ニューラルネットワークによる Twitter エンゲージメントの予測

石川県立金沢泉丘高等学校理数科

岩本 周也　篠地 佑宜　清水 慶人　白井 元己

要旨

Twitter API を用いて Twitter からテキストデータとそれに付随する情報を集め、Doc2Vec を用いて下処理を行い、ニューラルネットワークという仕組みを用いて予測モデルを作成した。

1. 研究背景・目的

2022 年 1 月現在、Twitter には 4500 万人ものアクティブユーザーがおり、一般的なメディアとなってきている。その中で、本論文では「バズる」という現象について着目した。「バズる」とは Twitter 上で多くのエンゲージメントを得ることを指す。「バズる」ということの解明を最終目標として、本論文ではツイートに対してのエンゲージメントを予測する数理的モデルの作成を目指す。

2. 研究手法

この研究では「エンゲージメント（Engagement）」を Twitter から収集することのできる「Twitter 社のエンゲージメント」の要素である、「いいね数とリツイート数の和」として定義する。私たちはエンゲージメント予測モデルを作成しそのモデルを分析することで、Twitter の特性を理解し「バズる」ということの再現につながるだろうと考えた。そこで、予備実験にてモデルを作成しそのモデルを考察してよりよいモデルを作成する、といった研究手法をとることにした。以下はモデル作成の手順である。

1. Twitter API を用いてツイートを収集する。
2. テキストの数値化を適切に行うためにテキスト内の重ね表現や、数、絵文字に関して処理を行う。
3. テキストをベクトル（Vector）に変換（数値化）する。
4. そのベクトルとツイートの情報の複合データ（データセット）を用いて、ニューラルネットワークモデルを構築する。
5. データセットの一部をテスト用データとして用いて、ニューラルネットワークモデルの精度を検証する。

3. 評価関数

I. MSE

誤差を二乗するため、誤差を大きく評価するという特徴がある。

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

II. RMSE

MSE の平方根である。2 乗の操作を戻すため、評価基準として用いる。

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

III. MAE

誤差を二乗する MSE に対し誤差を小さく評価するため、外れ値の影響を受けづらいという特徴がある。

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

IV. RMSLE

対数を用いることでデータの値の範囲が広い場合でも誤差を幅として評価できるという特徴がある。

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(log(1+\hat{y}_i) - log(1+y_i)\right)^2}$$

y: i*番目のデータの実際の値*

$\hat{y}$: i*番目のデータの予測値*

4. 予備実験

4-1　データの収集

この研究は将来的に広告の作成に応用することを視野に入れているため、まずはエンゲージメントを求めて投稿されたツイートに多く含まれるであろう、「オススメ」というワードに着目した。

そのため、Twitter から「オススメ」というワードの入った 22029 件のツイートを Twitter API[1]を用いて収集した。

4-2　データの選別

以下の 3 つの条件を含むツイートはモデルの作成において、十分なエンゲージメントを獲得できないために外れ値となることが推測されるため、予め収集の除外対象としデータの選別を行った。(図 1)

図 1. 除外対象のツイート

0 いいね、0 リツイートのもの
経過時間が一日未満のもの
リプライ、引用リツイート

4-3　前処理

後述の文章の数値化の精度を上げるために以下の前処理を行う。(図 2)

図 2. ツイートの前処理の内容

全角・半角の統一
絵文字の除去
重ね表現の除去
桁区切りの除去と数字の置換
URL の除去
記号の置き換え

4-4　数値化

文章データでは数理的処理ができないため、Doc2Vec[2]を用いて収集したテキストを数値化した。

Doc2Vec とは Gensim から提供されている任意の文章をベクトル化する技術、つまり文章から分散表現を獲得することのできる技術である(Embedding)。この研究では、ニューラルネットワークで学習させる際に、文章を数値データとして入力するために用いた。Doc2Vec とは「Document To Vector」の略であり、「PV-DM」と「PV-DBOW」という二つのアルゴリズムを総称したものであるが、この研究では「PV-DM」を用いている。「PV-DM」は、文章 ID とその文章に含まれる任意の数の単語を入力し次に出てくる単語を予測するというタスクを行うことで学習を進めるものである。具体的には、

