後，評価データのテキストのみを最適設定のモデルに入力し，予測結果の Acc と F1 値を計算する．

5.1.2　プロンプト

ChatGPT には日本語に対し，使い方や認識が間違っている絵文字がある．この問題を解決するために，絵文字の日本語においての一般的な使い方を ChatGPT に解説する．8 クラスの「絵文字の使用法解説なし」と「絵文字の使用法解説あり」のプロンプトを表 4 に示す．6 クラスの場合，選択肢の中から 😇と 💦(及びそれらの使用法解説) を削除する．「絵文字の使用法解説なし」と「絵文字の使用法解説あり」の実験は同様に 5.1.1 節で述べた方法で行う．

5.2　BERT

4.2 節で作成された R_{8b} で fine-tuning を行う．開発データでバッチサイズ，学習率，エポック数の最適設定を探索する．fine-tuning 後，評価データのテキストのみを最適設定のモデルに入力し，予測結果の Acc と F1 値を計算する．

6　評価

各モデルの T_{8h}，T_{6h} に対する予測結果を表 5 に示す．T_{8M}，T_{6H}，T_{8u}，T_{6u} に対する予測結果を付録の表 6，表 7 に示す．

fine-tuning された gpt-3.5-turbo-1106[6] では T_{8h} においての Acc，F1 スコアは 0.80，0.63 であったのに対し，BERT では 0.76，0.48 であった．また，fine-tuning された gpt-3.5-turbo-1106 では T_{6h} においての Acc，F1 スコアは 0.83，0.76 であったのに対し，BERT[7] では 0.80，0.64 であった．fine-tuning された gpt-3.5-turbo-1106 では全ての ChatGPT モデルの設定かつ BERT を上回る精度を達成している．

gpt-3.5-turbo-1106 の fine-tuning では，「絵文字の使用法解説なし」と「絵文字の使用法解説あり」でほぼ同様な精度が得られたが，「絵文字の使用法解説なし」には 160 件の訓練データが使われているのに対し，「絵文字の使用法解説あり」の方には 80 件の訓練データが使われている．これは絵文字の使い方の解説が 80 件の訓練データと同じようにモデルに作用していることを意味している．すなわち，ChatGPT の fine-tuning に絵文字の使い方の解説をプロンプトに入力することで，「絵文字の使用法解説なし」より少ない訓練データで同様な精度が得られる．

7　おわりに

本研究では、大規模言語モデルによる日本語絵文字予測の精度を調べた．絵文字のグループ化により，クラス数を減少させ，人手により絵文字予測可能なデータを選出することで高い予測精度を達成した．また，大規模言語モデルの fine-tuning に絵文字の使い方の解説をプロンプトに入力することで，「絵文字の使用法解説なし」より少ない訓練データで同様な精度が得られることが明らかになった．

6)　「絵文字の使用法解説なし」に R_8^{160}，エポック数 3，「絵文字の使用法解説あり」に R_8^{80}，エポック数 3 を設定している．

7)　ハイパーパラメータにバッチサイズ 32，エポック数 2，学習率 1e-4 を設定している．

参考文献

[1] F. Barbieri, L. Espinosa-Anke, J. Camacho-Collados, S. Schockaert, and H. Saggion. Interpretable emoji prediction via label-wise attention LSTMs. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 4766–4771, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[2] F. Barbieri, J. Camacho-Collados, F. Ronzano, L. Espinosa-Anke, M. Ballesteros, V. Basile, V. Patti, and H. Saggion. SemEval 2018 task 2: Multilingual emoji prediction. In **Proceedings of the 12th International Workshop on Semantic Evaluation**, pp. 24–33, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[3] S. Lee, D. Jeong, and E. Park. MultiEmo: Multi-task framework for emoji prediction. **Knowledge-Based Systems**, Vol. 242, p. 108437, 2022.

[4] P. Golazizian, B. Sabeti, S. Ashrafi Asli, Z. Majdabadi, O. Momenzadeh, and R. Fahmi. Irony detection in Persian language: A transfer learning approach using emoji prediction. In **Proceedings of the Twelfth Language Resources and Evaluation Conference**, pp. 2839–2845, Marseille, France, May 2020. European Language Resources Association.