ツイートデータの ID とそのツイートに含まれる単語から「window」の数値分だけ取り出した単語を、それぞれベクトル化して中間層で結合させ、取り出した単語の次の単語を予測し、文章ベクトルと中間層から出力層への重みを変更する、という形式になっている。この研究で用いた Doc2Vec の学習におけるパラメーターは以下に設定した。(図 3)

図 3. Doc2Vec のパラメーター設定

パラメーター	値
dm	1
window	20
min_count	10
workers	4
epoch	20
alpha	0.075

4-5　予測モデルの作成

今回はモデル作成の時にニューラルネットワークを用いた。これは入力を線形変換するユニットが人間の脳細胞であるニューロンのようにネットワークを構成している数理モデルである。入力層、中間層、出力層の 3 つの層から構成されていて、入力層にデータを入力して、中間層でデータに重みをかけ、出力層から計算結果を出力するというモデルであり、1 次関数を複数組み合わせているため、複雑な事象に対しても予測を行える可能性を保持している。

学習とは、ニューラルネットワークモデルの重みを変化させることで、実際の値と予測値との誤差を減らすようにモデルを最適化させることである。その時に用いるアルゴリズムのことを最適化関数と言い、今回の実験では、Adam[3]というアルゴリズムを用いている。加えて、モデル全体でかける重みを 1 度調整すること、すなわち学習を 1 回行うことを 1Epoch という。

また、学習に伴い、データセットを「訓練データ」と「検証データ」、「テストデータ」に分割した。訓練データは学習時のモデルの精度の向上、検証データは学習推移の可視化、テストデータは最終的なモデルの精度の確認、を用途とする。以下はそれらの分割の割合である。(図 4)

図 4. データセットの分割割合

訓練データ	64%
検証データ	16%
テストデータ	20%

4-6　損失・評価関数

予測の誤差を求める関数で、学習に用いる損失関数は MSE を、テストデータでの精度確認につかう評価関数は前述した MSE 以外を使用した。

4-7　モデル作成に用いた要素

ニューラルネットワークに入力する要素として、各ツイートに対して以下のような要素を組み合わせたデータセットを用意した。(図 5)

図 5. データセットの内訳

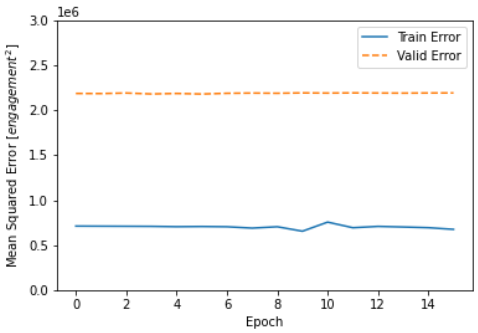
ツイートのベクトルデータ
ツイートをした人の Follow 数
ツイートをした人の Follower 数
ツイートをした曜日
ツイートをした時刻

4-8　結果

ニューラルネットワークにデータセットを入力し学習して、学習過程を以下のようなグラフにした。(図 6) 青色のグラフは訓練データ、オレンジ色のグラフは検証データである。縦軸は MSE の値であり、横軸は

Epoch である。なお、Epoch に関してはニューラルネットワーク側が十分に学習を完了したと認識した時に学習を終える Early Stop を使用した。

図 6.　学習過程



また、テストデータでの評価関数の値は以下のようになった。(図 7)

図 7.　テストデータによる検証

	MAE	RMSE	RMSLE
予備実験	76.6	1084.8	1.63

4-9　考察

グラフより学習が進んでも訓練データ・検証データともに MSE の値が低下していないため、学習がうまく進んでいないとわかる。学習が失敗した要因として以下のような理由が考えられる。(図 8)

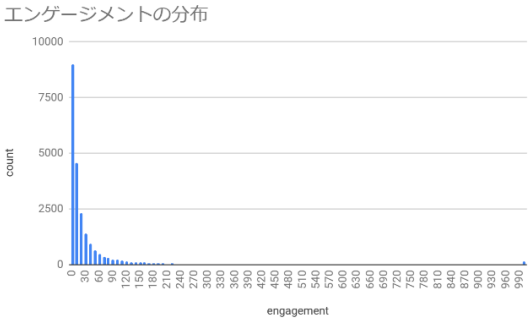
図 8.　学習が進まない要因

- Doc2Vec の Embedding の精度が悪い。
- ツイートの処理方法が悪い。
- ツイートがモデル作成方法に適したものでなかった。
- モデル作成方法に問題がある。

テストデータに関しては、これから行う本実験での比較対象として利用する。

ツイートに関して検証するために、今回の実験で使用したツイートデータに含まれるエンゲージメントのヒストグラムを作成した。(図 9)

図 9.　エンゲージメントの分布



このヒストグラムから今回使用したツイートデータの大半が、エンゲージメントの少ないものであるということがわかる。しかしながら、多くのエンゲージメントを含んだツイートもいくつかあり、ツイートデータの分布が学習において適切なものであったとは言えないと考えられる。

5. 本実験

以降、予備実験の考察よりモデルを改善する。

5-1　実験I

5-1-1　概要

考察からツイートデータが偏って分布し、エンゲージメントの多いツイートが外れ値として学習に影響を及ぼしていると考えられたため、エンゲージメントの多いツイートを除外するためにエンゲージメントの範囲を 250 以下に制限した。また、学習に使用する特徴量を減らすために Doc2Vec で出力するベクトルの次元数を 100 次元に減らして再度モデルを作成した。(図 10)

図 10.　実験Iでの変更点

変更点	予備実験	実験I
Engagement	無制限	250 以下
Vector	400 次元	100 次元

5-1-2　結果

学習過程(図 11)とテストデータによる検証(図 12)は以下のようになった。

図 11.　学習過程

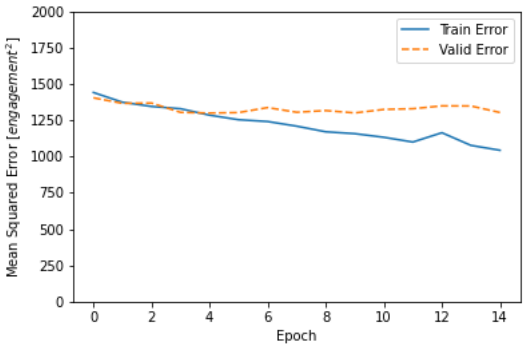


図 12.　テストデータによる検証

	MAE	RMSE	RMSLE
予備実験	76.6	1084.8	1.63
実験I	22.51	37.95	1.28

5-1-3　考察

訓練データに関しては MSE の値が若干下がっているように読み取れるが、検証データでの MSE の値は下がっていないように読み取れる。したがって、未知のデータについて予測ができておらず、学習がうまくいっていないと言える。また、Early Stop によって学習が止められているためこの先 MSE の値が小さくなるということは考えにくい。テストデータに関しては、予備実験からすべての評価関数において減少傾向がみられるため、予備実験のモデルよりは精度のよいモデルが作成できたといえる。この結果はエンゲージメントの範囲制限や次元数削減が精度向上に効果がある可能性を示唆する。

5-2 実験II

5-2-1　概要

Embedding の精度が悪い可能性のある Doc2Vec の代わりに、Google などの検索エンジンに用いられる BERT というアルゴリズムを用いることにした。ここでは、Doc2Vec と BERT の差を比較するためにその他の要素は実験Iと同じにする。(図 13)