[5] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann. 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**, pp. 1615–1625, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[6] W. Ma, R. Liu, L. Wang, and S. Vosoughi. Multi-resolution annotations for emoji prediction. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing**, pp. 6684–6694, Online, November 2020. Association for Computational Linguistics.

A　評価結果の詳細

「M_8 における 8 クラスの分布に従った評価データに対する評価結果」および「8 クラスが一様分布に従った評価データ (T_{8u}，T_{6u}) に対する評価結果」を，それぞれ，表 6 および表 7 に示す．

表 6　マイクロブログデータセット M_8 における 8 クラスの分布に従った評価データ (T_{8M}，T_{6M}) に対する評価結果 (「絵文字の使用法解説あり・なしモデル」・「8 / 6 クラス」のクラスごとの最大 Acc / F1 を**太字**で示す．)

モデル	絵文字の使用法解説なし		絵文字の使用法解説あり	
	T_{8M} Acc / F1	T_{6M} Acc / F1	T_{8M} Acc / F1	T_{6M} Acc / F1
gpt-3.5-turbo-1106 (zero-shot)	0.24 / 0.18	0.25 / 0.21	0.51 / 0.27	0.49 / 0.33
gpt-3.5-turbo-1106 (8-shot)	0.40 / 0.24	0.45 / 0.31	0.42 / 0.26	0.48 / 0.32
gpt-3.5-turbo-1106 (16-shot)	0.43 / 0.23	0.43 / 0.29	0.43 / 0.27	0.46 / 0.30
gpt-3.5-turbo-1106 (fine-tuning)	0.50 / 0.35	0.56 / **0.45**	**0.55** / **0.36**	**0.59** / **0.48**
gpt-4-1106-preview (zero-shot)	0.33 / 0.26	0.41 / 0.30	0.43 / 0.34	0.48 / 0.38
gpt-4-1106-preview (8-shot)	0.39 / 0.32	0.49 / 0.41	0.42 / 0.34	0.52 / 0.43
gpt-4-1106-preview (16-shot)	0.41 / 0.31	0.50 / 0.40	0.42 / 0.33	0.52 / 0.40
cl-tohoku/bert-base-japanese-v3	**0.58** / **0.35**	**0.61** / 0.42	—	

表 7　8 クラスが一様分布に従った評価データ (T_{8u}，T_{6u}) に対する評価結果 (「絵文字の使用法解説あり・なしモデル」・「8 / 6 クラス」のクラスごとの最大 Acc / F1 を**太字**で示す．)

モデル	絵文字の使用法解説なし		絵文字の使用法解説あり	
	T_{8u} Acc / F1	T_{6u} Acc / F1	T_{8u} Acc / F1	T_{6u} Acc / F1
gpt-3.5-turbo-1106 (zero-shot)	0.34 / 0.29	0.43 / 0.39	0.31 / 0.23	0.47 / 0.43
gpt-3.5-turbo-1106 (8-shot)	0.32 / 0.27	0.46 / 0.42	0.31 / 0.27	0.41 / 0.39
gpt-3.5-turbo-1106 (16-shot)	0.32 / 0.28	0.42 / 0.40	0.34 / 0.29	0.45 / 0.41
gpt-3.5-turbo-1106 (fine-tuning)	**0.43** / **0.39**	**0.54** / **0.51**	0.39 / 0.34	**0.53** / **0.50**
gpt-4-1106-preview (zero-shot)	0.36 / 0.34	0.46 / 0.42	0.41 / 0.38	0.52 / 0.49
gpt-4-1106-preview (8-shot)	0.38 / 0.35	0.48 / 0.43	**0.42** / **0.39**	0.48 / 0.44
gpt-4-1106-preview (16-shot)	0.37 / 0.34	0.47 / 0.43	0.41 / 0.37	0.48 / 0.44
cl-tohoku/bert-base-japanese-v3	0.36 / 0.30	0.48 / 0.44	—	