図 13.　実験IIでの変更点

変更点	予備実験	実験II
Embedding	Doc2Vec	BERT
Engagement	無制限	250 以下
Vector	400 次元	100 次元

5-2-2　BERT について

BERT とは Google 社が開発を行った自然言語処理モデルである。BERT は、「Bidirectional Encoder Representations from Transformers」の略であり、文章を文頭と文末の双方向から学習することで高い精度で自然言語処理を行うことが可能となったものである。この研究では、東北大学が Transformers に提供している BERT モデルを利用した。このモデルは 12 層のレイヤー、768 次元の隠れ層、12 個のアテンションの構造となっており、日本語 Wikipedia で学習を行ったものである。

5-2-3　結果

学習過程(図 14)とテストデータによる検証(図 15)は以下のようになった。

図 14.　学習過程

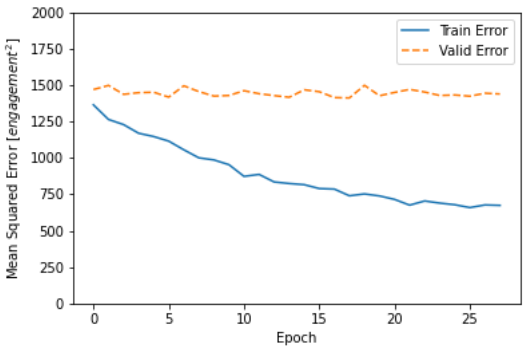


図 15.　テストデータによる検証

	MAE	RMSE	RMSLE
予備実験	76.6	1084.8	1.63
実験I	22.51	37.95	1.28
実験II	22.14	40.25	1.07

5-2-4　考察

訓練データに関しては MSE の値が順調に減少しているが検証データに関しては MSE の値の減少傾向がみられない。よって、このモデルも正しく学習できているとは言えない。テストデータに関しては、予備実験よりも全体的に良い値がでている。実験Iと比較するとあまり大きな差はないが、MSE の値が少し大きいため誤差が少し大きいと考えられる。

5-3 実験III

5-3-1　概要

実験I・実験IIではエンゲージメントの範囲を 250 以下に制限した。これはエンゲージメントの分布が小さいものに偏っていたからである。反対に、この実験では、エンゲージメントの多いツイートからモデルを作成することとした。(図 16)

図 16.　実験IIIでの変更点

変更点	予備実験	実験III
Embedding	Doc2Vec	BERT
Engagement	無制限	popular
Vector	400 次元	次元

5-3-2　ツイートの収集

Twitter API では収集するツイートの対象を popular または recent に設定できる。popular とは、エンゲージメントの多いツイートのことで、recent とは最新のツイートのことである。これまでの実験では、それらを両方とも集めていたが、この実験ではエンゲージメントの多いツイートを収集したいため、popular なツイートを収集する。しかし、popular なツイートは少なく、「オススメ」という単語で絞れば 3 件しか収集できなかったため、格助詞で検索をすることとし、154 件のツイートを収集した。これは、格助詞がほとんどのツイートに含まれていることから疑似的にすべてのツイートから popular なツイートを探すという操作となり、多くのツイートを内容に依らず収集できると考えられる。

5-3-3　結果

MSE での学習過程(図 17)と、学習毎の MSLE(RMSLE の根号を外したもの)の値(図 18)、テストデータによる検証(図 19)は以下のようになった。

図 17.　MSE での学習過程

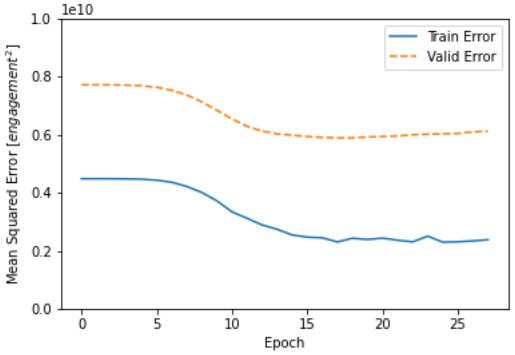


図 18.　学習毎の MSLE の値

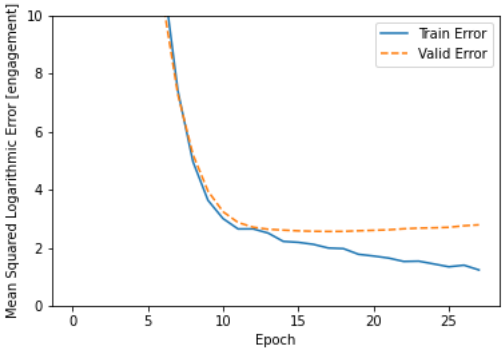


図 19. テストデータによる検証

	MAE	RMSE	RMSLE
予備実験	76.6	1084.8	1.63
実験III	29532.27	44851.09	1.26

5-3-4　考察

今回は MSE のグラフがうまく右肩下がりとなったため、MSLE に関してもグラフを作成した。両方のグラフにおいて訓練データと検証データに関して MSE、MSLE の値が同様に減少しているため、このモデルは正しく学習していると言える。このようになった原因として、popular なツイートの絶対数が少なくモデルを構成するツイート数が少ないことで、かえって学習がうまくいったという理由や、エンゲージメントが多いという特性により学習による予測値の変化が顕著に表れて誤差が減りやすくなった、などの理由が考えられる。テストデータでの検証に関しては、予備実験よりもエンゲージメントが大幅に多くなったため MAE や RMSE が大きくなるのは必然であるため、比較には適していないと考えるが、RMSLE については、減少がみられているため、より精度の高いモデルの作成が行えたと考えられる。

6. 結論

この研究において、エンゲージメントの範囲を絞ることで学習が進行する様子を見ることができた。これより、ツイートについて傾向を見出すことができる可能性があるため、より高い精度でエンゲージメントを予測できる可能性が存在すると結論付ける。最終目標である「バズる」ということの数理的再現には至らなかったが、ツイートの収集やモデル作成等の改善を行えば十分に目標を達成することができるであろうと推測される。

7. 今後の展望

ツイートデータの改善、モデルのパラメーターの変更、学習方法の変更などを通して、より高精度なモデルの作成を行いたい。また、作成したモデルを用いて「バズる」ということの数理的再現を目指す。他にも、「バズる」現象が起こるか否かという視点で、popular なツイートと popular でないツイートをそれぞれラベル付けして分類タスクとして学習することで、「バズる」の数理的再現へのアプローチを増やすことも方法として考えられるだろう。そして、これらを応用すれば、より効果的な広告の作成やツイートの伝達範囲の拡大につながる可能性があるだろう。

8. 参考文献

[1]Joshua Roesslein Revision. “API Reference — tweepy 3.10.0 documentation “. Tweepy. https://docs.tweepy.org/en/v3.10.0/api.html

[2] Quoc Le,Tomas Mikolov. . “Distributed Representations of Sentences and Documents” https://arxiv.org/pdf/1405.4053.pdf

[3]大西健太. “形態素解析前の日本語文書の前処理 (Python)”. 2019-02-09. https://ohke.hateblo.jp/entry/2019/02/09/141500

[4]Tensorflow. “tf.keras.optimizers.Adam|TensorFlow Core v2.7.0 ”. https://www.tensorflow.org/versions/r2.7/api_docs/python/tf/keras/optimizers/Adam

[5]Tensorflow. “回帰：燃費を予測する”. https://www.tensorflow.org/tutorials/keras/regression?hl=ja

[6]Hugging Face. “Transformers”. V.4.16.0. https://huggingface.co/docs/transformers/

9. ソースコード

以下の Github にて本研究で作成したソースコードを公開している。
https://github.com/iwashisardine/TwitterEngagement/tree/main/colab